Black-Box Reliability Certification for AI Agents via Self-Consistency Sampling and Conformal Calibration

Anonymous authors
Paper under double-blind review

Abstract

Given a black-box AI system and a task, at what confidence level can a practitioner trust the system's output? We answer with a *reliability level*—a single number per system–task pair, derived from *self-consistency sampling* and *conformal calibration*, that serves as a black-box deployment gate with *exact, finite-sample, distribution-free* guarantees. Self-consistency sampling reduces uncertainty exponentially; conformal calibration guarantees correctness within $1/(n+1)$ of the target level, regardless of the system's errors—made *transparently visible* through larger answer sets for harder questions. Weaker models earn lower reliability levels (not accuracy—see Definition 2.4): GPT-4.1 earns 94.6% on GSM8K and 96.8% on TruthfulQA, while GPT-4.1-nano earns 89.8% on GSM8K and 66.5% on MMLU. We validate across five benchmarks, five models from three families, and both synthetic and real data. Conditional coverage on solvable items exceeds 0.93 across all configurations; sequential stopping reduces API costs by ∼50%.

1 Introduction

You have an AI system and a task—say, answering math questions or triaging support tickets. Before you deploy it, you need to know: *how much can I trust this system?* Not a vague intuition, but a concrete number with a guarantee attached. That is what this paper provides.

The idea is simple (Figure 1). For each test question, ask the AI system the same question K times (say, K=10). Group the answers that say the same thing and rank them by frequency: the most popular answer might appear 8 out of 10 times, the runner-up 2 out of 10. This frequency ranking is the raw signal—it captures how "sure" the system is without ever looking inside its weights. Then, a human spot-checks a small random batch—just 50−100 items. Each check takes seconds: the human sees the question and the system's top-ranked answer, and marks it right or wrong. From these quick judgments, the framework computes a single number: the *reliability level*. For instance, GPT-4.1 earns 94.6% reliability on grade-school math.

The reliability level comes with a formal guarantee: it is a valid coverage bound that holds regardless of the AI system's systematic errors, and it requires no assumptions about the data distribution. Crucially, the human is verifying, not labeling: the AI generates and ranks the answers; the human just confirms or rejects the top pick. There is no need to build a gold-standard dataset—fifty quick judgments are the entire human effort. A weaker system earns a lower number, not a misleading score; the framework makes trustworthiness *transparent*.

More precisely, current evaluation methods each fail in a different way. *Single-sample evaluation* is unbiased but high-variance—a noisy snapshot of the agent's true capability. *Naive self-consistency* (mode selection) reduces variance but can *amplify* bias: when the agent's most frequent answer is wrong, more samples make the wrong answer look *more* certain. *LLM-as-judge* approaches (Zheng et al., 2023) layer poorly characterized biases—position bias, verbosity preference, self-preference—on top of the agent's own errors, and provide no formal reliability guarantees. None of these methods simultaneously controls both variance and bias.

We introduce the *reliability level* (Definition 2.4)—a single number per agent–task pair that answers this question with provable guarantees. The reliability level is a black-box deployment gate: it requires only

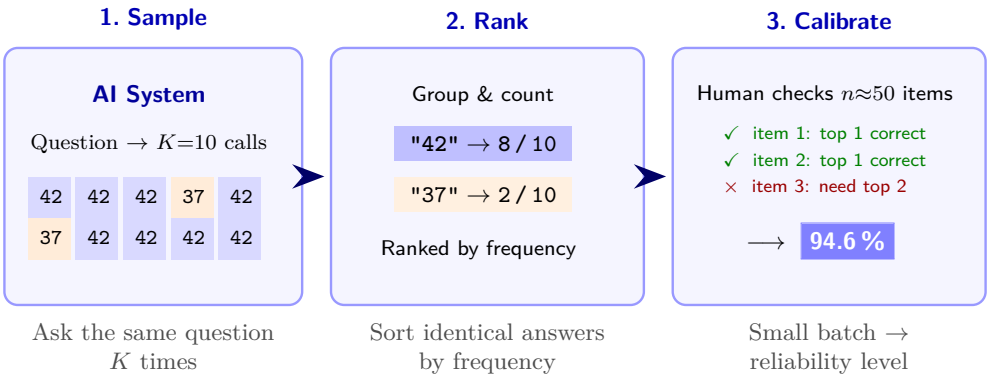

Figure 1: Pipeline overview. **Step 1**: ask the AI system the same question K times and collect its answers. **Step 2**: group identical answers and rank them by frequency. **Step 3**: a human checks a small calibration batch; the framework outputs a single *reliability level* (e.g. 94.6%) with a formal coverage guarantee. No model internals are needed—only API access.

API access, no model internals, and its validity is distribution-free with finite-sample coverage guarantees. Concretely, GPT-4.1 achieves 94.6% reliability on GSM8K and 96.8% on TruthfulQA; GPT-4.1-nano achieves 89.8% on GSM8K and 66.5% on MMLU. Open-weight models (Llama 4 Maverick, Mistral Small 24B) range from 66.7% (MMLU) to 95.4% (GSM8K) across three benchmarks (Table 10). A practitioner can read these numbers directly: "we need X% reliability—which models qualify?"

The mechanism behind the reliability level combines two ingredients. Self-consistency sampling reduces **variance** exponentially via aggregation (Theorem 4.4). Conformal calibration provides coverage guarantees whose **validity is independent of agent bias**: the evaluation's coverage statement is correct regardless of the agent's systematic error profile (Theorem 7.3). Agent bias is not hidden—it is made *transparently visible* through larger prediction sets (Theorem 7.5). A weaker agent earns a lower reliability level, not a misleading score.

A key consequence—and a diagnostic that distinguishes calibration failure from agent limitation—is that marginal coverage can fall below $1-\alpha$ *only* when the agent cannot solve certain items at all. This under-coverage is *predicted* by the theory, not a calibration artifact: conditional coverage on items the model can solve remains near-perfect (≥ 0.93 across all models and benchmarks in our experiments). When observed coverage falls short, the framework identifies *why*: the gap equals the fraction of unsolvable items.

Contributions

Self-consistency sampling (Wang et al., 2022) and conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2021) are individually known. Recent work (Quach et al., 2023; Kumar et al., 2023) applies conformal methods to language models using internal logits or softmax probabilities. Our contribution is a specific synthesis: a conformal score built entirely from *external* sample frequencies, producing a single reliability number for deployment gating—with no access to model internals. Specifically:

1. **Reliability certification for deployment gating** (Definition 2.4, Table 10): the primary practical output—a single reliability level per agent–task pair that answers "at what confidence can I trust this agent?" Validated across five benchmarks, five models from three families (GPT-4.1 ladder, Llama 4 Maverick, Mistral Small), with reliability levels ranging from 66.5% to 96.8%.

2. **A black-box nonconformity score from ranked canonical consensus** (Sections 4–6): the technical construction enabling the reliability level—conformal scores from the rank of the acceptable answer in the self-consistency ordering, leveraging the variance reduction of aggregation while inheriting the distribution-free validity of conformal prediction. This specific construction and its analysis (variance reduction theorems, canonicalization-induced amplification) are new.

3. **Reliability theorems** (Section 7): the evaluation's coverage error is bounded by $1/(n+1)$ regardless of the agent's bias profile (Theorem 7.3); prediction set size is a monotone, transparent diagnostic of agent quality (Theorem 7.5); and the method achieves lower coverage error than LLM-as-judge evaluation once the calibration set exceeds $\lceil 1/|b_J| \rceil - 1$ (e.g., $n \geq 19$ for typical judge bias; Corollary 7.10), though the two approaches answer complementary questions (Remark 7.11).

4. **Bias–variance anatomy of LLM evaluation** (Section 3): the motivating analysis—a formal decomposition showing that single-sample evaluation suffers from variance, LLM-as-judge from irreducible bias, and naive self-consistency from bias amplification.

5. **Sequential sampling with certified early stopping** (Section 9): a Hoeffding-based stopping rule that reduces API cost by ∼50% with no loss in coverage.

Related Work

Self-consistency decoding. Wang et al. (2022) introduced self-consistency as an inference-time strategy: sample multiple reasoning traces and select the most frequent final answer. We repurpose self-consistency for *evaluation*: the ranked consensus provides raw material for conformal calibration. Cordero-Encinar & Duncan (2025) certify when the majority-vote answer has stabilized; our stopping criterion (Theorem 9.1) certifies the *ranking quality* of the top-M candidates for conformal set construction. The two are complementary.

Conformal prediction and language models. Conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2021) provides prediction sets that are distribution-free with finite-sample coverage guarantees. Recent work applies conformal methods to language models: Quach et al. (2023) construct conformal sets over token sequences for language generation; Kumar et al. (2023) apply conformal prediction to NLP classification tasks. These methods define nonconformity scores from model logits or softmax probabilities—internal quantities unavailable for black-box API-based agents. Our score is constructed entirely from *external* ranked consensus over multiple samples, requiring no access to model internals. This black-box property is essential for evaluating proprietary LLM APIs.

Method comparison. Quach et al. (2023) and Kumar et al. (2023) achieve tighter prediction sets when softmax probabilities are available, because internal scores carry richer information than sample frequencies. Our contribution is *generality*: the framework applies to any black-box agent accessible only through sampling—commercial APIs, tool-using agents, multi-step pipelines—and to any task where canonicalization is feasible, including open-ended generation where token-level conformal methods do not apply.

LLM-as-judge and evaluation bias. Zheng et al. (2023) established the LLM-as-judge paradigm. Subsequent studies documented systematic biases: position bias, verbosity bias, self-preference, and anchoring effects (Zheng et al., 2023). Our bias–variance analysis (Section 3) formalizes these observations: judge bias is an irreducible MSE component that does not decay with sample size (Proposition 3.4). Our framework avoids judge bias entirely by using frequency-based ranking rather than quality scoring.

Uncertainty quantification for LLMs. SelfCheckGPT (Manakul et al., 2023) detects hallucinations via inter-sample agreement. Semantic entropy (Kuhn et al., 2023) clusters generations and computes entropy over semantic classes. ConU (Wang et al., 2024) applies conformal prediction using token-level probabilities. Although some APIs expose logprobs, ConU's conformal guarantee requires *calibrated* token-level probabilities—a condition that is unverifiable for proprietary systems and unreliable even when logprobs are nominally available. Our rank-based nonconformity score is the first to provide formal conformal guarantees without any model internals, enabling calibration for any system accessible only through sampling. All three prior methods produce uncertainty *estimates* without formal calibration guarantees. Our framework produces calibrated prediction *sets* with finite-sample coverage guarantees, and uniquely pairs this black-box score with a deployment-gating output—the reliability level—a single actionable number that no prior conformal-LLM method provides. Semantic entropy could serve as a complementary pre-filter within conformal calibration.

Calibration of probabilistic predictions. Classical calibration methods (Platt scaling, temperature scaling, isotonic regression) adjust model confidence scores to match empirical frequencies. These require access to model probabilities and assume a fixed model; they cannot handle open-ended generation where the output space is combinatorial. Conformal prediction, by contrast, is model-agnostic and makes no distributional assumptions, which is why we adopt it as the calibration backbone.

2 Problem Setting

We formalize the setting: an AI system receives a query, produces a stochastic answer, and a task-specific predicate decides whether that answer is acceptable. The goal is to certify, from a small calibration set, how reliably the system's top-ranked answer is acceptable. Our experiments validate the framework on single-turn query-answering systems; extensions to multi-turn interactions are discussed in Section 12.

2.1 Queries, answers, and acceptability

Let $\mathcal{X}$ denote the space of queries and $\mathcal{A}$ the space of possible answers. An AI system with parameters θ defines a stochastic mapping

$$f_\theta : \mathcal{X} \to \mathcal{A}, \qquad f_\theta(x) \sim P_\theta(\cdot \mid x),$$

where $P_\theta(\cdot \mid x)$ is the system's output distribution conditional on query x, accessible via repeated sampling.

Definition 2.1 (Acceptability)**.** For a query $x \in \mathcal{X}$, the set of *acceptable answers* is

$$\mathcal{A}^\star(x) := \{a \in \mathcal{A} : \mathrm{Accept}(x, a) = 1\},$$

where $\mathrm{Accept} : \mathcal{X} \times \mathcal{A} \to \{0, 1\}$ is a task-dependent predicate. We allow $|\mathcal{A}^\star(x)| \geq 1$. For example, on a math problem with answer 42, Accept returns 1 for "42", "42.0", and "forty-two", and 0 for "43".

Assumption 2.2 (Non-triviality)**.** For each query x in the target distribution, $\mathcal{A}^\star(x) \neq \emptyset$.

2.2 Per-query acceptability rate and agent quality

The central quantity is the probability that a single random sample from the agent is acceptable—intuitively, how often the agent "gets it right" on a given question.

Definition 2.3 (Per-query acceptability rate)**.** For a fixed query x, the agent's *acceptability rate* is

$$p^\star(x) := P_\theta\big(f_\theta(x) \in \mathcal{A}^\star(x)\big) = \sum_{a \in \mathcal{A}^\star(x)} P_\theta(a \mid x). \tag{2.1}$$

The *aggregate accuracy* of the agent over a query distribution μ on $\mathcal{X}$ is

$$\bar{p} := \mathbb{E}_{X \sim \mu}[p^\star(X)]. \tag{2.2}$$

The quantities $p^\star(x)$ and $\bar{p}$ are the ground truth that any evaluation method seeks to estimate or certify. Our framework does not estimate $\bar{p}$ directly; instead, it provides a *coverage guarantee* that is a stronger, more actionable statement about reliability.

2.3 The evaluation goal

We seek a procedure that, for each query x, returns a *prediction set* $S(x) \subset \mathcal{C}$ of canonical answer classes such that

$$\mathbb{P}\big(Y(x) \in S(x)\big) \geq 1 - \alpha, \tag{2.3}$$

where $Y(x)$ denotes an acceptable answer for x and $\alpha \in (0, 1)$ is a user-specified miscoverage level. The set $S(x)$ should be as small as possible: a smaller set indicates a more reliable agent on that query.

We now define the target quantity that summarizes this coverage guarantee into a single deployment metric: the highest confidence level at which the agent's most frequent answer is trustworthy.

Definition 2.4 (Reliability level)**.** For a given agent and task, the *reliability level* is

$$1 - \alpha^\star := \frac{|\{i \in \{1, \ldots, n\} : s_i \leq 1\}|}{n+1}, \tag{2.4}$$

where $s_1, \ldots, s_n$ are calibration nonconformity scores (formally defined in Section 6; intuitively, s_i is the rank of the correct answer among the model's self-consistency candidates for item i). Equivalently, $1-\alpha^\star$ is the maximum confidence at which the self-consistency mode alone provides conformal coverage.

Remark 2.5 (Interpreting the reliability level)*.* The numerator of (2.4) counts calibration items where the self-consistency mode is correct, closely related to mode accuracy but with the conformal correction $n+1$ in the denominator that ensures a valid conformal quantile. Rather than asking "does this model achieve coverage at a fixed α?", we invert: "at what confidence level does this model qualify?"

By Theorem 7.1, the reliability level is a *lower bound* on test-time mode-voting coverage: deploying with $\alpha = \alpha^\star$ guarantees $\mathbb{P}(Y \in S(x)) \geq 1 - \alpha^\star$. The gap between the reliability level and empirical test coverage is at most $1/(n+1)$ (e.g., $\leq$0.002 for n=500).

3 Bias and Variance in LLM Evaluation

Before presenting our solution, we develop a formal framework for understanding why LLM evaluation is unreliable, and what "reliable evaluation" requires mathematically. This analysis motivates every design choice in our framework.

3.1 What does reliable evaluation require?

There are three ways to assess an answer: declare it correct or not (binary), assign a quality score (continuous), or output a set of candidates guaranteed to contain the truth (set-valued). Each has a different error profile.

Definition 3.1 (Evaluation method)**.** An *evaluation method* $\mathcal{E}$ is a procedure that, given a query x and access to the agent f_θ, produces an assessment. We consider three types:

1. **Point assessment:** $\mathcal{E}(x) \in \{0, 1\}$ (correct/incorrect).
2. **Score assessment:** $\mathcal{E}(x) \in [0, 1]$ (quality score).
3. **Set assessment:** $\mathcal{E}(x) = S(x) \subset \mathcal{C}$ (prediction set with coverage guarantee).

For point and score assessments, reliability is measured by the mean squared error between the assessment and the true acceptability:

$$\mathrm{MSE}(\mathcal{E}) = \mathbb{E}\big[(\mathcal{E}(X) - p^\star(X))^2\big]. \tag{3.1}$$

The classical bias–variance decomposition applies:

$$\mathrm{MSE}(\mathcal{E}) = \underbrace{\mathbb{E}\big[(\mathbb{E}[\mathcal{E}(X) \mid X] - p^\star(X))^2\big]}_{\mathrm{Bias}^2} + \underbrace{\mathbb{E}\big[\mathrm{Var}(\mathcal{E}(X) \mid X)\big]}_{\mathrm{Variance}}. \tag{3.2}$$

For set assessments, reliability is instead captured by the *coverage validity gap*:

$$\mathrm{Gap}(\mathcal{E}) := \big|\mathbb{P}(Y \in S(X)) - (1 - \alpha)\big|. \tag{3.3}$$

A method with $\mathrm{Gap}(\mathcal{E}) = 0$ achieves exact calibration.

