When Models Hesitate: Answer Instability as a Label-Free Uncertainty Signal for LLMs

Jasper Meynard P. Arana[1,2], Kristine Ann M. Carandang[1,2], Ethan Robert A. Casin[1,2], Christian M. Alis[1,2], Christopher P. Monterola[1,2]

[1] Asian Institute of Management, Makati, Philippines,
[2] Analytics, Computing and Complex Systems Laboratory (ACCeSs@AIM),

jarana.PhDinDS2027@aim.edu

Abstract

Large language models (LLMs) are increasingly deployed in high-stakes settings, yet reliably estimating when their outputs should be trusted remains an open challenge. Existing uncertainty estimation approaches—such as calibration, token-level probabilities, or semantic entropy—typically require access to model internals, additional supervision, or computationally intensive pipelines. We propose answer instability, defined as the variability of a model's final answer across repeated stochastic generations of the same prompt, as a simple, label-free, and black-box uncertainty signal. Evaluated across three task types — reasoning, multiple-choice QA, and constraint-following — using four LLMs and 520 prompt-model pairs, our approach achieves performance competitive with semantic entropy while requiring no semantic similarity model. Our results show that instability strongly correlates with prediction errors and reliably discriminates correct from incorrect outputs. We further demonstrate its utility for selective prediction and targeted repair, improving reliability without access to internal probabilities or additional training.

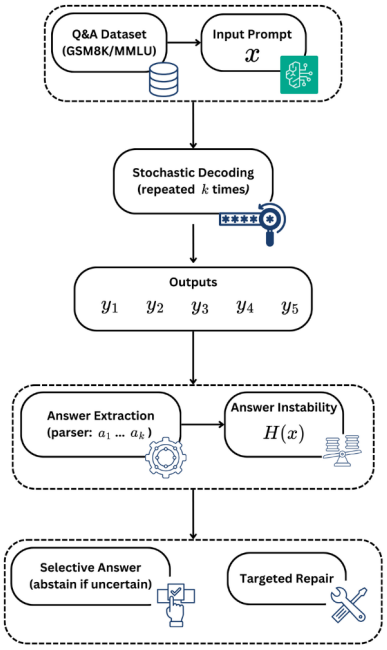

Figure 1: **Overview of the proposed method.** A prompt is decoded k times stochastically; discrete answers are extracted and aggregated into an empirical distribution. The entropy of this distribution serves as a label-free uncertainty signal, which drives selective answering or targeted repair depending on threshold τ.

1 Introduction

Large language models (LLMs) are increasingly deployed in applications that require reliable reasoning and decision-making. Despite their strong performance across a wide range of tasks, these models remain prone to producing incorrect or hallucinated outputs while exhibiting high confidence (Desai and Durrett, 2020; Xiong et al., 2023; Carandang et al., 2025). This mismatch between accuracy and confidence presents a fundamental challenge: how can we determine when a model's output should be trusted?

Existing approaches to uncertainty estimation typically rely on internal model signals such as token probabilities (Jiang et al., 2021; Kadavath et al., 2022), post-hoc calibration techniques (Guo et al., 2017; Desai and Durrett, 2020), or additional training procedures (Kong et al., 2020; Tian et al., 2023). However, these methods require access to model internals, labeled calibration data, or task-specific tuning, limiting their applicability in real-world deployment settings where models are often accessed as black-box APIs.

In parallel, recent work has demonstrated that stochastic decoding and repeated sampling can improve performance through techniques such as self-consistency (Wang et al., 2023; Aggarwal et al., 2023; Lightman et al., 2023). While these approaches exploit variation across sampled outputs to improve accuracy, they do not directly leverage this variation as a signal of model uncertainty

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 816–826

(Kuhn et al., 2023; Vashurin et al., 2025). Unlike SelfCheckGPT (Manakul et al., 2023), which measures consistency at the passage level using semantic similarity, our approach operates on discrete extracted answers and formalizes their distribution as an entropy-based signal, enabling direct use as an abstention threshold or repair trigger without a semantic model.