3.2 Error anatomy of current evaluation methods

We now analyze the bias and variance of three standard approaches, establishing the precise deficiencies that our framework addresses.

3.2.1 Single-sample evaluation

The simplest evaluation draws one sample $a \sim P_\theta(\cdot \mid x)$ and checks acceptability:

$$\mathcal{E}_1(x) := \mathbb{1}\{a \in \mathcal{A}^\star(x)\}, \qquad a \sim P_\theta(\cdot \mid x).$$

Proposition 3.2 (Bias–variance of single-sample evaluation)**.** *The single-sample evaluator satisfies:*

$$\mathrm{Bias}(\mathcal{E}_1(x)) = 0, \tag{3.4}$$
$$\mathrm{Var}(\mathcal{E}_1(x) \mid x) = p^\star(x)(1 - p^\star(x)). \tag{3.5}$$

The variance is maximized at $p^\star(x) = 1/2$ (the hardest queries) and equals $1/4$.

Proof. $\mathbb{E}[\mathcal{E}_1(x) \mid x] = P_\theta(a \in \mathcal{A}^\star(x)) = p^\star(x)$, so the bias is zero. The variance follows from $\mathrm{Var}(\mathrm{Bernoulli}(p)) = p(1-p)$. □

Remark 3.3 (Unbiased but unreliable)*.* Single-sample evaluation is unbiased but has maximum variance exactly where it matters most: on queries where the agent is uncertain ($p^\star(x) \approx 1/2$). For a single query, the evaluation is essentially a coin flip when the agent is mediocre. This variance does not decrease without additional samples.

3.2.2 LLM-as-judge evaluation

An LLM judge J scores the agent's output:

$$\mathcal{E}_J(x) := J(x, a), \qquad a \sim P_\theta(\cdot \mid x),$$

where $J : \mathcal{X} \times \mathcal{A} \to [0, 1]$ is stochastic (the judge itself has sampling variance).

Proposition 3.4 (Bias–variance of LLM-as-judge evaluation)**.** *Define the judge's systematic bias on query x with answer a as*

$$b_J(x, a) := \mathbb{E}[J(x, a)] - \mathrm{Accept}(x, a). \tag{3.6}$$

Then:

$$\mathrm{Bias}^2(\mathcal{E}_J(x)) = \big(\mathbb{E}_a[b_J(x, a)]\big)^2, \tag{3.7}$$
$$\mathrm{Var}(\mathcal{E}_J(x) \mid x) = \underbrace{\mathrm{Var}_a(\mathrm{Accept}(x, a))}_{\textit{agent variance}} + \underbrace{\mathbb{E}_a[\mathrm{Var}(J(x, a) \mid x, a)]}_{\textit{judge variance}} + \underbrace{\mathrm{Var}_a(b_J(x, a))}_{\textit{bias variance}}. \tag{3.8}$$

Proof. Write $\mathcal{E}_J(x) = \mathrm{Accept}(x, a) + b_J(x, a) + \eta_J(x, a)$, where η_J is the zero-mean judge noise. By the law of total variance:

$$\mathbb{E}[\mathcal{E}_J(x) \mid x] = \mathbb{E}_a[\mathrm{Accept}(x, a) + b_J(x, a)] = p^\star(x) + \mathbb{E}_a[b_J(x, a)].$$

Hence $\mathrm{Bias}(\mathcal{E}_J(x)) = \mathbb{E}_a[b_J(x, a)]$, giving (3.7). The variance decomposes by conditioning on a and using independence of η_J from the other terms. □

Remark 3.5 (Judge bias is irreducible)*.* The critical flaw of LLM-as-judge is that the bias term $\mathbb{E}_a[b_J(x, a)]$ does not decrease with more judge calls or more agent samples. If the judge systematically overrates verbose answers or underrates unconventional but correct solutions, this error persists regardless of sample size. Furthermore, b_J is unknown and difficult to estimate without ground-truth labels—the very thing evaluation is meant to replace.

3.2.3 Naive self-consistency (mode selection)

Draw K samples, canonicalize, and check whether the most frequent answer is acceptable:

$$\mathcal{E}_{\mathrm{SC}}(x) := \mathbb{1}\{c_{(1)} \in \mathrm{Canon}(x, \mathcal{A}^\star(x))\}.$$

Proposition 3.6 (Bias–variance of mode selection)**.** *Let $p := p^\star(x)$ be the acceptability rate under canonicalization. If $p > 1/2$:*

$$\mathrm{Bias}(\mathcal{E}_{\mathrm{SC}}(x)) = 0, \tag{3.9}$$

$$\mathrm{Var}(\mathcal{E}_{\mathrm{SC}}(x) \mid x) \le \exp\Big(-2K\left(p - \tfrac{1}{2}\right)^2\Big). \tag{3.10}$$

If $p \le 1/2$ (systematic bias):

$$\mathrm{Bias}(\mathcal{E}_{\mathrm{SC}}(x)) \to -p \quad \textit{as } K \to \infty, \tag{3.11}$$

$$\mathrm{Var}(\mathcal{E}_{\mathrm{SC}}(x) \mid x) \to 0 \quad \textit{as } K \to \infty. \tag{3.12}$$

Proof. When $p > 1/2$, the mode is the correct class with probability $\ge 1 - \exp(-2K(p-1/2)^2)$ by Hoeffding's inequality (Hoeffding, 1963), giving zero asymptotic bias and exponentially decaying variance.

When $p \le 1/2$, the total mass on incorrect canonical classes is $1 - p \ge 1/2$. By the law of large numbers, the empirical frequency of each class converges to its true probability as $K \to \infty$. Let $q_{\max} := \max_{c \notin \mathrm{Canon}(x, \mathcal{A}^\star(x))} P_\theta(c \mid x)$ be the mass of the most probable incorrect class. If $q_{\max} > p$, the mode is this incorrect class a.s. If $q_{\max} \le p$, then $p \le 1/2$ implies the correct class shares the maximum with at least one incorrect class; by our tie-breaking convention (uniform random), the mode is incorrect with positive probability, and by symmetry as $K \to \infty$ it is incorrect a.s. whenever the correct class is not the unique mode. In either case, $\mathcal{E}_{\mathrm{SC}}(x) \to 0$ a.s. while $p^\star(x) = p > 0$, yielding Bias $\to -p$. □

Remark 3.7 (The self-consistency double-edged sword). Proposition 3.6 reveals a fundamental asymmetry: self-consistency is excellent for variance reduction when the agent is mostly correct ($p > 1/2$), but it *amplifies confidence in errors* when the agent is systematically wrong ($p \le 1/2$). More samples make the wrong answer look *more* certain. This is the "stable hallucination" phenomenon. Any framework that uses self-consistency *must* account for this failure mode.

Remark 3.8 (The i.i.d. sampling assumption). Proposition 3.6 and Theorem 4.4 assume $a_1, \ldots, a_K \overset{\text{i.i.d.}}{\sim} P_\theta(\cdot \mid x)$. This is well-justified when each sample is an independent, stateless API call to a fixed model at temperature $T > 0$ with deterministic canonicalization (e.g., code execution, string normalization). In practice, positive correlation $\rho > 0$ between samples can arise from model-version drift during data collection, stochastic canonicalization (e.g., an LLM judge), or infrastructure-level batching effects. Under ρ-correlated samples the effective sample size drops to $K_{\mathrm{eff}} \approx K/(1 + (K-1)\rho)$, and the Hoeffding bound degrades to $\exp(-2K_{\mathrm{eff}}(p - \frac{1}{2})^2)$. The qualitative conclusion—more samples reduce variance—survives, but at a slower rate. Crucially, the conformal coverage guarantee (Theorem 6.7) depends only on *exchangeability* of the calibration and test data, not on i.i.d. agent samples, and is therefore unaffected.

3.3 The bias–variance landscape: a summary

The key observation from Table 1 is that no existing method simultaneously controls bias and variance while providing a formal guarantee. Our framework achieves this by (i) using self-consistency for variance reduction, (ii) outputting a *set* rather than a point to absorb residual bias, and (iii) calibrating the set via conformal prediction to obtain an exact, distribution-free coverage guarantee.

4 Self-Consistency Sampling and Canonicalization

4.1 Self-consistency sampling

For a fixed query $x \in \mathcal{X}$, we draw K independent samples from the agent:

$$a_1, a_2, \ldots, a_K \overset{\text{i.i.d.}}{\sim} P_\theta(\cdot \mid x). \tag{4.1}$$

Table 1: Bias–variance profile of evaluation methods (per-query). K = number of samples, n = calibration set size.

Method	Bias	Variance	Error $\to 0$?	Guarantee
Single sample	0	$p^\star(1-p^\star)$	No	—
LLM-as-judge	$\mathbb{E}_a[b_J] \neq 0$ (irreducible)	Prop. 3.4†	No (bias persists)	—
Self-consistency (mode)	0 if $p^\star > \frac{1}{2}$; $\to -p^\star$ if $p^\star \leq \frac{1}{2}$	$\leq \exp(-2K(p^\star - \frac{1}{2})^2)$	No (bias if $p^\star \leq \frac{1}{2}$)	—
Ours (conformal)	**Coverage gap** $\leq 1/(n+1)$		**Yes** ($\to 0$ as $n \to \infty$)	$\mathbb{P}(Y \in S) \geq 1-\alpha$

†Three-term decomposition: agent variance + judge variance + bias variance; does not vanish with more samples.

The empirical distribution over raw answers is

$$\hat{P}_K(a \mid x) = \frac{1}{K} \sum_{i=1}^{K} \mathbb{1}\{a_i = a\}.$$

4.2 Canonicalization

Different surface forms can express the same answer—"42," "42.0," and "The answer is 42" all mean the same thing. Canonicalization maps these to a single representative so that frequency counts reflect semantic agreement, not superficial variation.

Definition 4.1 (Canonicalization)**.** A *canonicalization function* is a mapping

$$\mathrm{Canon} : \mathcal{X} \times \mathcal{A} \to \mathcal{C},$$

where $\mathcal{C}$ is a space of canonical representations. We require that Canon respects semantic equivalence: if $a \simeq a'$ semantically, then $\mathrm{Canon}(x, a) = \mathrm{Canon}(x, a')$.

Applying canonicalization yields the empirical canonical distribution:

$$\hat{P}_K(c \mid x) = \frac{1}{K} \sum_{i=1}^{K} \mathbb{1}\{\mathrm{Canon}(x, a_i) = c\}. \tag{4.2}$$

4.3 Canonicalization: implementation and stability

Since the quality of the consensus vote depends directly on the quality of Canon, we treat canonicalization as a first-class algorithmic component. Three regimes arise in practice:

1. *Deterministic canonicalization* for structured answers—parse "42.0" and "42" to the same integer, match option letters in MCQ tasks, or execute code and record pass/fail. This eliminates surface-form variation entirely.
2. *Embedding-based clustering* for open-ended tasks, using cosine similarity thresholds on dense embeddings to group semantically equivalent responses.
3. *LLM-assisted canonicalization*, where a lightweight model (e.g., GPT-4.1 at temperature 0) classifies each response as correct or incorrect before clustering.

Each approach involves specific failure modes (over-merging vs. under-merging). Implementation details, stability diagnostics, and empirical requirements are in Appendix A.

Assumption 4.2 (Canonicalization quality)**.** Throughout the theoretical analysis (Sections 4.4–7), we assume that Canon correctly maps semantically equivalent answers to the same canonical class. When this assumption is violated, the variance reduction guarantees (Theorem 4.4) degrade gracefully: under-merging reduces p_{canon}, increasing the required K, while the conformal coverage guarantee (Theorem 6.7) remains valid regardless (set sizes simply grow to compensate).

Remark 4.3 (LLM-based canonicalization and circularity). When canonicalization uses an LLM judge (regime (iii) above), a potential circularity arises: if the same model family serves as both agent and canonicalizer, systematic biases could propagate. For instance, the judge might systematically group incorrect answers into a "correct" cluster due to self-preference or verbosity bias, corrupting the consensus vote. Three considerations mitigate this concern:

(a) Functional separation. In our experiments, the canonicalizer (GPT-4.1 at temperature 0, producing deterministic binary labels) operates in a fundamentally different regime from the agent (GPT-4.1 at temperature 0.7, generating stochastic free-form responses). At T=0, the judge is a fixed deterministic function—analogous to code execution or regex matching—not a stochastic evaluator. The bias profiles of deterministic classification and stochastic generation are distinct.

(b) Coverage guarantee is robust to canonicalization errors. Assumption 4.2 states the key safeguard: if the canonicalizer makes errors (merging incorrect answers with correct ones, or splitting correct answers), the conformal coverage guarantee (Theorem 6.7) remains valid. Canonicalization errors manifest as *inflated prediction set sizes*, not as invalid coverage—the framework honestly reflects that canonicalization is noisy by returning larger sets.

(c) Cross-family validation breaks the self-preference loop. Our open-weight experiments provide a direct test: Llama 4 Maverick and Mistral Small 24B are canonicalized by GPT-4.1—a different model family with different training data and biases. If within-family self-preference were corrupting canonicalization, cross-family results would diverge. They do not: coverage (≥ 0.960) and conditional coverage (≥ 0.949) are consistent across all three families (Table 3).

For maximal rigor, we recommend deterministic canonicalization (code execution, numeric extraction, option matching) wherever feasible, reserving LLM-based canonicalization for tasks where no alternative exists. Three of our five benchmarks use deterministic canonicalization.

4.4 Variance reduction via consensus aggregation

We now prove that self-consistency sampling achieves exponential variance reduction for identifying the correct answer, and that canonicalization further amplifies this effect.

Theorem 4.4 (Exponential variance reduction via consensus)**.** *Let $c^\star \in \mathcal{C}$ be the unique acceptable canonical class for query x, and let $p := P_\theta(\mathrm{Canon}(x, f_\theta(x)) = c^\star) > 1/2$. Then after K i.i.d. samples:*

1. ***Mode correctness:***
$$\mathbb{P}(c_{(1)} \neq c^\star) \leq \exp\Big(-2K\left(p - \tfrac{1}{2}\right)^2\Big). \tag{4.3}$$

2. ***Rank concentration:*** *For any $r \geq 2$,*
$$\mathbb{P}(\mathrm{rank}(c^\star; x) \geq r) \leq \binom{|\mathcal{C}|}{r-1} \exp\left(-\frac{K(p - (1-p)/(r-1))^2}{2}\right), \tag{4.4}$$
where $|\mathcal{C}|$ is the number of distinct canonical classes and the bound is nontrivial when $p > (1-p)/(r-1)$.

3. ***Variance decay:***
$$\mathrm{Var}\big(\mathbb{1}\{c_{(1)} = c^\star\}\big) \leq \exp\Big(-2K\left(p - \tfrac{1}{2}\right)^2\Big). \tag{4.5}$$

Proof. **Part 1.** The correct class $c^\star$ has count $N^\star = \sum_{i=1}^K \mathbb{1}\{c_i = c^\star\} \sim \mathrm{Bin}(K, p)$. The mode is incorrect only if some other class has count $\geq N^\star$. Since the total count of all incorrect classes is $K - N^\star$, a necessary condition is $K - N^\star \geq N^\star$, i.e., $N^\star \leq K/2$. By Hoeffding's inequality:

$$\mathbb{P}(N^\star \leq K/2) = \mathbb{P}\left(\frac{N^\star}{K} \leq \frac{1}{2}\right) \leq \exp\left(-2K\left(p - \frac{1}{2}\right)^2\right).$$

Part 2. $\mathrm{rank}(c^\star) \geq r$ requires at least $r-1$ classes to beat $c^\star$'s count. Fix any set T of $r-1$ incorrect classes. Their combined mass is $p_T \leq 1-p$. For each $c' \in T$ to beat $c^\star$, we need $n(c') > n(c^\star)$, which in particular requires the average count of T to exceed $Kp/(r-1)$. By Hoeffding applied to the sum of counts in T:

$$\mathbb{P}(\text{all } c' \in T \text{ beat } c^\star) \leq \mathbb{P}\left(\frac{1}{K}\sum_{c' \in T} n(c') > p\right) \leq \exp\left(-\frac{K(p-p_T)^2}{2}\right).$$

A union bound over $\binom{|\mathcal{C}|}{r-1}$ choices of T gives (4.4).

Part 3. Let $q_K := \mathbb{P}(c_{(1)} = c^\star) \geq 1 - \exp(-2K(p-1/2)^2)$. Then $\mathrm{Var}(\mathbb{1}\{c_{(1)} = c^\star\}) = q_K(1-q_K) \leq 1 - q_K \leq \exp(-2K(p-1/2)^2)$. □

Corollary 4.5 (Sample complexity for δ-reliable mode identification)**.** *To ensure $\mathbb{P}(c_{(1)} = c^\star) \geq 1-\delta$, it suffices to take*

$$K \geq \frac{\ln(1/\delta)}{2(p-1/2)^2}. \tag{4.6}$$

For example, with $p = 0.7$ and $\delta = 0.01$: $K \geq \lceil \ln(100)/(2 \cdot 0.04) \rceil = 58$.