In this work, we propose answer instability as a simple and practical proxy for uncertainty in LLMs. The key idea is to measure how much a model's final answer varies across multiple stochastic generations of the same prompt. Intuitively, when a model is confident, repeated sampling yields consistent answers; when uncertain, the model produces diverse and conflicting outputs.

We formalize answer instability as the entropy of the empirical distribution of answers obtained from repeated sampling. This formulation requires no access to model probabilities, no additional training, and no labeled data. It operates purely on observable outputs, making it applicable to any black-box LLM.

We evaluate this approach across multiple models and task types, including reasoning, multiple-choice question answering, and constraint-following tasks. Our results show that instability-based entropy reliably predicts model errors, achieving consistent separation between correct and incorrect outputs. We further demonstrate that this signal can be used to guide selective answering and to trigger targeted repair strategies, improving accuracy while reducing computational cost.

Our contributions are:

- We introduce answer instability as a label-free, black-box uncertainty signal for LLMs.
- We demonstrate that entropy over sampled answers consistently identifies incorrect outputs across models and tasks, achieving performance competitive with semantic entropy while requiring no semantic similarity model or equivalence class definitions.
- We show that instability enables efficient compute allocation via targeted repair, achieving comparable accuracy to uniform repair with reduced token usage.

2 Related Work

Uncertainty estimation and calibration. A key challenge in deploying language models is determining when their outputs can be trusted. Guo et al. (2017) showed that deep networks are often overconfident and can be improved with temperature scaling. Extending this to Transformers, Desai and Durrett (2020) and Kong et al. (2020) demonstrated that calibration degrades under domain shift but can be partially addressed through regularization. For generative models, Jiang et al. (2021) and Zhao et al. (2021) found that QA systems and few-shot predictions exhibit significant miscalibration due to biases and context effects. Instruction-tuned models further complicate calibration, with Tian et al. (2023) and Xiong et al. (2023) showing that confidence elicitation varies across models and tasks. More recent work focuses on semantic-level uncertainty, including semantic entropy (Kuhn et al., 2023), kernel-based similarity (Nikitin et al., 2024), and meaning-aware scoring (Bakman et al., 2024). Benchmarks such as LM-Polygraph (Vashurin et al., 2025) confirm that these approaches outperform token-based methods, while Kadavath et al. (2022) and Geng et al. (2024) highlight ongoing challenges in factual reliability and generalization.

Sampling-based reasoning. Another line of work leverages stochastic sampling to improve accuracy rather than measure uncertainty. Large language models have been shown to perform well in few-shot settings (Brown et al., 2020), with reasoning performance further improved through chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022). Self-consistency (Wang et al., 2023) aggregates multiple sampled reasoning paths via majority voting, achieving substantial gains. Subsequent work improves efficiency through adaptive stopping (Aggarwal et al., 2023) and enhances reasoning quality via process-level reward models (Lightman et al., 2023). More broadly, Snell et al. (2024) show that increasing test-time compute through sampling can rival scaling model size. These approaches treat sampling primarily as a mechanism for improving final predictions.

Selective prediction. Selective prediction allows models to abstain when uncertain. Geifman and El-Yaniv (2017) formalized this framework, showing that abstention can guarantee bounded risk. In NLP, Kamath et al. (2020) and Xin et al. (2021) demonstrated that confidence estimation improves accuracy-coverage trade-offs, while Chen et al. (2023) extended these ideas to instruction-tuned LLMs using parameter-efficient methods. However, these approaches typically require calibration

data or additional training to estimate confidence reliably.

3 Methodology

Given an input prompt x, an LLM M_θ produces an output y through stochastic decoding. In typical black-box deployment settings, the model does not expose calibrated confidence scores, token-level probabilities, or internal representations, and ground-truth labels are unavailable at inference time.