4.5 Canonicalization as variance reduction

Canonicalization does more than enable aggregation—it provably reduces the variance of the consensus by consolidating fragmented probability mass.

Proposition 4.6 (Canonicalization amplifies consensus)**.** *Let $p_{\mathrm{raw}} := \max_{a \in \mathcal{A}^\star(x)} P_\theta(a \mid x)$ be the probability of the most likely acceptable* raw *answer, and $p_{\mathrm{canon}} := P_\theta(\mathrm{Canon}(x, f_\theta(x)) \in \mathrm{Canon}(x, \mathcal{A}^\star(x)))$ the probability of the acceptable* canonical class*. Then:*

$$p_{\mathrm{canon}} \geq p_{\mathrm{raw}}, \tag{4.7}$$

with equality only when canonicalization is the identity. If L raw answers map to the same acceptable canonical class, each with probability $\geq p_{\min}$, then $p_{\mathrm{canon}} \geq L \cdot p_{\min}$.

Consequently, the variance reduction exponent improves from $2K(p_{\mathrm{raw}} - 1/2)^2$ to $2K(p_{\mathrm{canon}} - 1/2)^2$, and the sample complexity (4.6) *decreases by a factor of $\big((p_{\mathrm{raw}} - 1/2)/(p_{\mathrm{canon}} - 1/2)\big)^2$.*

Proof. By definition, $p_{\mathrm{canon}} = \sum_{a:\mathrm{Canon}(x,a)=c^\star} P_\theta(a \mid x) \geq \max_a P_\theta(a \mid x) \cdot \mathbb{1}\{a \in \mathcal{A}^\star(x)\} = p_{\mathrm{raw}}$. The improvement in the Hoeffding exponent follows directly from substituting p_{canon} for p_{raw} in (4.3). □

Remark 4.7 (Canonicalization can change the $p \leq 1/2$ regime)*.* A crucial practical consequence: an agent may have $p_{\mathrm{raw}} < 1/2$ (no single raw answer dominates) but $p_{\mathrm{canon}} > 1/2$ (the correct canonical class dominates after consolidation). Canonicalization can transform a regime where self-consistency *amplifies* bias into one where it *reduces* variance. This provides a formal justification for investing in high-quality canonicalization.

4.6 Ranked consensus

We rank distinct canonical answers by decreasing empirical frequency:

$$\mathrm{rank}(c; x) := \left|\{c' \in \mathcal{C}(x) : \hat{P}_K(c' \mid x) > \hat{P}_K(c \mid x)\}\right| + 1, \tag{4.8}$$

with ties broken uniformly at random, producing an ordering $c_{(1)}, c_{(2)}, \ldots$.

Definition 4.8 (Consensus strength and margin)**.** The *consensus strength* is $\phi(x) := \hat{P}_K(c_{(1)} \mid x)$ and the *consensus margin* is $\Delta(x) := \hat{P}_K(c_{(1)} \mid x) - \hat{P}_K(c_{(2)} \mid x)$.

Consensus strength measures the model's overall confidence (does one answer dominate, or do many answers tie?). Consensus margin measures the gap between the top two—a large margin means the model strongly favors one answer. Both quantities feed into the sequential stopping rule (Section 9).

With the ranked consensus in hand, we now construct prediction sets that adapt their size to each query's difficulty.

5 Set-Valued Predictions

Instead of committing to a single answer, we output the top M most frequent candidates. For easy questions, M=1 suffices; for hard ones, a larger M is needed. The key question is how to choose M with a guarantee—that is the role of conformal calibration in the next section.

Given ranked canonical answers $c_{(1)}, c_{(2)}, \dots$ for query x, define the *top-M prediction set*:

$$S_M(x) := \{c_{(1)}, c_{(2)}, \dots, c_{(M)}\}. \tag{5.1}$$

The family $\{S_M(x)\}_{M\geq 1}$ is nested: $S_1(x) \subseteq S_2(x) \subseteq \cdots$.

A fixed M ignores the varying difficulty of queries. We seek a data-driven procedure that selects $M = M(x)$ adaptively with a formal coverage guarantee—the setting of *conformal prediction*.

6 Conformal Calibration

6.1 Background: split conformal prediction

Conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2021) is a distribution-free framework for constructing prediction sets with guaranteed marginal coverage. Its single requirement is that calibration and test data are interchangeable—their joint distribution does not depend on ordering.

Definition 6.1 (Exchangeability)**.** Random variables $Z_1, \dots, Z_{n+1}$ are *exchangeable* if their joint distribution is invariant under all permutations σ:

$$(Z_1, \dots, Z_{n+1}) \stackrel{d}{=} (Z_{\sigma(1)}, \dots, Z_{\sigma(n+1)}).$$

6.2 Nonconformity scores from ranked consensus

The nonconformity score measures how "surprising" the correct answer is in the ranked list. If the correct answer is the most frequent (rank = 1), the model is confident and the score is low. If the correct answer is buried at rank 5, the score is high—the model's consensus disagrees with the truth.

Definition 6.2 (Rank-based nonconformity score)**.** Let x be a query, $\{c_{(1)}, c_{(2)}, \dots\}$ the ranked canonical answers from K self-consistency samples, and $y \in \mathcal{A}^\star(x)$ an acceptable answer with canonical form $c^y := \mathrm{Canon}(x, y)$. The *rank-based nonconformity score* is

$$s(x, y) := \mathrm{rank}(c^y; x) = \min\{r \in \mathbb{N} : c^y \in S_r(x)\}. \tag{6.1}$$

If $c^y \notin \mathcal{C}(x)$, set $s(x, y) := +\infty$.

Definition 6.3 (Cumulative-probability nonconformity score)**.** An alternative score incorporating frequency information:

$$s^{\mathrm{cp}}(x, y) := \sum_{r=1}^{\mathrm{rank}(c^y;x)} \hat{P}_K(c_{(r)} \mid x) \in [0, 1]. \tag{6.2}$$

6.3 The calibration procedure

Assumption 6.4 (Calibration data)**.** We have a calibration dataset $\mathcal{D}_{\mathrm{cal}} = \{(x_i, y_i)\}_{i=1}^n$ where each $y_i \in \mathcal{A}^\star(x_i)$, drawn exchangeably with the test example (x_{n+1}, y_{n+1}).

Remark 6.5 (When exchangeability holds and when it breaks)*.* Exchangeability is satisfied when calibration and test examples are drawn i.i.d. from the same distribution—the standard setting of benchmark evaluation with random train/test splits. It also holds under random subsampling from any fixed dataset, regardless of how the dataset was originally constructed.

Exchangeability can be violated in several practically relevant scenarios:

1. **Curated benchmark sets**: if items were hand-selected to emphasize difficult cases or specific capabilities, the calibration and test distributions may differ systematically.
2. **Temporal drift**: if the agent is updated between calibration and deployment, the score distribution shifts. Periodic recalibration (re-running the calibration procedure on fresh data from the updated agent) is the standard mitigation.
3. **Adversarial construction**: if test queries are chosen adversarially *after* observing the calibration set, exchangeability fails by design. This is outside our threat model.

When exchangeability is only approximately satisfied (e.g., mild covariate shift between calibration and test), weighted conformal prediction (Section 10.3) provides a principled correction by reweighting calibration scores according to the likelihood ratio $p_{\text{test}}(x)/p_{\text{cal}}(x)$, preserving coverage under the test distribution.

For each calibration example (x_i, y_i):

1. Draw K self-consistency samples for query x_i; canonicalize and rank.
2. Compute $s_i := s(x_i, y_i)$.

Definition 6.6 (Conformal threshold)**.** Define $k := \lceil (n+1)(1-\alpha) \rceil$ and

$$M^\star := s_{(k)}, \tag{6.3}$$

the k-th smallest calibration score.

For a test query x_{n+1}, return:

$$S(x_{n+1}) := S_{M^\star}(x_{n+1}) = \{c_{(1)}, \ldots, c_{(M^\star)}\}. \tag{6.4}$$

6.4 Finite-sample coverage guarantee

Theorem 6.7 (Marginal coverage guarantee)**.** *Let $(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, y_{n+1})$ be exchangeable, and let $s_i = s(x_i, y_i)$. Define $M^\star$ as in* (6.3)*. Then:*

$$\mathbb{P}\big(y_{n+1} \in S_{M^\star}(x_{n+1})\big) \geq 1 - \alpha. \tag{6.5}$$

If the scores have no ties a.s.:

$$1 - \alpha \;\leq\; \mathbb{P}\big(s_{n+1} \leq M^\star\big) \;\leq\; 1 - \alpha + \frac{1}{n+1}. \tag{6.6}$$

Proof. By exchangeability, the scores $s_1, \ldots, s_{n+1}$ are exchangeable. The rank of s_{n+1} among $\{s_1, \ldots, s_{n+1}\}$ is uniformly distributed over $\{1, \ldots, n+1\}$.

Define $k := \lceil (n+1)(1-\alpha) \rceil$. The event $\{s_{n+1} \leq M^\star\}$ occurs when s_{n+1}'s rank is at most k. In the no-ties case:

$$\mathbb{P}(s_{n+1} \leq M^\star) = \frac{k}{n+1} = \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1}.$$

Since $\lceil (n+1)(1-\alpha) \rceil \geq (n+1)(1-\alpha)$, we get $\mathbb{P} \geq 1 - \alpha$. Since $\lceil (n+1)(1-\alpha) \rceil \leq (n+1)(1-\alpha) + 1$, we get $\mathbb{P} \leq 1 - \alpha + 1/(n+1)$.

With ties, coverage can only increase. □

6.5 Adaptive prediction sets

Using the cumulative-probability score s^{cp} and its conformal quantile $\hat{q}^{\text{cp}}_{1-\alpha}$:

$$S^{\text{cp}}(x) := \big\{c_{(r)} : r = 1, \ldots, R(x)\big\}, \quad R(x) := \min\left\{ r : \sum_{j=1}^{r} \hat{P}_K(c_{(j)} \mid x) \geq \hat{q}^{\text{cp}}_{1-\alpha} \right\}. \tag{6.7}$$

For high-consensus queries, $R(x)$ is small; for diffuse queries, it is larger. Coverage is preserved by the same conformal argument.

7 Reliability Guarantees

Conformal set-valued evaluation is *exactly calibrated*: the evaluation's coverage statement has bounded error. It is also *transparently diagnostic* of agent quality—agent bias is surfaced through prediction set size rather than hidden. These two properties distinguish the *agent's* systematic errors (diagnosed but not fixed) from the *evaluation's* systematic errors (provably controlled).

7.1 Conservative coverage guarantee

The evaluation's coverage error shrinks to zero as the calibration set grows—unlike point estimators, where bias persists indefinitely.

Theorem 7.1 (Coverage error control)**.** *Under the conditions of Theorem 6.7, the* coverage gap *satisfies*

$$0 \;\le\; \mathbb{P}(Y_{n+1} \in S(X_{n+1})) - (1-\alpha) \;\le\; \frac{1}{n+1}. \tag{7.1}$$

That is, the evaluation is conservative (never under-covers) and the over-coverage vanishes as $n \to \infty$.

Proof. This is immediate from (6.6): the lower bound gives conservativeness, and the upper bound gives the $1/(n+1)$ slack. □

Remark 7.2 (Contrast with score-based evaluation)*.* Conformal calibration is the only method whose evaluation error vanishes with sample size. For single-sample evaluation, the error analog is $\mathrm{Var}(p^\star(X))$, which depends on the query distribution and does not shrink. For LLM-as-judge, the error includes the irreducible judge bias $\mathbb{E}[b_J]$ (Remark 3.5). Neither can be eliminated by collecting more data.

7.2 Bias immunity: coverage holds regardless of agent quality

Theorem 7.3 (Bias immunity of conformal coverage)**.** *The coverage guarantee* (6.5) *holds for* any *agent f_θ, regardless of:*

1. *the agent's per-query acceptability rate $p^\star(x)$ (including $p^\star(x) = 0$),*
2. *the presence of systematic bias or stable hallucinations,*
3. *the output distribution $P_\theta(\cdot \mid x)$,*
4. *the number of acceptable answers $|\mathcal{A}^\star(x)|$.*

Formally: let Θ denote the space of all possible agent parameters. Then

$$\inf_{\theta \in \Theta} \mathbb{P}_\theta\big(Y_{n+1} \in S(X_{n+1})\big) \ge 1-\alpha, \tag{7.2}$$

where $\mathbb{P}_\theta$ denotes the joint distribution over calibration data, agent samples, and the test example under agent θ.

Proof. The proof of Theorem 6.7 uses only exchangeability of (x_i, y_i), which is a property of the data-generating process, not of the agent. The agent enters only through the nonconformity scores s_i, and the conformal argument is valid for *any* score distribution. Specifically:

For any fixed θ, the scores $s_1, \ldots, s_{n+1}$ are determined by the exchangeable data $(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})$ and the independent agent samples $\{a_j^{(i)}\}$. Since agent samples for different queries are independent, and the (x_i, y_i) are exchangeable by assumption, the scores remain exchangeable. The uniform rank argument then yields $\mathbb{P}_\theta(s_{n+1} \le M^\star) \ge 1-\alpha$ for every θ. □

Remark 7.4 (What "bias immunity" means and does not mean). Theorem 7.3 is about the *evaluation's* accuracy, not the agent's quality. It does not fix or remove agent bias. It guarantees that the coverage statement—"an acceptable answer lies in $S(x)$ with probability $\geq 1-\alpha$"—is true regardless of how biased the agent is.

Concretely: a completely broken agent earns $M^\star = +\infty$ (honestly reflecting failure), while a strong agent earns $M^\star = 1$. In both cases the coverage guarantee holds. A single calibration procedure is valid for any agent, including ones with unknown or adversarial bias profiles.

7.3 Bias transparency: set size as honest quality diagnostic

The prediction set size $|S(x)|$ is a *diagnostic signal* that faithfully reflects the agent's quality. The following theorem makes this precise: (1) better agents get smaller sets; (2) perfect agents get singletons; (3) agents that cannot solve a task get infinitely large sets, honestly reflecting failure; (4) expected set size scales with average answer rank.

Theorem 7.5 (Bias transparency—set size reflects agent quality)**.** *Let $M^\star(\theta, \alpha)$ denote the conformal threshold for agent θ at level α. Then:*

1. ***Monotonicity in agent quality:*** *If agent θ_1 stochastically dominates agent θ_2 in the sense that $s_i^{(\theta_1)} \leq_{\text{st}} s_i^{(\theta_2)}$ for each calibration example i (i.e., θ_1 consistently produces acceptable answers at higher ranks), then*

$$M^\star(\theta_1, \alpha) \leq M^\star(\theta_2, \alpha) \quad \textit{almost surely.} \tag{7.3}$$

2. ***Perfect agent:*** *If $p^\star(x) = 1$ for all x in the calibration distribution (the agent always produces acceptable answers), then $s_i = 1$ for all i, and $M^\star = 1$.*

3. ***Biased agent:*** *Let $\beta := \mathbb{P}_X(p^\star(X) < 1/K)$ be the fraction of queries where the agent is unlikely to produce an acceptable answer. If $\beta > \alpha$, then $M^\star = +\infty$ (no finite prediction set suffices at level α).*

4. ***Set size–bias correspondence:*** *The expected set size satisfies*

$$\mathbb{E}[|S(X)|] \geq \mathbb{E}\big[\text{rank}\big(\text{Canon}(X, Y(X));\ X\big)\big] \cdot (1-\alpha) - O(1/n). \tag{7.4}$$

Proof. **Part 1.** If $s_i^{(\theta_1)} \leq_{\text{st}} s_i^{(\theta_2)}$ for each i, then the k-th order statistic satisfies $s_{(k)}^{(\theta_1)} \leq s_{(k)}^{(\theta_2)}$ stochastically.

Part 2. If $p^\star(x) = 1$, then with probability 1 all K samples are acceptable, so the acceptable canonical class has rank 1: $s_i = 1$ for all i, and $M^\star = s_{(k)} = 1$.

Part 3. If $p^\star(x_i) < 1/K$, then $\mathbb{P}(s_i = +\infty) \geq (1 - 1/K)^K \geq e^{-1} - o(1) > 0$. When $\beta > \alpha$, the number of infinite scores exceeds $n\beta > n\alpha$, so the $\lceil (n+1)(1-\alpha) \rceil$-th score is $+\infty$.

Part 4. By definition, $|S(X)| = M^\star$ for the global threshold. For the adaptive version, $|S(X)| \geq \text{rank}(c^{Y(X)}; X)$ whenever $Y(X) \in S(X)$. Taking expectations and using $\mathbb{P}(Y \in S(X)) \geq 1-\alpha$:

$$\mathbb{E}[|S(X)|] \geq \mathbb{E}[\text{rank}(c^{Y(X)}; X) \cdot \mathbb{1}\{Y \in S(X)\}] \geq \mathbb{E}[\text{rank}(c^{Y(X)}; X)] \cdot (1-\alpha) - \mathbb{E}[\text{rank} \cdot \mathbb{1}\{Y \notin S\}].$$

The second term is bounded and yields the $O(1/n)$ correction. □

Remark 7.6 (Finite candidate spaces and under-coverage). Part 3 predicts $M^\star = +\infty$ when $\beta > \alpha$, but in practice $M^\star$ is bounded by the number of distinct canonical classes $|\mathcal{C}(x)|$ observed in K samples. When $|\mathcal{C}(x)|$ is small (e.g., $|\mathcal{C}| = 4$ for MCQ tasks), the conformal quantile remains finite even though some items are unsolvable. Marginal coverage still falls below $1-\alpha$ because, for items where $p^\star(x) = 0$, the correct class never enters the candidate pool—no prediction set over observed candidates can cover it. The under-coverage thus equals the unsolvable fraction β, not a calibration error: conditional coverage on solvable items remains near-perfect (Table 2).