Our goal is to estimate a label-free uncertainty signal $U(x)$ directly from observable model outputs. The central hypothesis is that prompts yielding inconsistent outputs under repeated stochastic sampling correspond to higher epistemic uncertainty and, consequently, higher probability of error.

Formally, under stochastic decoding with temperature $T > 0$, we draw k independent samples from the model:

$$y_i \sim M_\theta(x; T), \quad i = 1, \ldots, k. \tag{1}$$

From each output y_i we extract a discrete final answer a_i using a task-specific parser. For reasoning tasks, the parser extracts the final numerical value; for MCQ, it extracts the selected option letter; for constraint-following tasks, it checks whether the structural constraint is satisfied and returns a binary outcome. Constraint satisfaction is assessed programmatically using rule-based string matching (e.g., checking for valid JSON structure or absence of digit characters). This yields the multiset of sampled answers:

$$A(x) = \{a_1, a_2, \ldots, a_k\}. \tag{2}$$

The final predicted answer is taken as the majority vote:

$$\hat{a}(x) = \arg\max_{a \in A} \; p(a \mid x), \tag{3}$$

where $p(a \mid x) = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}[a_i = a]$ is the empirical frequency of answer a.

The answer parser is designed to be robust to surface-level variation that does not reflect genuine uncertainty. For reasoning tasks, the parser extracts the final numerical value from each response, normalizing integer and decimal formats so that equivalent answers such as 26 and 26.00 are treated as identical. This ensures that entropy reflects disagreement over the final answer rather than formatting variation introduced by stochastic decoding. In practice, parser normalization was most frequently triggered on reasoning tasks, where $T = 0.7$ occasionally produced responses with different arithmetic intermediate steps converging to the same numerical answer.

3.1 Answer-Distribution Entropy

We quantify uncertainty as the Shannon entropy of the empirical answer distribution over $A(x)$:

$$U_{\text{entropy}}(x) = -\sum_{a \in A(x)} p(a \mid x) \log p(a \mid x). \tag{4}$$

Entropy is a natural choice here because it captures the full shape of the answer distribution, not just whether the top answer is dominant. A prompt where all k samples agree yields zero entropy regardless of k; a prompt where answers are spread uniformly across k distinct values yields maximum entropy $\log k$. This makes entropy more sensitive to genuine uncertainty than simpler measures such as vote margin (which ignores the distribution of non-majority answers) or unique answer count (which ignores frequency differences).

With $k = 5$ samples, the empirical answer distribution is necessarily discrete, with entropy values determined by a limited set of partition patterns (e.g., 4–1, 3–2, 3–1–1). In practice, however, this resolution is sufficient to capture the most informative distinction: full agreement ($H = 0$) versus any disagreement ($H > 0$).

This suggests that coarse-grained variability is already a strong indicator of model uncertainty, and that fine-grained estimation may offer diminishing returns in structured tasks. Increasing the sampling budget would provide a smoother entropy spectrum and may improve discriminative resolution, but at higher computational cost. We leave a systematic exploration of this trade-off to future work.

Crucially, this estimate requires no access to token-level probabilities, no model retraining, and no labeled data. It operates solely on the discrete answers observable from any black-box LLM.

3.2 Predictive Validity

To evaluate whether $U(x)$ reflects correctness, we define a binary outcome variable:

$$z(x) = \begin{cases} 1 & \text{if } \hat{a}(x) \text{ is correct,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

We assess the discriminative power of $U(x)$ for predicting error $(1 - z(x))$ using the Area Under

the ROC Curve (AUC-ROC). An AUC of 0.5 corresponds to a random baseline; values above 0.5 indicate that higher instability reliably co-occurs with incorrect predictions.

3.3 Selective Answering

Beyond predicting errors, we evaluate whether $U(x)$ can be operationalized as a decision rule. Given a threshold τ, we define a selective answering policy:

$$\text{answer if } U(x) \leq \tau, \quad \text{abstain otherwise.} \tag{6}$$

This induces a coverage–accuracy trade-off: lower τ produces higher selective accuracy at the cost of answering fewer prompts. We vary τ across the full range of observed entropy values and report the resulting coverage–accuracy curve, which evaluates whether instability is not only predictive but also *actionable* as a deployment-time abstention policy.