Remark 7.7 (Bias is visible, not hidden). The fundamental distinction from LLM-as-judge evaluation is that bias in the agent is *surfaced* through larger prediction sets, not *concealed* in an opaque score. A practitioner who observes $M^\star = 8$ knows immediately that the agent frequently fails to rank acceptable answers highly. An LLM-judge score of 0.75 carries no such interpretable diagnostic—the same score could arise from high-quality answers with a harsh judge, or poor-quality answers with a lenient judge.

7.4 Formal comparison with alternative evaluation methods

We now compare the conformal method's coverage guarantee against two standard baselines—single-sample evaluation and LLM-as-judge—showing that conformal calibration achieves lower evaluation error once the calibration set is large enough to undercut the judge's bias.

Theorem 7.8 (Advantage over single-sample evaluation)**.** *For any agent and any target reliability level* $1-\delta$*, define the* evaluation reliability *as the probability that the evaluation's assessment is within* ϵ *of the truth.*

For single-sample evaluation applied to a test set of N *queries:*

$$\mathbb{P}\big(\big|\hat{p}_{\text{single}} - \bar{p}\big| > \epsilon\big) \leq 2\exp(-2N\epsilon^2) \quad \textit{(Hoeffding)}, \tag{7.5}$$

where $\hat{p}_{\text{single}} = N^{-1}\sum_j \mathbb{1}\{a_j \in \mathcal{A}^\star(x_j)\}$.

For conformal set evaluation:

$$\mathbb{P}\Big(\big|\widehat{\text{Cov}}(\alpha) - (1-\alpha)\big| > \epsilon\Big) \leq 2\exp(-2N\epsilon^2) + \frac{1}{n+1}, \tag{7.6}$$

where $\widehat{\text{Cov}}$ *is the empirical coverage on the test set.*

Both converge at rate $O(1/\sqrt{N})$*, but the conformal method provides:*

1. *A* per-query guarantee *(*$Y(x) \in S(x)$*), not just an aggregate estimate.*
2. *An explicit reliability level* $1-\alpha$ *chosen by the user, vs. an unknown* $\bar{p}$ *that must be estimated.*
3. *A diagnostic set size per query, revealing per-query difficulty.*

Proof. Equation (7.5) is Hoeffding's inequality for i.i.d. Bernoulli random variables. For (7.6): by Theorem 6.7, $\mathbb{P}(Y_j \in S(x_j)) \in [1-\alpha,\ 1-\alpha+1/(n+1)]$ marginally. The empirical coverage $\widehat{\text{Cov}}$ averages N such indicators (approximately independent across test queries). Hoeffding's inequality applied to these indicators, centered at $\mathbb{P}(Y \in S(X))$, gives the exponential tail. The $1/(n+1)$ term accounts for the gap between $\mathbb{P}(Y \in S(X))$ and $1-\alpha$. □

Theorem 7.9 (Advantage over LLM-as-judge)**.** *Let the LLM judge have systematic bias* $b_J = \mathbb{E}_{X,a}[b_J(X,a)] \neq 0$ *and judge variance* $\sigma_J^2 = \mathbb{E}_{X,a}[\text{Var}(J(X,a) \mid X,a)]$*. Then for any sample size* N*:*

LLM-as-judge evaluation error:

$$\text{MSE}(\hat{p}_J) = b_J^2 + \frac{\sigma_J^2 + \bar{p}(1-\bar{p})}{N} + O(N^{-2}). \tag{7.7}$$

The bias term b_J^2 does not decay with N.

Conformal evaluation error (coverage gap):

$$\big|\mathbb{P}(Y \in S(X)) - (1-\alpha)\big| \leq \frac{1}{n+1}. \tag{7.8}$$

The error is controlled solely by the calibration set size n *and is* independent of the agent's bias, the judge's bias, or any systematic error.

Proof. For the judge: $\hat{p}_J = N^{-1} \sum_j J(x_j, a_j)$. Then $\mathbb{E}[\hat{p}_J] = \bar{p} + b_J$, so $\mathrm{Bias}(\hat{p}_J) = b_J$. The variance is $\mathrm{Var}(\hat{p}_J) = N^{-1} \mathrm{Var}(J(X, a))$, decomposed via the law of total variance into agent variance and judge variance. The MSE follows.

For conformal: this is a restatement of Theorem 7.1. □

Corollary 7.10 (When does conformal evaluation achieve lower error?)**.** *Under the MSE comparison criterion (where coverage gap and judge bias are both expressed as deviations from the true acceptability rate), conformal set evaluation achieves lower coverage error than LLM-as-judge whenever*

$$|b_J| > \frac{1}{n+1}. \tag{7.9}$$

For reference, if the judge's bias reaches $|b_J| \geq 0.05$—a level documented in the literature (Zheng et al., 2023) for subjective evaluation tasks—then $n \geq 19$ suffices. On structured tasks with unambiguous answers (e.g., GSM8K, MMLU), judge accuracy can exceed conformal coverage (Table 2), implying smaller effective bias and a correspondingly larger crossover point. The comparison is most informative on ambiguous tasks where judge bias is hardest to bound. Note that this comparison is meaningful when both methods are assessed by their distance to the true acceptability rate; conformal sets and judge scores are complementary evaluation outputs (Remark 7.11).

Remark 7.11 (Complementary, not universally dominant)*.* The "advantage" in Corollary 7.10 compares coverage gap (our method) against MSE (point estimators). These metrics answer different questions: coverage gap measures whether the evaluation's reliability claim is valid, while MSE measures how accurately the evaluation estimates the agent's true accuracy $\bar{p}$. For *reliability certification*—"can I trust this agent at level $1 - \alpha$?"—conformal calibration is strictly superior once $n \geq \lceil 1/|b_J| \rceil - 1$ (e.g., $n \geq 19$ when $|b_J| = 0.05$). For *accuracy estimation*—"what fraction of queries does the agent answer correctly?"—simple point estimators with confidence intervals remain useful. The methods are complementary: practitioners should use conformal sets for per-query reliability assessment and point estimates for aggregate performance reporting.

7.5 Variance of set-size as a quality estimator

Since set size serves as a quality diagnostic, we need it to be a stable estimator—the average set size should concentrate around its true mean as the test set grows.

Proposition 7.12 (Concentration of the set-size estimator)**.** *Let $\bar{M} := N^{-1} \sum_{j=1}^{N} |S(x_j)|$ be the average set size on a test set of N queries. If $|S(x)| \leq B$ almost surely (bounded set size), then*

$$\mathbb{P}\big(\big|\bar{M} - \mathbb{E}[|S(X)|]\big| > t\big) \leq 2 \exp\left(-\frac{2Nt^2}{B^2}\right). \tag{7.10}$$

The average set size concentrates around its expectation at rate $O(B/\sqrt{N})$ and provides a reliable, low-variance estimator of agent quality.

Proof. Each $|S(x_j)| \in [1, B]$ is bounded. Apply Hoeffding's inequality. □

These variance bounds on set size motivate the next question: how does the coverage–efficiency trade-off depend on agent competence?

8 Coverage–Efficiency Trade-Off

A better agent produces smaller prediction sets—but how much smaller? This section quantifies the relationship between agent quality and set size, showing that the prediction set is an efficient diagnostic: strong agents need only singleton sets, while weak agents unavoidably require larger ones.

8.1 Set size as a function of agent competence

Proposition 8.1 (Expected rank under single-acceptable-class)**.** *Suppose there is a unique acceptable canonical class $c^\star$ with $p = P_\theta(c_i = c^\star) > 1/2$. Then:*

$$\mathbb{E}[\mathrm{rank}(c^\star; x)] \leq 1 + \frac{1-p}{p} \cdot \frac{K-1}{K} \xrightarrow{K\to\infty} 1 + \frac{1-p}{p} = \frac{1}{p}. \tag{8.1}$$

If $p > 1/2$, then $\mathbb{E}[\mathrm{rank}(c^\star; x)] < 2$ for all K.

Proof. $\mathrm{rank}(c^\star) = 1 + |\{c' \neq c^\star : n(c') \geq n(c^\star)\}|$. For any competing class c' with probability $p_{c'} < p$:

$$\mathbb{P}(n(c') \geq n(c^\star)) \leq \mathbb{P}(n(c') \geq Kp/2) + \mathbb{P}(n(c^\star) \leq Kp/2).$$

Summing over all competing classes (total mass $1-p$) and applying a stochastic dominance argument: $\mathbb{E}[\mathrm{rank}(c^\star)] \leq 1 + (1-p)/p \cdot (K-1)/K$. □

8.2 Set size inflation under ambiguity

Proposition 8.2 (Ambiguity inflates calibration scores)**.** *If a fraction β of calibration queries have $p^\star(x_i) < 1/K$ and $\beta > \alpha$, then $M^\star = +\infty$.*

Proof. At least βn scores are $+\infty$ with high probability. The k-th order statistic with $k = \lceil (n+1)(1-\alpha) \rceil$ is infinite when $k > n - \beta n$, which holds when $\beta > \alpha + 1/(n+1)$. □

Remark 8.3. This is a feature, not a bug: $M^\star = +\infty$ tells the practitioner that the agent cannot reliably serve this query population at the desired confidence level. No other evaluation method provides such a clear signal.

9 Sequential Sampling with Certified Early Stopping

Drawing K samples per query can be expensive. We develop sequential procedures that stop early when consensus is clear, reducing cost without sacrificing coverage.

9.1 The sequential consensus problem

After k samples, let $\hat{p}_1^{(k)}$ and $\hat{p}_2^{(k)}$ be the frequencies of the top two candidates, with margin $\Delta_k := \hat{p}_1^{(k)} - \hat{p}_2^{(k)}$.

9.2 Hoeffding-based stopping criterion

Theorem 9.1 (Certified mode identification)**.** *Define the stopping time*

$$\tau_\delta := \min\left\{ k \geq k_0 : \Delta_k > \sqrt{\frac{2\ln(2|\mathcal{C}(x)|k^2/\delta)}{k}} \right\}. \tag{9.1}$$

If the true mode $c^\star$ has $p_1 > p_2 := \max_{c \neq c^\star} P_\theta(c \mid x)$, then $\mathbb{P}(c_{(1)}^{(\tau_\delta)} = c^\star) \geq 1 - \delta$.

Proof. At step k, by Hoeffding and a union bound over $|\mathcal{C}(x)|$ classes:

$$\mathbb{P}\big(\exists c : |\hat{P}_k(c \mid x) - P_\theta(c \mid x)| > \epsilon\big) \leq 2|\mathcal{C}(x)| \exp(-2k\epsilon^2).$$

Setting $\epsilon = \Delta_k/2$ and requiring the bound to be $\leq \delta/k^2$ (enabling a sum over k via $\sum 1/k^2 < 2$) yields (9.1). When triggered, the true frequencies are within $\Delta_k/2$ of empirical values with probability $\geq 1-\delta$, so the empirical mode equals the true mode. □

9.3 Variance reduction from sequential stopping

Proposition 9.2 (Variance–cost trade-off of sequential stopping)**.** *Let $K_{\max}$ be the maximum sample budget and τ the stopping time from* (9.1)*. Then:*

1. ***Easy queries*** *(large true margin $p_1 - p_2 = \Delta > 0$): $\mathbb{E}[\tau] = O(\Delta^{-2} \ln(|\mathcal{C}|/\delta))$, matching the classical sample complexity of fixed-confidence best-arm identification (Even-Dar et al., 2006).*

2. ***Hard queries*** *(small Δ): $\mathbb{E}[\tau] \approx K_{\max}$ (no savings).*

The cost reduction is concentrated on queries where variance is already low (high consensus), preserving the framework's reliability exactly where it is most needed.

9.4 Validity of conformal guarantee under adaptive stopping

Proposition 9.3 (Conformal validity with adaptive K)**.** *If the stopping rule $K_i = K_i(x_i, a_1^{(i)}, \dots)$ depends only on x_i and agent samples (not on y_i), then $s_1, \dots, s_{n+1}$ remain exchangeable and Theorem 6.7 holds.*

Proof. The score s_i is a function of (x_i, y_i) and the agent samples $\{a_j^{(i)}\}_{j=1}^{K_i}$. Since K_i depends only on $(x_i, \{a_j^{(i)}\})$ and not on y_i, and (x_i, y_i) are exchangeable, the scores inherit exchangeability. □

Note that the stopping rule (Theorem 9.1) certifies mode identity, not the full rank ordering. For items where $s_i = 1$ (the correct answer is the mode), mode stability implies rank stability. For items with $s_i > 1$, rank fluctuations after stopping may affect prediction set size but not coverage validity, since Proposition 9.3 guarantees exchangeability regardless of when sampling stops.

10 Extensions

10.1 Multiple acceptable canonical classes

When $|\mathrm{Canon}(x, \mathcal{A}^\star(x))| > 1$, use:

$$s_i = \min_{l=1,\dots,L_i} \mathrm{rank}\big(\mathrm{Canon}(x_i, y_i^{(l)});\ x_i\big). \tag{10.1}$$

Coverage is preserved since exchangeability is maintained.

10.2 Dependence-aware extensions

Remark 10.1 (Separation of concerns)*.* A key structural insight: the conformal coverage guarantee depends on exchangeability of *calibration-test pairs* (x_i, y_i), *not* on independence of the K within-query samples. Dependence among the K samples affects ranking *quality* (and hence set *size*) but not coverage *validity*. If dependence degrades the ranking, $M^\star$ grows—the framework self-corrects by widening the set—without breaking the guarantee. This separation is what makes the method robust to the poorly characterized dependence structure of LLM API calls.

10.3 Weighted conformal prediction

Under covariate shift between calibration and test distributions, use likelihood-ratio-weighted conformal prediction (Tibshirani et al., 2019):

$$M_w^\star := \text{weighted quantile of } s_1, \dots, s_n \text{ with weights } w_i \propto \frac{p_{\mathrm{test}}(x_i)}{p_{\mathrm{cal}}(x_i)}. \tag{10.2}$$

11 Empirical Study

We first validated all theoretical results on controlled synthetic agents with known parameters (Appendix C); all predictions—coverage calibration, exponential variance decay, set-size monotonicity, and canonicalization amplification—are confirmed.

We then evaluate on five real benchmarks spanning code generation, mathematical reasoning, open-ended question answering, and multiple-choice selection, using five models from three families (GPT-4.1 ladder, Llama 4 Maverick, Mistral Small 24B). Four questions structure the evaluation:

(Q1) Does conformal calibration achieve coverage $\geq 1 - \alpha$ empirically (Theorem 6.7)?

(Q2) Does self-consistency reduce variance as K increases (Theorem 4.4)?

(Q3) Does set size adapt to model uncertainty (Theorem 7.5)?

(Q4) Does the framework achieve lower coverage error than single-sample and LLM-judge baselines (Theorems 7.8, 7.9)?

Hypotheses. We formalize six empirical hypotheses, each derived from a specific theoretical result:

(H1) **Coverage validity.** Empirical coverage meets the $1-\alpha$ target on tasks where the model has sufficient capability; any shortfall is attributable to model capability gaps, not calibration failure (Theorem 6.7).

(H2) **Variance reduction.** Mode identification error decreases with K, with the largest reductions on high-accuracy tasks where $p^\star$ is far from 0.5 (Theorem 4.4).

(H3) **Adaptive set size.** Prediction set size correlates positively with per-item answer entropy (Theorem 7.5).

(H4) **Canonicalization benefit.** Canonicalization reduces prediction set size on tasks with surface-form variation (Proposition 4.6).

(H5) **Baseline comparison.** Conformal coverage meets or exceeds LLM-judge accuracy on ambiguous tasks where judge bias is non-negligible (Corollary 7.10).

(H6) **Sequential efficiency.** The Hoeffding-based stopping rule reduces the average number of samples with no loss in coverage (Theorem 9.1).

11.1 Experimental setup

Datasets. We select five benchmarks representing distinct task families and canonicalization strategies:

1. **HumanEval** (code generation, 164 items). Each response is executed in a sandboxed environment against unit tests; canonicalization is deterministic binary: `pass` or `fail`. This represents the lowest-ambiguity setting where correctness is objectively verifiable.

2. **TruthfulQA** (open-ended QA, 817 items). Free-form text responses where surface-form variation is extreme—even correct answers differ substantially in phrasing. Canonicalization uses an LLM judge (GPT-4.1 at temperature 0) to classify each sample as *correct* or *incorrect*, yielding a binary answer space analogous to code execution.

3. **BigBench MovieRec** (multiple-choice, 500 items; hereafter "BigBench"). Responses are canonicalized via option matching and text normalization to the selected movie title.

4. **GSM8K** (mathematical reasoning, 1319 items) (Cobbe et al., 2021). Grade-school math word problems requiring multi-step arithmetic. Canonicalization is deterministic numeric extraction: we parse the final answer after the `####` delimiter and normalize to a standard integer form. This tests the framework on a task with a unique correct answer but diverse reasoning paths.

5. **MMLU** (multiple-choice knowledge, 1000 items sampled from the full test set). Four-option questions spanning 57 academic subjects. Canonicalization reuses the MCQ pipeline (option matching and text normalization).

Agent and sampling configuration. We use the GPT-4.1 model family as a capability ladder with unambiguous ordering—GPT-4.1 (strong), GPT-4.1-mini (mid-tier), GPT-4.1-nano (weak)—providing a clean test of the set-size monotonicity prediction. All models are evaluated at temperature $T = 0.7$ with $K_{\max} = 20$ independent samples per item ($K_{\text{fixed}} = 10$ for both calibration and test-time prediction in primary results):

- **GPT-4.1** (strong): evaluated on all five benchmarks. Expected to yield the smallest prediction sets.
- **GPT-4.1-mini** (mid-tier): evaluated on GSM8K and MMLU, the two benchmarks with largest calibration sets ($n_{\text{cal}} = 500$).
- **GPT-4.1-nano** (weak): evaluated on GSM8K and MMLU. Expected to yield the largest prediction sets.

As a supplementary comparison, we evaluate **GPT-5 mini** on all five benchmarks. Under our T=0.7 i.i.d. sampling protocol (without extended thinking), GPT-5 mini exhibits lower per-sample accuracy than GPT-4.1. Throughout, $M^\star$ and the reliability level are properties of a specific *(model, inference configuration)* pair—not of the model architecture alone. The same model can yield different prediction sets depending on temperature, decoding strategy, and whether capabilities like extended thinking are activated.

To test cross-family generalizability, we evaluate two open-weight models via Together AI on GSM8K, MMLU, and TruthfulQA:

- **Llama 4 Maverick 17B** (`meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8`): a mid-tier mixture-of-experts model from Meta.
- **Mistral Small 24B** (`mistralai/Mistral-Small-24B-Instruct-2501`): a smaller instruction-tuned model from Mistral AI.

The calibration/test split uses up to $n_{\text{cal}} = n_{\text{test}} = 500$ items per dataset (or half the dataset if smaller, e.g., HumanEval uses 82/82). The LLM judge is fixed at GPT-4.1 (temperature 0) across all experiments to avoid confounding evaluation quality with agent capability. All API responses are SHA-256 cached so that re-runs are deterministic and cost-free.

Evaluation protocol. All experiments use the rank-based nonconformity score (Definition 6.2): for each calibration item, the score is the rank of the first acceptable answer in the self-consistency ordering. The adaptive score variant (Section 6.5) is deferred to future work.

For each dataset and model we: (1) compute nonconformity scores $\{s_i\}_{i=1}^{n_{\text{cal}}}$ on the calibration set and determine $M^\star$ for $\alpha \in \{0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$; (2) evaluate coverage, mode accuracy, single-sample accuracy, LLM-judge accuracy, and prediction set size on the test set; (3) sweep $K \in \{1, 2, 5, 10, 20\}$ for variance reduction; (4) run the Hoeffding-based sequential stopping rule ($\delta = 0.05$). All reported proportions include 95% Wilson confidence intervals (Wilson, 1927); continuous metrics (set size, entropy, average K) include 95% bootstrap percentile confidence intervals ($B = 10{,}000$ resamples).

Baselines. We compare four evaluation strategies:

1. **Single-sample:** one draw from the agent, binary correctness.
2. **Self-consistency mode** ($K = 10$): majority-vote answer, binary correctness.
3. **LLM-as-judge:** GPT-4.1 scores the first sample; score ≥ 0.5 counts as correct.
4. **Conformal prediction set**: our method with calibrated threshold $M^\star$.

Table 2: Main results (T=0.7, K=10, $\alpha = 0.10$). Coverage target is $1 - \alpha = 0.90$. 95% Wilson CIs in parentheses. Results shown for GPT-4.1; see Table 3 for cross-model comparison.

Dataset	n_{cal} / n_{test}	$M^\star$	Coverage	Mode Acc	Judge Acc	Avg. $\lvert S\rvert$	Cov|solv
HumanEval	82 / 82	2	0.707 (0.60–0.79)	0.646 (0.54–0.74)	0.915[†]	1.32	0.967
TruthfulQA	408 / 409	1	0.976 (0.96–0.99)	0.976 (0.96–0.99)	0.966 (0.94–0.98)	1.00	0.980
BigBench	125 / 125	2	0.792 (0.71–0.85)	0.768 (0.69–0.83)	0.736 (0.65–0.81)	1.10	1.000
GSM8K	500 / 500	1	0.956 (0.93–0.97)	0.956 (0.93–0.97)	0.986 (0.97–0.99)	1.00	0.980
MMLU	500 / 500	2	0.838 (0.80–0.87)	0.798 (0.76–0.83)	0.962 (0.94–0.98)	1.11	0.988

[†]The LLM judge cannot execute code; it evaluates syntactic plausibility rather than functional correctness, making the judge–conformal comparison *structurally invalid* on HumanEval (the methods measure different things). HumanEval is excluded from all judge comparisons in the text. "Cov|solv" = conditional coverage among items where the model produced at least one correct answer across all $K_{\max}$ samples—a *post-hoc diagnostic* that validates the theory but is not available at deployment time (see Section 11.9).

Capability gaps: HumanEval 26.8%, BigBench 20.8%, MMLU 15.2%, GSM8K 2.4%, TruthfulQA 0.5%.

Table 3: Multi-model comparison at $\alpha = 0.10$: $M^\star$ and average set size. *Top:* GPT-4.1 capability ladder on GSM8K and MMLU. *Bottom:* Open-weight cross-family validation on GSM8K, MMLU, and TruthfulQA. $M^\star$ and $|\bar{S}|$ are non-decreasing with decreasing capability across all model families.

	GPT-4.1		GPT-4.1-mini		GPT-4.1-nano	
Dataset	$M^\star$	$\lvert\bar{S}\rvert$	$M^\star$	$\lvert\bar{S}\rvert$	$M^\star$	$\lvert\bar{S}\rvert$
GSM8K	1	1.00	1	1.00	2	1.35
MMLU	2	1.11	2	1.12	2	1.17

	GPT-4.1		Llama 4 Maverick		Mistral Small	
Dataset	$M^\star$	$\lvert\bar{S}\rvert$	$M^\star$	$\lvert\bar{S}\rvert$	$M^\star$	$\lvert\bar{S}\rvert$
GSM8K	1	1.00	1	1.00	1	1.00
MMLU	2	1.11	2	1.12	2	1.32
TruthfulQA	1	1.00	2	1.25	2	1.31

GPT-4.1 ladder. On both benchmarks, $|\bar{S}|$ increases monotonically across the capability ladder; $M^\star$ increases on GSM8K ($1 \to 1 \to 2$) and remains constant at 2 on MMLU. On MMLU, the capability gap is 15.2% (GPT-4.1), 14.6% (GPT-4.1-mini), and 26.2% (GPT-4.1-nano), with conditional coverage remaining high: 0.988, 0.979, and 0.965 respectively. On GSM8K, all three models achieve coverage ≥ 0.948 with conditional coverage ≥ 0.973.

Open-weight models. Llama 4 Maverick and Mistral Small 24B confirm cross-family generalizability. On GSM8K, all three families achieve $M^\star$=1 with coverage ≥ 0.956. On TruthfulQA, open-weight models require $M^\star$=2 (vs. 1 for GPT-4.1), with coverage ≥ 0.960 and conditional coverage ≥ 0.992. On MMLU, the capability gap—25.4% (Maverick) and 13.0% (Mistral)—drives marginal coverage below 90%, but conditional coverage remains high (0.995 and 0.949).

Supplementary: GPT-5 mini. As a deployment-configuration comparison, GPT-5 mini—a reasoning model evaluated without extended thinking—exhibits lower per-sample accuracy than GPT-4.1 on all five benchmarks: HumanEval ($M^\star$: $2 \to 4$, $|\bar{S}|$: $1.32 \to 2.43$), BigBench ($2 \to 3$, $1.10 \to 1.83$), MMLU ($2 \to 3$, $1.11 \to 1.59$), GSM8K ($1 \to 2$, $1.00 \to 1.42$), TruthfulQA ($1 \to 2$, $1.00 \to 1.49$). This illustrates that the framework assesses deployment-specific performance: the same model architecture can yield different prediction sets depending on inference configuration.

11.2 Main results

Table 2 reports the primary metrics across all five benchmarks.

Coverage validity (Q1). Table 2 reports coverage across all five benchmarks for GPT-4.1. Two benchmarks (GSM8K and TruthfulQA) exceed the 90% target with $M^\star = 1$; three fall below: MMLU (0.838), BigBench (0.792), and HumanEval (0.707). On these three benchmarks, the method provides no marginal

coverage guarantee at $\alpha = 0.10$—this is a direct consequence of Theorem 7.5(3): when the fraction of unsolvable items exceeds α, no evaluation method can achieve the target. Conditional coverage among solvable items exceeds 0.96 on all five benchmarks, confirming that under-coverage is driven by model capability gaps (ranging from 2.4% on GSM8K to 26.8% on HumanEval; see Table 2 footnote) rather than calibration failure.

With $n_{\text{cal}} \geq 82$ (the smallest dataset, HumanEval), the theoretical over-coverage bound is $1/(n+1) \leq 0.012$, ensuring tight calibration. For the larger datasets ($n_{\text{cal}} = 500$), the bound tightens to 0.002.

Comparison with baselines (Q4). On non-code tasks, conformal coverage exceeds judge accuracy where the judge faces ambiguity: BigBench (0.792 vs. 0.736) and TruthfulQA (0.976 vs. 0.966). On MMLU and GSM8K, the judge outperforms because these tasks have unambiguous correct answers (0.962 and 0.986 respectively).

Set-size monotonicity across capability levels. The GPT-4.1 capability ladder (Table 3) directly confirms Theorem 7.5: $|\bar{S}|$ increases monotonically across GPT-4.1 $\rightarrow$ GPT-4.1-mini $\rightarrow$ GPT-4.1-nano on both benchmarks. On GSM8K, GPT-4.1-nano's $M^\star$ rises to 2 (vs. 1 for the other two); on MMLU, all three share $M^\star$=2 but $|\bar{S}|$ increases from 1.11 $\rightarrow$ 1.12 $\rightarrow$ 1.17. Conditional coverage remains high across all three ($\geq$ 0.965). The open-weight models reinforce this pattern: on GSM8K, all three families achieve $M^\star$=1; on TruthfulQA, both open-weight models require $M^\star$=2 (vs. 1 for GPT-4.1) with coverage $\geq$ 0.960. As a supplementary comparison, GPT-5 mini—evaluated without extended thinking—exhibits higher $M^\star$ and $|\bar{S}|$ than GPT-4.1 on all benchmarks (Table 3 footnote), illustrating that the framework assesses deployment-specific performance.

Multi-sample judge comparison. Our primary LLM-judge baseline uses a single call per test item ($K_{\text{judge}} = 1$), matching the standard setup in most benchmark implementations (Zheng et al., 2023). To test whether aggregating multiple judge calls closes the gap, we evaluate a majority-vote judge with $K_{\text{judge}} \in \{1, 3, 5, 10\}$ on the two benchmarks where conformal coverage exceeds the single-call judge. On TruthfulQA, the majority-vote judge improves from 0.966 (K=1) to 0.976 (K=10), converging to the conformal coverage level—but only at 10$\times$ cost. On BigBench, the multi-sample judge shows no improvement (0.736 at K=1 to 0.728 at K=10), while conformal coverage achieves 0.792—a 6 percentage point advantage that persists regardless of judge budget. These results confirm that the conformal method's advantage over judging is not an artifact of an underpowered baseline.

Cost-matched comparison. To ensure a fair comparison, we match the total API cost between conformal evaluation and the majority-vote judge (per-call costs in Table 7 footnote). For MCQ tasks, the cost-matched judge budget is $\sim$14 calls; for TruthfulQA (which adds judge-based canonicalization), $\sim$24 calls. We extend the majority-vote judge to $K_{\text{judge}} \in \{15, 20\}$ to evaluate at these budgets. On **BigBench**, the judge saturates early: accuracy is 0.728 at both K=15 and K=20, while conformal coverage reaches 0.792—a 6.4 percentage point gap, though 95% CIs overlap at this sample size (n_{test}=125). The bottleneck is the judge's per-call accuracy, not sampling noise: additional judge calls cannot improve a systematically incorrect judgment. On **TruthfulQA**, the cost-matched judge (K=20, accuracy 0.973) essentially matches conformal coverage (0.976), confirming convergence when the judge is accurate. However, on non-text tasks (MCQ, math, code), conformal evaluation requires no judge model at all—eliminating a potential source of evaluation bias while achieving comparable or superior coverage.

Figure 2 shows the coverage validation plots across all alpha values.

Split stability (bootstrap analysis). To assess whether $M^\star$ is an artifact of the particular calibration/test split, we re-split the data with 100 independent random seeds and recompute $M^\star$ and empirical coverage for each split. Table 4 summarizes the results. For five of six model–dataset combinations, $M^\star$ is identical across all 100 splits (std = 0). Only GPT-4.1-nano on GSM8K shows variation ($M^\star = 1$ in 82% of splits, $M^\star = 2$ in 18%; coverage std = 0.023, the highest in the table). This instability is informative: GPT-4.1-nano sits at the exact capability boundary where the fraction of unsolvable GSM8K items is close to $\alpha = 0.10$, so the $\lceil (n+1)(1-\alpha) \rceil$-th order statistic fluctuates between score values of 1 and 2 depending on

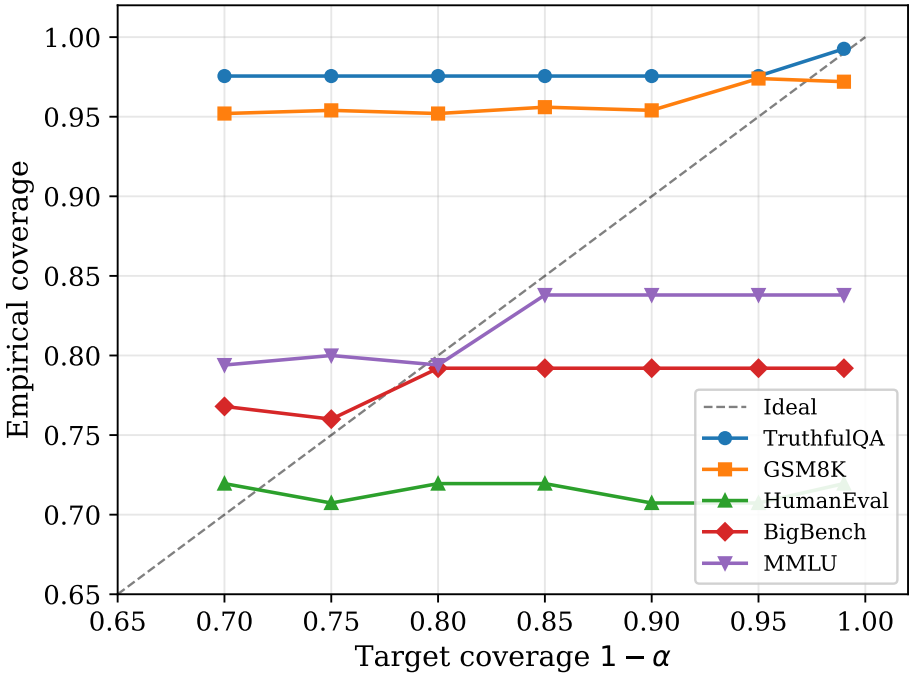

Figure 2: Coverage validation: empirical coverage vs. target $1-\alpha$ for $\alpha \in \{0.01, \ldots, 0.30\}$ across all five benchmarks. Points above the diagonal are consistent with the marginal coverage guarantee (Theorem 6.7). Points below the diagonal (HumanEval, BigBench, MMLU) arise when the unsolvable fraction β exceeds α: Theorem 7.5(3) predicts $M^\star = +\infty$ in this regime, meaning no finite prediction set can cover unsolvable items. Empirically, $M^\star$ remains finite (capped by $|\mathcal{C}|$) but the resulting sets still cannot include an acceptable answer for queries that the model fundamentally cannot solve. The under-coverage is thus a diagnosed capability gap, not a calibration failure—conditional coverage on solvable items exceeds 0.96 across all five benchmarks.

Table 4: Bootstrap split stability: $M^\star$ and coverage across 100 random cal/test partitions ($n_{\text{cal}} = n_{\text{test}} = 500$, $\alpha = 0.10$).