3.4 Uncertainty-Triggered Repair

For prompts where $U(x) > \tau$, rather than abstaining we optionally apply a *repair* strategy: the model is re-prompted with an explicit chain-of-thought instruction requesting step-by-step reasoning, and the resulting output replaces the original majority answer. We compare four policies:

- **No repair**: always return the majority answer.
- **Random repair**: apply repair to a randomly selected subset of prompts of the same size as the triggered set.
- **Uncertainty-triggered repair**: apply repair only when $U(x) > \tau$.
- **Always repair**: apply repair to all prompts.

We measure accuracy and total token cost for each policy. This experiment evaluates whether instability can guide *targeted* reasoning improvements — allocating additional computation selectively to the cases most likely to benefit from it, rather than uniformly to all outputs.

4 Experimental Setup

We evaluate the proposed instability-based uncertainty signal across multiple models and task types. Experiments are conducted using four LLMs from two providers, including both high-capacity and lightweight variants from OpenAI and Claude model families. Specifically, we evaluate `GPT-4o`, `GPT-4o-mini` (OpenAI), `claude-sonnet-4-5-20250929`, and `claude-haiku-4-5-20251001` (Anthropic). This setup allows us to assess whether the relationship between answer instability and model error generalizes across architectures and capability levels.

To assess the generality of our approach across qualitatively different output types, we evaluate on three task categories: deterministic multi-step reasoning, multiple-choice question answering (MCQ), and constraint-following generation requiring structured outputs. For reasoning, we sample problems from the GSM8K benchmark (Cobbe et al., 2021), a dataset of 8.5K grade school math word problems, and for MCQ, we draw questions from MMLU (Hendrycks et al., 2021), a 57-subject benchmark spanning various domains. For constraint-following, we construct a set of handcrafted instruction prompts that impose explicit structural or lexical constraints on the output—for example, *"List 3 colors as a JSON array with exactly 3 items"* or *"Write a sentence about weather without using any numbers or digits."* This task type probes whether instability manifests in settings where correctness is determined by rule compliance rather than factual accuracy. The full evaluation set consists of 130 prompts (50 GSM8K, 50 MMLU, and 30 constraint-following), evaluated across four models, yielding 520 prompt-model pairs in total.

For each prompt x, we generate $k = 5$ independent samples using stochastic decoding with non-zero temperature. In all experiments, we use temperature T = 0.7. From each generated response, we extract the final answer using task-specific parsers. The final prediction is determined via majority voting across samples.

Uncertainty is computed using entropy of the empirical answer distribution derived from the sampled responses. We evaluate the predictive validity of this uncertainty signal using AUC-ROC, and assess its usefulness for selective answering and uncertainty-triggered repair strategies.

To situate our answer instability measure against a theoretically grounded uncertainty baseline, we implement semantic entropy (SE) following (Kuhn et al., 2023). SE groups the $k = 5$ sampled responses into semantic equivalence classes and computes Shannon entropy over the resulting class distribution. For structured tasks with categorical or numeric outputs, exact-match equivalence serves as the clustering criterion — numerically equiva-

lent answers (e.g., `26` and `26.00`) are normalized before comparison, MCQ responses are mapped to their letter label (`A`–`D`), and constraint outputs are grouped by constraint satisfaction (`satisfied`/`not satisfied`). This yields a consistent, reproducible SE estimate across all three task types without requiring an external NLI model or embedding threshold. We note that for tasks with discrete output spaces, this SE implementation is functionally equivalent to answer instability under exact-match normalization. Since both SE and our instability signal $U(x)$ are computed from the same k samples, any difference in predictive performance isolates the contribution of semantic normalization rather than sampling cost.