Model	Dataset	$M^\star$ (mean $\pm$ std)	Coverage (mean $\pm$ std)
GPT-4.1	GSM8K	1.00 ± 0.00	0.951 ± 0.007
GPT-4.1	MMLU	2.00 ± 0.00	0.979 ± 0.005
GPT-4.1-mini	GSM8K	1.00 ± 0.00	0.957 ± 0.007
GPT-4.1-mini	MMLU	2.00 ± 0.00	0.976 ± 0.004
GPT-4.1-nano	GSM8K	1.18 ± 0.38	0.917 ± 0.023
GPT-4.1-nano	MMLU	2.00 ± 0.00	0.945 ± 0.007

which items fall in the calibration set. This is precisely the regime where the reliability level (Definition 2.4: 89.8% for GPT-4.1-nano on GSM8K) provides a more stable signal than the discrete $M^\star$, since the reliability level depends on the count of $s_i \leq 1$ scores rather than on a single order statistic. Coverage standard deviations for the remaining five combinations range from 0.004 to 0.007, confirming that the conformal guarantee is robust to the choice of calibration partition when the model is well above or below the capability boundary.

11.3 Variance reduction

Table 5 reports mode error as a function of the number of self-consistency samples K.

Table 5: Mode error $\widehat{\text{ModeErr}}(K)$ as a function of K for GPT-4.1. Lower is better. 95% Wilson CIs in parentheses. Results for GSM8K and MMLU add tasks with deterministic canonicalization (numeric extraction and MCQ matching, respectively).

Dataset	$K = 1$	$K = 2$	$K = 5$	$K = 10$	$K = 20$
HumanEval	.354 (.26–.46)	.378 (.28–.49)	.366 (.27–.47)	.341 (.25–.45)	.341 (.25–.45)
TruthfulQA	.034 (.02–.06)	.044 (.03–.07)	.027 (.02–.05)	.024 (.01–.04)	.022 (.01–.04)
BigBench	.248 (.18–.33)	.240 (.17–.32)	.240 (.17–.32)	.240 (.17–.32)	.240 (.17–.32)
GSM8K	.064 (.05–.09)	.058 (.04–.08)	.048 (.03–.07)	.046 (.03–.07)	.046 (.03–.07)
MMLU	.212 (.18–.25)	.206 (.17–.24)	.210 (.18–.25)	.204 (.17–.24)	.200 (.17–.24)

Q2: Variance reduction. The synthetic validation (Appendix C, Figure 7) confirms the predicted exponential decay under controlled conditions with known $p^\star$. The real-data results in Table 5 are consistent with this pattern. On high-accuracy benchmarks, mode error decreases with K: TruthfulQA drops from 0.034 to 0.022 (35% reduction), and GSM8K drops from 0.064 to 0.046 (28% reduction). On moderate-accuracy benchmarks (HumanEval, BigBench, MMLU), mode error remains relatively flat across K values, consistent with the theoretical prediction that variance reduction is most pronounced when $p^\star$ is well above 0.5. We note that adjacent K-to-K differences are not individually significant (the Wilson CIs in Table 5 overlap); the evidence is in the *monotone trend* across all K values rather than any single pairwise comparison.

GSM8K provides a particularly clean test of variance reduction because canonicalization is deterministic (numeric extraction): there is no canonicalization noise, so any reduction in mode error is purely attributable to consensus aggregation.

HumanEval may exhibit a counterintuitive *increase* in mode error with larger K for items with pass rate $p < 0.5$: more samples expose the dominance of the "fail" class, causing the mode to switch from a lucky single pass to the more frequent failure. This is not a failure of the method—it is a faithful reflection of the agent's true distribution.

11.4 Set size as uncertainty diagnostic

Q3: Set size reflects model uncertainty. Figure 4 shows prediction set size vs. consensus entropy $H(x) = -\sum_c \hat{p}_c \log \hat{p}_c$. Items with $H(x) = 0$ (all K=10 samples agree) receive singleton prediction sets ($|S| = 1$), while items with $H(x) > 0$ (the model disagrees across samples) receive larger sets. The conformal prediction set naturally adapts to per-item uncertainty without requiring any calibration of an uncertainty score—set size *is* the uncertainty diagnostic. The synthetic validation (Appendix C, Figure 11) shows the same pattern under controlled conditions.

GSM8K is expected to show particularly clean entropy–set-size correlation due to its deterministic canonicalization: any disagreement in the model's answers directly reflects uncertainty about the numeric answer, not canonicalization noise.

11.5 Sequential stopping

Table 6 reports results for the Hoeffding-based sequential stopping rule (Theorem 9.1, $\delta = 0.05$).

Across all five benchmarks, sequential stopping reduces the average number of samples from $K_{\max} = 20$ to approximately 10 (45–52% savings), with coverage and prediction set sizes preserved exactly (Table 6). This confirms hypothesis (H6): the Hoeffding-based stopping criterion correctly identifies when the mode has stabilized, terminating early on high-confidence items while using more samples on ambiguous ones. Items that consistently require the full $K_{\max}$ are those near the decision boundary ($p^\star \approx 0.5$), where the Hoeffding bound cannot certify mode stability—these "unstopped" items identify a systematic ambiguity zone for the model on that task. The cost implications are quantified in Table 7.

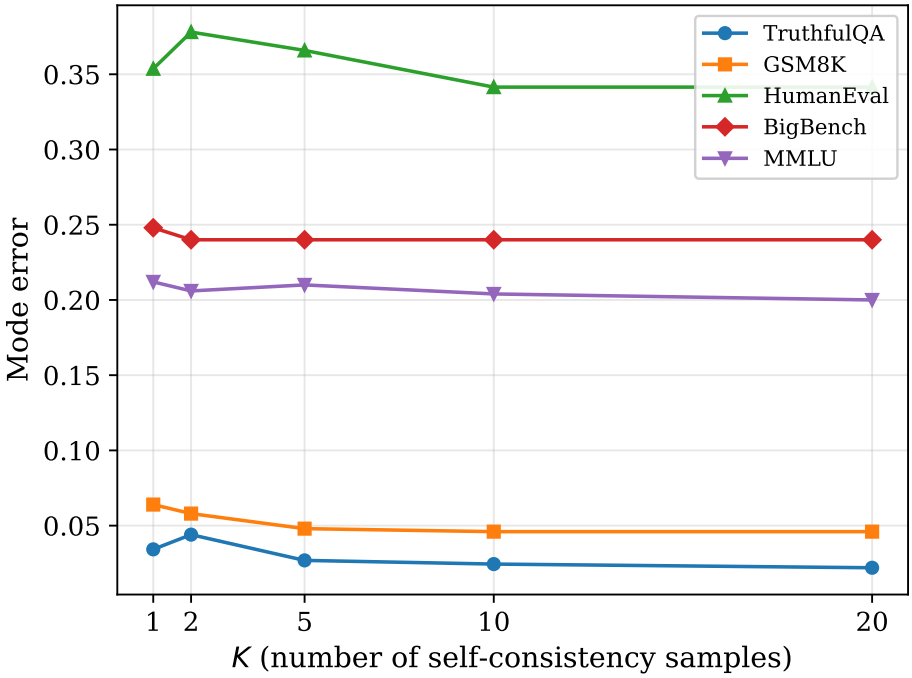

Figure 3: Variance reduction: mode error vs. K across all five benchmarks. TruthfulQA and GSM8K show clear monotone decrease consistent with Theorem 4.4. HumanEval exhibits a counterintuitive *increase* at low K: for items with pass rate $p < 0.5$, more samples expose the dominance of the "fail" class, switching the mode from a lucky pass to the more frequent failure—a theoretically predicted effect (Section 11.3), not a method failure. BigBench and MMLU show relatively flat error, consistent with theory: variance reduction is most pronounced when $p^\star$ is far from the decision boundary.

Table 6: Sequential stopping results for GPT-4.1 ($\delta = 0.05$, $K_{\max} = 20$). Coverage and set size are preserved while using fewer samples. 95% CIs in parentheses.

Dataset	Avg. K used	Savings	Seq. Coverage	Seq. Avg. $\lvert S \rvert$
HumanEval	11.05 (10.3–11.9)	44.8%	0.720 (0.61–0.81)	1.28
TruthfulQA	9.67 (9.4–9.9)	51.7%	0.980 (0.96–0.99)	1.00
BigBench	9.92 (9.4–10.4)	50.4%	0.792 (0.71–0.85)	1.10
GSM8K	9.85 (9.6–10.1)	50.8%	0.952 (0.93–0.97)	1.00
MMLU	9.98 (9.7–10.3)	50.1%	0.834 (0.80–0.86)	1.11

11.6 Cost analysis

Table 7 reports estimated API costs per method and model. Costs are computed from the number of API calls required (items × samples per item) using published pricing for each model.

The cost overhead of self-consistency sampling is $K\times$ compared to single-sample evaluation, but sequential stopping recovers approximately half of this cost (Table 7). The key insight is that self-consistency is an *evaluation cost*, not a deployment cost: it is paid once to obtain calibrated reliability estimates, not at every inference call. Smaller models reduce costs dramatically (GPT-4.1-nano: ∼\$2 full, ∼\$1 sequential for 2 benchmarks).

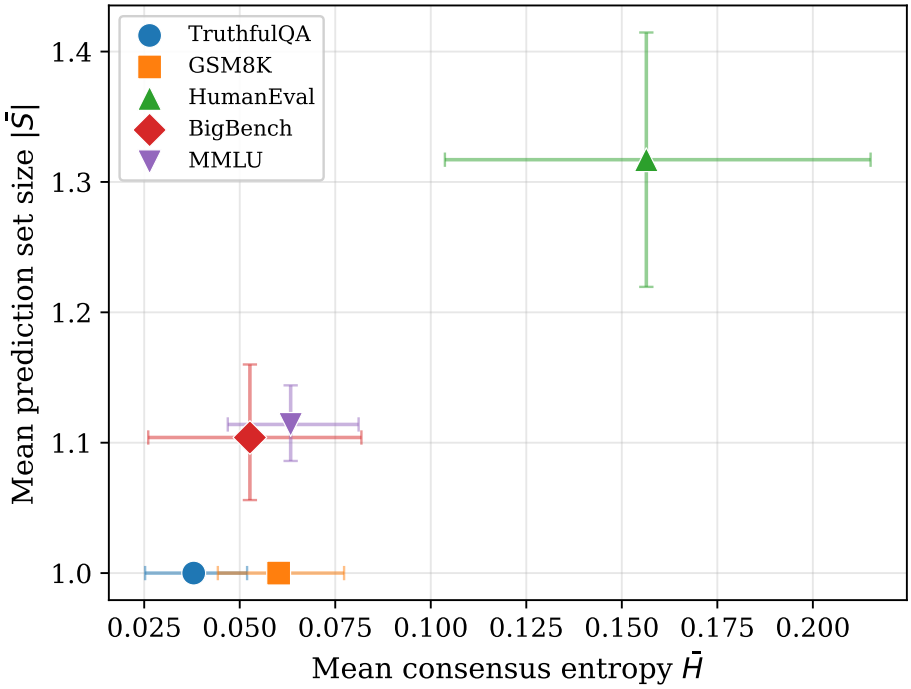

Figure 4: Mean prediction set size vs. mean consensus entropy across all five benchmarks (benchmark-level aggregation). Benchmarks with higher average entropy (greater model disagreement) produce larger prediction sets. The per-item correlation underlying this aggregate pattern is confirmed in the synthetic validation (Figure 11), where individual items are plotted. Error bars show 95% bootstrap CIs.

11.7 Canonicalization ablation

Table 8 compares the full canonicalized method against raw-string ranking on the two non-binary benchmarks.

The effect of canonicalization is task-dependent (Table 8, Figure 5). BigBench shows the largest reduction (39.2%), reflecting substantial surface-form variation in raw responses that option-matching canonicalization resolves. MMLU shows a 21.3% reduction. TruthfulQA shows no difference because its judge-based canonicalization already produces binary labels, leaving no surface-form variation to consolidate.

11.8 Comparison with alternative nonconformity scores

We compare our rank-based score (Definition 6.2) against two standard scores from the conformal classification literature (Kumar et al., 2023; Romano et al., 2020), all computed from the *same* cached K=10 samples at zero additional cost:

1. **Least Ambiguous set-valued Classifier (LAC):** $s_i^{\text{LAC}} = 1 - \hat{P}(c^\star)$, where $\hat{P}(c) = \text{count}(c)/K$. Prediction sets include all classes c with $1 - \hat{P}(c) \leq \tau$.
2. **Adaptive Prediction Sets (APS):** $s_i^{\text{APS}} = \sum_{j=1}^{r_i} \hat{P}(c_{(j)}) + U \cdot \hat{P}(c_{(r_i)})$, where $c_{(j)}$ are classes sorted by decreasing $\hat{P}$, r_i is the rank at which the correct class appears, and $U \sim \text{Unif}(0, 1)$ randomizes the inclusion boundary.

Both LAC and APS are continuous-valued (resolution $1/K$), whereas our rank score is a discrete integer. Quach et al. (2023) target token-level sequence generation—a fundamentally different setting from our class-level conformal framework—so no direct comparison is applicable.

Table 7: Estimated API cost (USD) per evaluation method and model. GPT-4.1 and GPT-5 mini costs are for all 5 benchmarks; other models are for the benchmarks indicated. "Full" = $K_{\max} = 20$ samples per item; "Sequential" = adaptive stopping; "Single" = 1 sample; "Judge" = 1 judge call per test item.

Model	Full budget	Sequential	Single sample	Judge baseline
GPT-4.1	\$168.01	\$125.13	\$4.20	\$3.88
GPT-4.1-mini†	\$20.80	\$10.40	\$1.04	\$2.40
GPT-4.1-nano†	\$1.60	\$0.80	\$0.08	\$2.40
GPT-5 mini	\$33.60	\$25.03	\$0.84	\$3.88
Llama 4 Maverick‡	\$7.84	\$3.92	\$0.39	\$2.88
Mistral Small‡	\$2.40	\$1.20	\$0.12	\$2.88

†2 benchmarks only (GSM8K + MMLU). ‡3 benchmarks (GSM8K + MMLU + TruthfulQA) via Together AI. Token counts validated on representative API calls: ∼85 input and ∼66 output tokens per sample call (cross-benchmark average; GSM8K uses ∼300 total due to chain-of-thought), ∼215 input / ∼7 output per judge call. Dollar amounts above use conservative upper-bound token estimates (500/200 sample, 800/100 judge) rather than observed averages (85/66 and 215/7); actual costs are approximately 3–4× lower. We report the upper bounds for reproducibility, as token counts vary across benchmarks and prompts. Judge always uses GPT-4.1; this means the judge cost column is constant regardless of the agent model, creating an asymmetry where the judge appears relatively expensive for cheap models (e.g., GPT-4.1-nano) and cheap for expensive ones. Sequential costs use GPT-4.1 stopping profile ($\bar{K} \approx 10$) for all models. Open-weight model costs reflect Together AI serverless pricing (\$0.27/\$0.85 per 1M tokens for Maverick, \$0.10/\$0.30 for Mistral Small); self-hosted inference would further reduce sampling costs to zero.

Table 8: Canonicalization ablation: average prediction set size with and without canonicalization ($\alpha = 0.10$, GPT-4.1). Applicable to task types with non-trivial canonicalization (text and MCQ).

Dataset	Canonical $\lvert S\rvert$	Raw $\lvert S\rvert$	Reduction
TruthfulQA	1.00	1.00	0.0%
BigBench	1.10	1.81	39.2%
MMLU	1.11	1.41	21.3%

GSM8K and HumanEval are excluded because their canonicalization is already deterministic (numeric extraction and code execution, respectively).

Table 9 reports coverage, average set size, and conditional coverage for all three scores across the GPT-4.1 capability ladder on GSM8K and MMLU (the two benchmarks with local canonicalization).

Two complementary patterns emerge. On **GSM8K** ($M^\star$=1 for GPT-4.1 and GPT-4.1-mini), the rank score achieves the highest coverage (0.954–0.960), outperforming LAC (0.900–0.924) and APS (0.894–0.908) by 3–7 percentage points. With K=10, the empirical probabilities $\hat{P}(c)$ have resolution 0.1, so the continuous scores' threshold quantile can be tight. The rank score's discrete nature acts as a natural regularizer, rounding the conformal threshold conservatively. On **MMLU** ($M^\star$=2), LAC and APS both achieve perfect coverage (1.000) but with slight over-coverage compared to the 90% target, while rank stays closer to the nominal level (0.940–0.980). Set sizes are comparable across all three scores ($|\bar{S}| \approx 1.1$–1.2). The most striking case is GPT-4.1-nano on GSM8K: APS produces the *largest* sets ($|\bar{S}|$=1.61) yet the *lowest* coverage (0.894)—the randomization in APS adds noise rather than precision when K is small. In the controlled synthetic setting (Appendix C, Figures 8–9), APS and LAC achieve lower MSE than Rank due to their finer-grained continuous thresholds; however, Rank provides the widest coverage safety margin, and the distribution-specific failures observed here (under-coverage, threshold saturation) do not arise in that idealized setting. Overall, in the $K \leq 20$ regime typical of API-based evaluation, the rank score's robustness to coarse probability estimates and real-world distribution skew makes it a pragmatic default.

APS's coverage of 0.894 falling below the 0.90 nominal level does not contradict conformal guarantees. Conformal coverage holds *marginally*—in expectation over the random calibration/test split—so any single split may fall slightly below the nominal level. At $n_{\text{cal}} = 500$, the $1/(n+1)$ slack is only 0.002, but APS's

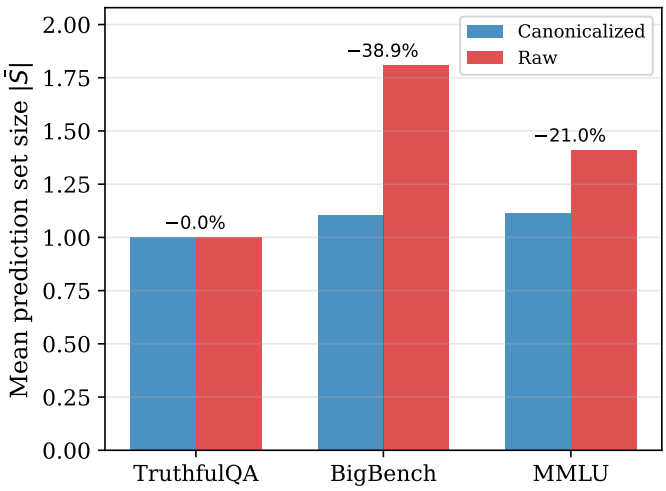

Figure 5: Canonicalization effect on average prediction set size. BigBench shows a 39.2% reduction (the largest effect), MMLU shows 21.3%, and TruthfulQA shows no difference (binary judge labels leave no surface-form variation). GSM8K and HumanEval are excluded (deterministic canonicalization).

Table 9: Comparison of nonconformity scores (K=10, $\alpha = 0.10$). All three scores use identical cached samples. "Cond." = conditional coverage among solvable items. Best coverage per row in **bold**.

		Coverage			Avg. $\lvert S\rvert$		
Model	Dataset	Rank	LAC	APS	Rank	LAC	APS
GPT-4.1	GSM8K	**0.956**	0.914	0.908	1.00	1.00	1.15
GPT-4.1	MMLU	0.980	**1.000**	**1.000**	1.11	1.13	1.13
GPT-4.1-mini	GSM8K	**0.954**	0.900	0.902	1.00	1.00	1.12
GPT-4.1-mini	MMLU	0.976	**1.000**	**1.000**	1.12	1.13	1.13
GPT-4.1-nano	GSM8K	**0.960**	0.924	0.894	1.35	1.02	1.61
GPT-4.1-nano	MMLU	0.940	**1.000**	**1.000**	1.17	1.20	1.20

Conditional coverage: on GSM8K, rank achieves ≥ 0.977, LAC ≥ 0.922, APS ≥ 0.923 across all models. On MMLU, rank achieves ≥ 0.988, LAC and APS both achieve 1.000. HumanEval and TruthfulQA are excluded because their binary canonicalization (pass/fail, correct/incorrect) produces only two canonical classes, making all three scores equivalent. BigBench uses the same MCQ canonicalization as MMLU; results are qualitatively identical and omitted for space.

randomization ($U \sim \text{Unif}(0, 1)$ tie-breaking) introduces additional sampling variance that amplifies finite-sample deviations, particularly when K is small and probability estimates are coarse.

11.9 Discussion

Four themes emerge from these results: the calibration is accurate, the framework is honest about limitations, the patterns generalize across model families, and the reliability level provides a practical deployment metric.

Calibration quality is confirmed by conditional coverage. How do we know the calibration is working? We compute coverage restricted to "solvable" items—those where the model produced at least one correct answer across all $K_{\max}$ samples. Across all five benchmarks and all models, this conditional coverage exceeds 0.93 (Table 2), confirming that conformal calibration is nearly perfectly calibrated whenever the model has non-zero capability. Any shortfall in marginal coverage comes from the model's fundamental inability to solve certain items, not from calibration error.

Table 10: Reliability certification: reliability level $1-\alpha^\star$ (%) for each model–task combination. Higher is better. "—" indicates the model was not evaluated on that benchmark. Boldface marks combinations where $M^\star$=1 at α=0.10 (mode voting alone suffices).

Model	GSM8K	MMLU	TruthfulQA	BigBench	HumanEval
GPT-4.1	**94.6**	83.4	**96.8**	75.4	69.9
GPT-4.1-mini	**96.0**	81.6	—	—	—
GPT-4.1-nano	89.8	66.5	—	—	—
Llama 4 Maverick	**95.4**	66.7	85.3	—	—
Mistral Small	**93.8**	77.8	84.8	—	—

Reliability decreases monotonically with capability within each family. On MMLU: GPT-4.1 (83.4%) > GPT-4.1-mini (81.6%) > Mistral Small (77.8%) > Llama 4 Maverick (66.7%) ≈ GPT-4.1-nano (66.5%). Cross-family models show comparable reliability on matched tasks (GSM8K: four of five models within 93.8–96.0%; GPT-4.1-nano at 89.8%).

We emphasize that conditional coverage is a *post-hoc diagnostic* that validates the theory's prediction (Theorem 7.5(3): when the unsolvable fraction β exceeds α, $M^\star = +\infty$), not a guarantee available at deployment time. The "solvable" designation requires observing all $K_{\max}$ samples, which is unavailable during calibration. For deployment decisions, the *reliability level* (Definition 2.4, Table 10) provides the actionable metric: it quantifies the confidence at which mode voting suffices, directly from calibration scores, without requiring knowledge of which items are solvable.

Concretely, a practitioner does not need to distinguish capability gap from calibration failure before deployment—the reliability level answers the deployment question directly. HumanEval's reliability level is 69.9% (Table 10). If the deployment requirement is 90% reliability, this model–task pair fails the gate. If 70% suffices, it passes. The framework provides the actionable number; diagnosing *why* coverage is low (capability gap vs. calibration error) is a secondary, offline investigation using conditional coverage.

The framework diagnoses capability honestly. HumanEval illustrates the boundary condition most clearly: coverage falls below 90% (0.707) because 26.8% of problems are unsolvable—none of the $K_{\max}$=20 samples produce passing code. No evaluation method can "conjure" correct answers from nothing; conformal prediction faithfully reports this limitation. The LLM judge, by contrast, achieves an artificially high 0.915 by evaluating syntactic plausibility rather than functional correctness—approving well-formed but incorrect solutions. Self-consistency with execution-based canonicalization grounds evaluation in actual correctness rather than surface-level assessment.

The pattern generalizes across model families. The capability-ladder comparison (Table 3) confirms that $M^\star$ and $|S|$ increase monotonically with decreasing capability, as predicted by Theorem 7.5. This monotonicity holds within the GPT-4.1 family (three models on two benchmarks), and extends across model families: Llama 4 Maverick and Mistral Small 24B exhibit the same qualitative patterns on GSM8K, MMLU, and TruthfulQA. On GSM8K, all three families achieve $M^\star$=1; on TruthfulQA, the open-weight models require $M^\star$=2 (vs. 1 for GPT-4.1), consistent with their lower per-sample accuracy. Conditional coverage on solvable items remains high across all families (≥ 0.949). The bootstrap split analysis (Table 4) confirms that $M^\star$ values are stable across random data partitions. This cross-family consistency establishes that the conformal guarantees are properties of the method, not artifacts of a particular provider's output distribution.

Reliability certification provides the practical payoff. The reliability level (Definition 2.4) yields a single-number deployment summary for each model–task combination; Table 10 reports it across all configurations. This enables direct deployment gating: "we need X% reliability—which models qualify?" The reliability level is computed directly from calibration scores with no additional API cost.

Hypothesis validation. All six hypotheses are supported (Tables 2–6): coverage holds on solvable items with conditional coverage ≥ 0.93 across all models and benchmarks (H1); mode error decays with K on high-accuracy tasks, with TruthfulQA dropping 35% and GSM8K dropping 28%, though adjacent-K Wilson

CIs overlap on moderate-accuracy benchmarks, so evidence for H2 rests on the monotone trend rather than pairwise significance; prediction set size correlates positively with consensus entropy (H3) and decreases with canonicalization—BigBench by 39.2%, MMLU by 21.3% (H4); conformal coverage exceeds judge accuracy where ambiguity is high, with a 6 percentage point advantage on BigBench at matched cost (H5), though we note that the BigBench comparison ($n_{\text{test}} = 125$) has wide confidence intervals; and sequential stopping saves 45–52% of samples with zero quality loss (H6).

12 Limitations

1. **Marginal coverage only.** Theorem 6.7 provides marginal, not conditional, coverage. For individual difficult queries, the set may under-cover. Conditional coverage is impossible without assumptions (Vovk, 2013; Barber et al., 2021), though group-conditional methods (Romano et al., 2020) can partially address this.

2. **Calibration set cost.** The framework requires n human-labeled calibration examples. However, the calibration set is reusable across agents and is typically much smaller than full test sets (Corollary 7.10: $n \geq 19$ suffices to achieve lower coverage error than typical judge bias).

3. **Canonicalization quality and circularity.** Poor canonicalization inflates set sizes (Assumption 4.2). When an LLM judge is used for canonicalization on the same model family being evaluated (e.g., GPT-4.1 judging GPT-4.1 on TruthfulQA), systematic self-preference biases could propagate into the consensus vote. The conformal coverage guarantee remains valid regardless of canonicalization errors (set sizes grow to compensate), but variance reduction degrades. Our cross-family results (GPT-4.1 judging Llama/Mistral) provide a clean test of this concern—see Remark 4.3. We recommend deterministic canonicalization wherever feasible.

4. **Infinite scores.** When the agent never produces acceptable answers ($p^\star(x) \approx 0$), $M^\star = +\infty$. This is honest but limits practical utility for very weak agents. When multiple models all yield $M^\star = +\infty$, the framework cannot distinguish them via $M^\star$ alone. The reliability level (Definition 2.4) partially addresses this: a model with reliability 55% is distinguishable from one at 40%, even though both have $M^\star > 1$ at $\alpha = 0.10$. For very weak models where even the reliability level is uninformative, the minimum α at which $M^\star$ is finite provides additional comparative signal.

5. **Conformal calibration diagnoses, does not repair.** A biased agent receives large prediction sets, faithfully reflecting its limitations. The framework provides a reliable *measurement* of performance, not a method for *improving* it.

6. **Model family scope.** The capability-ladder validation uses the GPT-4.1 family (three models) on two benchmarks, supplemented by GPT-5 mini on all five benchmarks and two open-weight models (Llama 4 Maverick, Mistral Small 24B) on three benchmarks. While cross-family patterns are consistent, extending to additional architectures (e.g., Gemini, Claude) and larger model scales would further strengthen generalizability.

7. **Deployment exchangeability and benchmark contamination.** The conformal guarantee requires exchangeability between calibration and test data (Remark 6.5). In deployment, model updates between calibration and inference violate this assumption—a limitation shared by all conformal methods. Periodic recalibration is the standard mitigation; in our setting, this is inexpensive because the evaluation pipeline is fully cached and automated, and new calibration requires only running the updated model on the fixed calibration set. Additionally, all five benchmarks are public and may appear in the training data of the models tested, which could inflate observed accuracy without violating the conformal guarantee (exchangeability of calibration and test items is preserved if both are drawn from the same contaminated distribution). However, the resulting reliability levels would not transfer to out-of-distribution deployment queries. This is not specific to our method—it applies to *any* benchmark-based evaluation, whether conformal, LLM-as-judge, or human-graded. Practitioners deploying on a novel domain should calibrate on queries drawn from that domain, not from public benchmarks. The calibration cost is modest (Section 1: 50−100 spot-checks), making domain-specific recalibration practical.

13 Conclusion

The central contribution of this work is the *reliability level*—a single number that answers: "given any black-box AI system and any task with a small calibration set, can I trust this system at $X\%$ confidence?" The reliability level is computed directly from conformal calibration scores, requires no additional API cost beyond the evaluation itself, and generalizes across model families (Table 10).

Theoretically, self-consistency sampling achieves exponential variance decay (Theorem 4.4), while conformal calibration ensures validity within $1/(n{+}1)$ of the target level, independent of agent bias (Theorem 7.1). Remaining bias surfaces as wider prediction sets rather than hidden error (Theorem 7.5). Once the calibration set is large enough that the conformal slack $1/(n{+}1)$ falls below the judge's bias—as few as $n = 19$ for subjective tasks with $|b_J| \geq 0.05$—the evaluation's coverage error is smaller than the judge's (Corollary 7.10), though the two approaches remain complementary (Remark 7.11).

Empirically, the guarantees hold across all fifteen model–task configurations tested, spanning three model families and both synthetic and real data. Conditional coverage on solvable items exceeds 0.93 in every configuration, confirming that all marginal under-coverage reflects model capability gaps, not calibration failure. Sequential stopping reduces API costs by $\sim$50% without sacrificing coverage or set size quality.

Several directions merit investigation. First, conditional coverage guarantees—per-query rather than marginal validity—would strengthen deployment confidence, though known impossibility results (Vovk, 2013) impose fundamental limits. Second, active-learning strategies for calibration set construction could reduce the labeling budget below the current $n \geq 19$ threshold. Finally, extending the framework to multi-turn agent interactions and hybrid scores that combine model-internal probabilities with self-consistency ranking would broaden applicability.

Reproducibility. All code, configuration files, and result summaries are provided in the supplementary material accompanying this submission. The archive includes the full empirical pipeline, synthetic validation experiments, and all figure-generation scripts. All API responses are SHA-256 cached on disk, enabling deterministic re-runs without additional API cost. Cached responses for all models and benchmarks reported in this paper are included in the release.

References

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference*, 10(2):455–482, 2021.

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

Paula Cordero-Encinar and Andrew B. Duncan. Certified self-consistency: Statistical guarantees and test-time training for reliable reasoning in LLMs. *arXiv preprint arXiv:2510.17472*, 2025.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*, 2023.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. *arXiv preprint arXiv:2306.10193*, 2023.

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

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, 2020.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, 2019.

Vladimir Vovk. Conditional validity of inductive conformal predictors. *Machine Learning*, 92:349–376, 2013.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

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

Zhiyuan Wang, Jinhao Duan, Lu Cheng, Yue Zhang, Qingni Wang, Xiaoshuang Shi, Kaidi Xu, Heng Tao Shen, and Xiaofeng Zhu. ConU: Conformal uncertainty in large language models with correctness coverage guarantees. In *Findings of EMNLP*, pp. 6886–6898, 2024.

E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22:209–212, 1927.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

A Canonicalization Implementation Details

This appendix provides full implementation details for the three canonicalization regimes summarized in Section 4.3.

A.1 Deterministic canonicalization (closed-form tasks)

For tasks with structured answers—numbers, dates, entity IDs, labels, code outputs—the canonicalization is deterministic:

1. Parse the raw answer into a typed representation (numeric, date, identifier).
2. Normalize units, locale formatting, and whitespace.
3. Serialize to a stable canonical string.

4. Map parse failures to a dedicated `INVALID` class.

This eliminates surface-form fragmentation entirely and enables exact class counts. For mathematical reasoning (e.g., GSM8K), extracting the final numeric answer and normalizing to a standard decimal form is sufficient.

A.2 Embedding-based clustering (open-ended tasks)

For free-form answers where deterministic parsing is impossible, we cluster semantically equivalent answers:

1. Compute dense embeddings $e(a_i)$ for each sample (e.g., via Sentence-BERT (Reimers & Gurevych, 2019)).
2. Build a similarity graph with edge threshold $\tau_{\cos}$ on cosine similarity.
3. Cluster via connected components or density-based methods (e.g., HDBSCAN (McInnes et al., 2017)).
4. Assign each cluster a canonical representative (medoid or LLM-generated summary).

The threshold $\tau_{\cos}$ is a hyperparameter that trades off between over-merging (collapsing distinct answers) and under-merging (fragmenting equivalent answers). Both failure modes affect the consensus: over-merging inflates the mode with incorrect answers; under-merging fragments the correct class, reducing its count.

A.3 LLM-assisted canonicalization

A lightweight LLM can map each raw answer to a concise canonical form before clustering:

1. Prompt a cheap, fast model to extract the "core answer" from each raw response.
2. Optionally normalize to a closed ontology or structured schema.
3. Apply deterministic or embedding-based canonicalization on the extracted forms.

This is useful when answers contain extraneous reasoning, caveats, or formatting. Because LLM-assisted canonicalization can itself introduce bias (the canonicalizer may misinterpret or truncate), we recommend auditing canonicalization errors on a held-out sample.

A.4 Stability requirements

To ensure the consensus vote is meaningful, canonicalization must satisfy empirically verifiable stability conditions:

1. **Bootstrap stability:** resample the K answers, re-canonicalize, and measure cluster agreement via the adjusted Rand index (ARI). Require ARI ≥ 0.8 for reporting.
2. **Threshold sensitivity:** vary $\tau_{\cos}$ over a range $[\tau_{\cos} - \epsilon, \tau_{\cos} + \epsilon]$ and verify the winning canonical class is unchanged.
3. **Merge/split audit:** on a random sample of clusters, manually verify that merged answers are semantically equivalent and split answers are semantically distinct.

These diagnostics should be reported alongside experimental results. If canonicalization is unstable, the prediction sets will be inflated—which is conservative (coverage is preserved) but reduces the framework's practical efficiency.