5 Results and Discussion

We evaluate whether answer instability can serve as a reliable proxy for model uncertainty. Our experiments address three research questions: (1) whether instability predicts model errors, (2) whether the signal can guide selective decision-making, and (3) whether it can improve computational efficiency through targeted repair.

5.1 Predictive Validity of Answer Instability

We first evaluate whether entropy derived from sampled answers can discriminate between correct and incorrect outputs, and compare our answer instability measure against a semantic entropy (SE) baseline. Figure 2 shows the receiver operating characteristic (ROC) curves for each evaluated model.

Across models, both answer instability and semantic entropy demonstrate consistent predictive ability above the random baseline. Notably, the two measures achieve nearly identical performance for several models, including `claude-haiku-4-5-20251001` (AUC = 0.750) and `claude-sonnet-4-5-20250929` (AUC = 0.799), indicating that semantic normalization provides little additional benefit in these settings. For `gpt-4o-mini`, instability slightly outperforms semantic entropy (0.793 vs. 0.766), while for `gpt-4o`, both measures perform similarly, with a modest advantage for instability (0.643 vs. 0.609).

The 95% bootstrap confidence intervals reported in Figure 2 overlap substantially between answer instability and semantic entropy across all models, confirming that the two measures are statistically indistinguishable in predictive performance. This supports the conclusion that semantic normalization provides no measurable benefit over surface-level answer entropy in structured output settings. The wide CI for `gpt-4o` (0.514–0.768) reflects the small number of errors at high baseline accuracy rather than fundamental unreliability of the signal. With `gpt-4o` achieving perfect accuracy on constraint tasks and 0.96 on reasoning, the AUC estimate rests on few positive cases, inflating resampling variance. This is consistent with the per-task breakdown, where instability is most discriminative on tasks with higher error rates. The lower CI bound approaching chance should therefore be interpreted as a consequence of near-ceiling performance rather than signal failure.

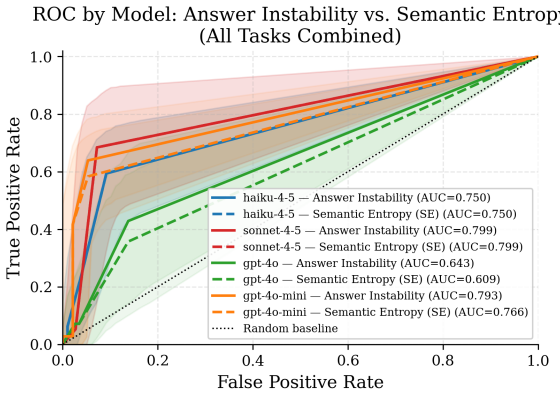

Figure 2: **ROC Curves.** Comparison of answer instability $U(x)$ and semantic entropy (SE) for detecting incorrect outputs across models. 95% bootstrap confidence intervals (1,000 resamples) are reported. Answer Instability AUCs: sonnet [0.712–0.879]; haiku [0.659–0.836]; gpt-4o [0.514–0.768]; gpt-4o-mini [0.710–0.872]. Semantic Entropy AUCs: sonnet [0.712–0.879]; haiku [0.657–0.834]; gpt-4o [0.514–0.768]; gpt-4o-mini [0.707–0.869].

These results suggest that the primary source of predictive signal arises from answer variability itself rather than from semantic clustering. In structured tasks such as reasoning, MCQ, and constraint-following - where outputs naturally collapse to discrete answer spaces — instability-based entropy already captures the relevant uncertainty signal without requiring semantic equivalence modeling.

The relatively lower discriminative power for `gpt-4o` may reflect its higher baseline accuracy (Table 1), which reduces the prevalence of errors and compresses the dynamic range of uncertainty. This indicates that instability is most informative when models operate below their performance ceiling, where variability in outputs is more pronounced.