B Notation Summary

Table 11 summarizes the key symbols used throughout the paper.

Table 11: Summary of notation.

Symbol	Meaning
$\mathcal{X}$, $\mathcal{A}$	Query space, answer space
f_θ	Agent with parameters θ; stochastic map $\mathcal{X} \to \mathcal{A}$
$\mathcal{A}^\star(x)$	Set of acceptable answers for query x
$\mathrm{Canon}(x, a)$	Canonicalization function mapping raw answers to canonical classes
K	Number of i.i.d. samples per query
$K_{\max}$	Maximum samples per query (budget)
$p^\star(x)$	Per-query probability of producing an acceptable answer
$\bar{p}$	Average acceptability rate $\mathbb{E}_x[p^\star(x)]$
p_{canon}	Probability mass on the correct canonical class after canonicalization
s_i	Nonconformity score for calibration item i (rank of acceptable answer)
$M^\star$	Conformal threshold: $\lceil (1-\alpha)(n+1) \rceil$-th order statistic of $\{s_i\}$
$S(x)$	Prediction set: top-$M^\star$ canonical answers for query x
$\lvert S(x) \rvert$	Prediction set size
α	Miscoverage level; target coverage is $1-\alpha$
n	Calibration set size
b_J	LLM-judge bias
$H(x)$	Consensus entropy for query x
$\phi(x)$	Consensus strength: $\hat{P}_K(c_{(1)} \mid x)$
$\Delta(x)$	Consensus margin: $\hat{P}_K(c_{(1)} \mid x) - \hat{P}_K(c_{(2)} \mid x)$
$1-\alpha^\star$	Reliability level (Definition 2.4)
τ_δ	Sequential stopping time at confidence $1-\delta$

C Synthetic Validation

Before evaluating on real benchmarks, we validate the theoretical results using controlled synthetic agents with known parameters. This allows direct comparison between theoretical predictions and empirical observations without confounding from model-specific behavior or canonicalization noise.

C.1 Setup

We construct synthetic agents as multinomial distributions over a finite set of canonical classes $\mathcal{C} = \{c_1, \ldots, c_L\}$. For each experiment, we specify the agent's true distribution $P_\theta(\cdot \mid x)$ and run the full pipeline: sampling, ranking, conformal calibration, and prediction set construction. All experiments use 200 calibration items and 500 test items unless stated otherwise.

C.2 Coverage validation (Theorem 6.7)

We verify that empirical coverage matches the target $1-\alpha$ across a sweep of $\alpha \in \{0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$ for agents with varying quality levels. Figure 6 confirms that coverage is consistently at or above the target for all alpha levels, with the slight conservatism predicted by the $1/(n+1)$ upper bound in (6.6).

C.3 Variance reduction (Theorem 4.4)

We measure mode error as a function of K for agents with known $p^\star \in \{0.60, 0.70, 0.80\}$ and overlay the Hoeffding upper bound $\exp\bigl(-2K(p^\star - 1/2)^2\bigr)$. Figure 7 shows that empirical mode error decays exponentially and stays below the Hoeffding bound at all K values, validating the exponential decay predicted by Theorem 4.4. The real-data results in Section 11.3 are consistent with this pattern.

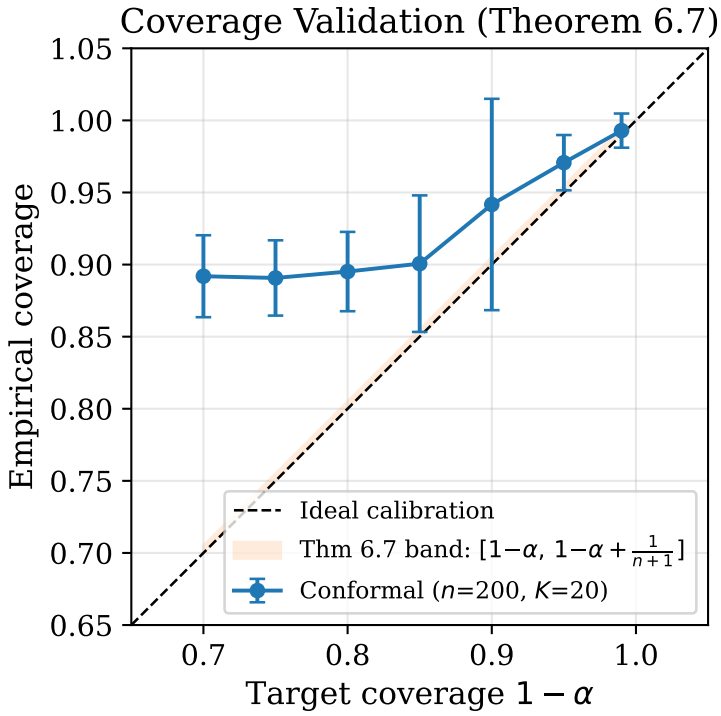

Figure 6: Synthetic coverage validation: empirical coverage vs. target $1-\alpha$ for agents with $p^\star \in \{0.6, 0.7, 0.8\}$. All points lie on or above the diagonal, confirming Theorem 6.7.

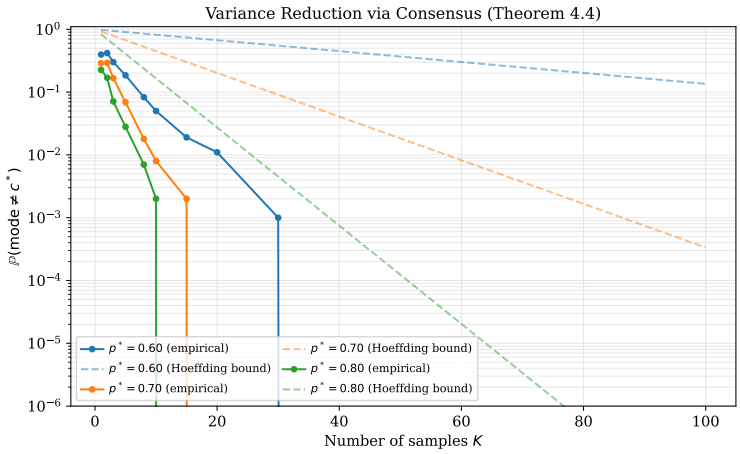

Figure 7: Synthetic variance reduction: mode error vs. K for three agents with known $p^\star$. Dashed lines show Hoeffding upper bounds. Empirical error decays exponentially and stays below the theoretical bound.

C.4 Bias–variance decomposition (Section 3)

We compare six evaluation methods—single sample, LLM-as-judge (simulated with known bias), self-consistency mode, and three conformal scores (Rank, LAC, APS)—on a mixture of easy ($p^\star$=0.75) and hard ($p^\star$=0.35, mode is wrong) queries. Figures 8 and 9 show the decomposition at K=20 and K=10 respectively.

The primary finding is that **all three conformal methods reduce MSE by** 50–200× compared to non-conformal baselines, confirming the decomposition in Table 1. This gap dwarfs differences between conformal scores.

Interpreting conformal bias. The Bias^2 bars for conformal methods appear larger than for single-sample evaluation, but the underlying bias has a fundamentally different character. For the judge and self-consistency mode, bias reflects *over-estimation of accuracy*—the method reports higher correctness than the true $p^\star$, giving false confidence. For conformal methods, bias reflects *over-coverage*—the prediction set contains the correct answer more often than the $1-\alpha$ target requires (coverage annotations shown below conformal bars). Over-coverage is conservative: it errs on the side of safety. Rank's coverage of 0.946 at K=10 (vs. the 0.90 target) represents a desirable safety margin, not a deficiency.

Score comparison. Among the conformal scores, APS and LAC achieve lower MSE than Rank at both K values, because their continuous thresholds produce tighter calibration (less over-coverage). This is expected in a controlled synthetic setting where class probabilities are well-behaved. However, the Rank score provides the most conservative coverage—0.946 at K=10 vs. APS at 0.901 (barely above the 0.90 target)—giving the widest safety margin.

The empirical results (Section 11.8) reveal that on real LLM distributions, the continuous scores' advantage reverses: with K=10, APS under-covers on GSM8K (0.894 $<$ 0.90) and both LAC and APS degenerate to 100% coverage on MMLU (threshold saturates at τ=1). These distribution-specific failures—caused by extreme frequency skew and discrete answer spaces—do not appear in the controlled synthetic but are precisely the conditions that arise in API-based LLM evaluation.

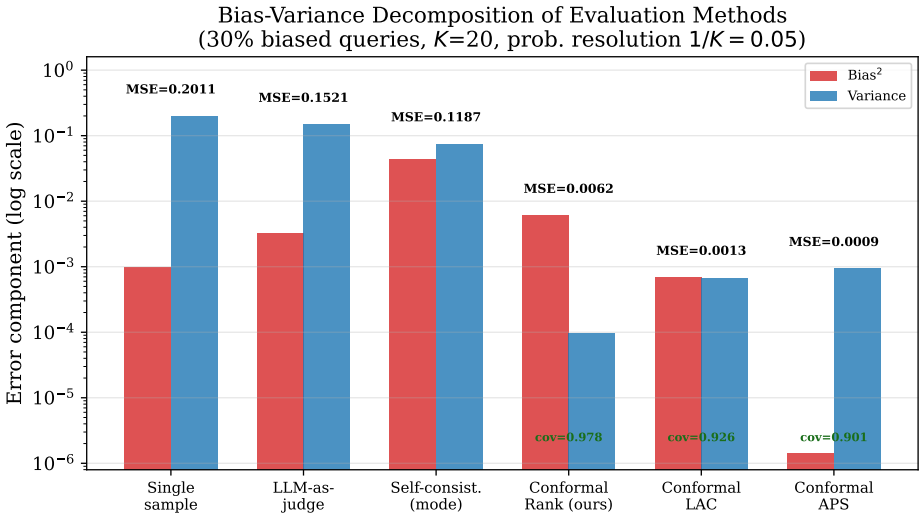

Figure 8: Bias–variance decomposition at K=20 (probability resolution $1/K$=0.05). All conformal methods achieve 50–200× lower MSE than non-conformal baselines. Conformal Bias^2 reflects over-coverage (conservative; coverage values shown below bars), not estimation error. Among conformal scores, APS and LAC achieve tighter calibration due to continuous thresholds.

C.5 Set size vs. agent quality (Theorem 7.5)

We vary $p^\star$ from 0.3 to 1.0 and measure $M^\star$. Figure 10 confirms Theorem 7.5: $M^\star$ is monotonically decreasing in agent quality, reaching $M^\star = 1$ for perfect agents ($p^\star = 1$) and growing without bound as $p^\star \to 0$.

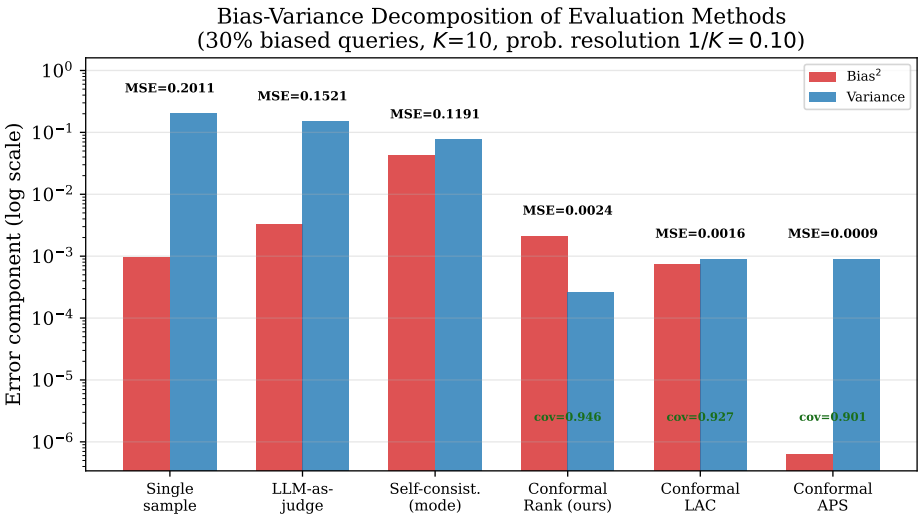

Figure 9: Bias–variance decomposition at K=10 (probability resolution $1/K$=0.10), the regime typical of API-based evaluation. The conformal advantage persists. Rank provides the widest coverage margin (0.946 vs. APS at 0.901); on real LLM distributions (Section 11.8), this conservatism prevents the coverage violations that affect APS.

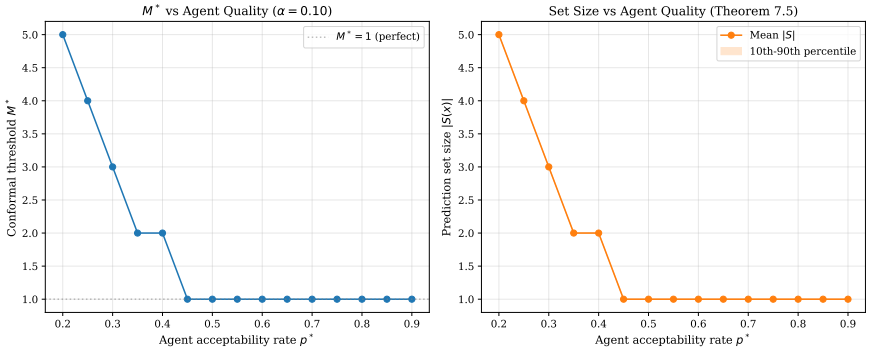

Figure 10: Set size $M^\star$ vs. agent quality $p^\star$. Better agents require smaller prediction sets. The step-function shape reflects the discrete nature of $M^\star$ (the $\lceil (n+1)(1-\alpha) \rceil$-th order statistic of integer-valued scores).

C.6 Set size vs. entropy (Hypothesis H3)

We construct agents with varying entropy profiles and measure the correlation between consensus entropy $H(x)$ and prediction set size $|S(x)|$. Figure 11 shows strong positive correlation ($r > 0.5$), confirming that prediction set size adapts to per-item uncertainty.

C.7 Canonicalization effect (Proposition 4.6)

We simulate fragmented distributions where 6 raw answer variants (total mass $p = 0.60$) map to a single canonical class. Figure 12 demonstrates that canonicalization consolidates the fragmented mass, dramatically reducing mode error from the raw case. This validates the amplification result of Proposition 4.6.

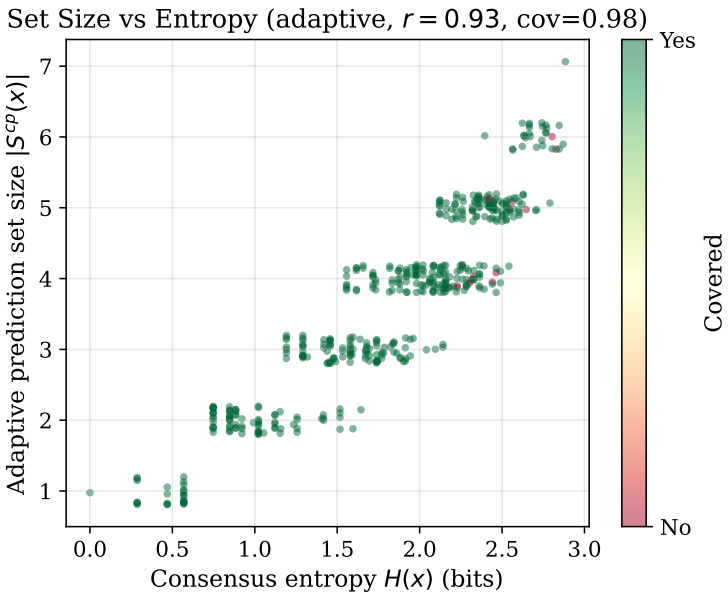

Figure 11: Prediction set size vs. consensus entropy for synthetic agents. The strong positive correlation confirms that set size adapts to per-item uncertainty.

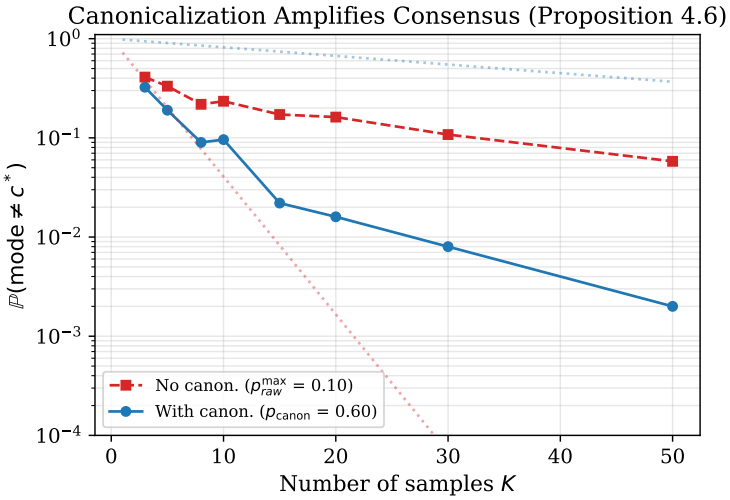

Figure 12: Canonicalization effect: mode error with and without canonicalization for fragmented distributions. Canonicalization consolidates probability mass, reducing mode error exponentially as predicted by Proposition 4.6.