Breaking down predictive validity by task type,

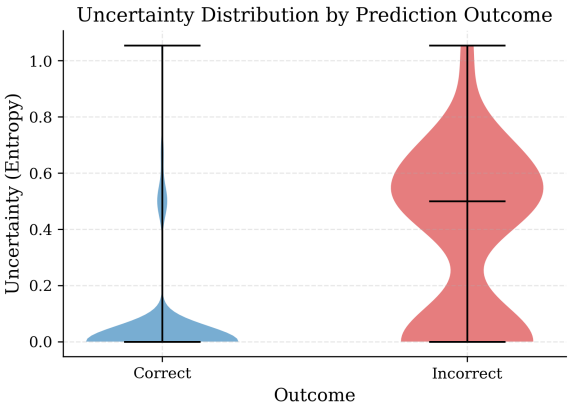

Figure 3: **Uncertainty distribution by prediction outcome.** Violin plots show the entropy distribution of answer instability U(x) for correct (blue) and incorrect (red) predictions, aggregated across all models and tasks. Correct outputs are heavily concentrated at zero entropy, while incorrect outputs exhibit a broad distribution centered around 0.5, reflecting genuine disagreement across sampled responses.

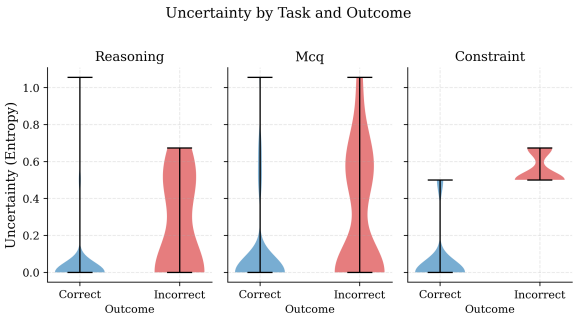

Figure 4: **Uncertainty distribution by task type and outcome.** Entropy distributions for correct (blue) and incorrect (red) predictions broken down by task category. Constraint-following tasks show the clearest separation, with incorrect outputs clustering tightly at high entropy consistent with their high AUC of 0.963. MCQ and reasoning tasks exhibit greater distributional overlap between correct and incorrect outputs, reflecting the more moderate AUC values of 0.656 and 0.680 respectively.

instability is most discriminative for constraint-following tasks (AUC = 0.963), with lower discriminative power on MCQ (AUC = 0.656) and reasoning tasks (AUC = 0.680). This pattern is consistent with the binary nature of constraint evaluation: outputs are either compliant or not, so answer variability maps directly onto error risk. In contrast, MCQ and reasoning tasks allow the model to recover correct answers through majority voting even under moderate uncertainty, compressing the discriminative signal.

Overall, the ROC analysis shows that answer instability provides a competitive and often equivalent alternative to semantic entropy, while remaining simpler and fully black-box. Unlike SE, which requires defining equivalence classes or semantic similarity criteria, instability operates directly on observable outputs, making it more practical for deployment across diverse models and task settings. To further characterize the mechanism underlying these ROC results, the following section examines the full entropy distributions for correct and incorrect outputs.

5.2 Uncertainty Distribution Analysis

To further characterize how instability relates to prediction outcomes, Figure 3 shows the entropy distributions for correct and incorrect predictions aggregated across all models and tasks. Correct outputs are heavily concentrated at zero entropy, indicating that the model produces consistent answers when it is right. Incorrect outputs exhibit a broad distribution with mass centered around 0.5, reflecting substantial variability across sampled responses, consistent with output space uncertainty. This asymmetry in distribution shape underlies the discriminative power and provides a clearer picture of the mechanism than aggregate error rates alone. Figure 4 breaks this down by task type, revealing that the quality of the separation varies substantially across tasks. In constraint-following tasks, incorrect outputs cluster tightly at high entropy while correct outputs remain near zero, consistent with the high AUC of 0.963. The separation is clean because constraint correctness is binary — outputs are either compliant or not — leaving little ambiguity in how instability maps onto error risk. For MCQ and reasoning tasks, the correct and incorrect distributions show greater overlap, explaining the more moderate AUC values of 0.656 and 0.680 respectively. In these settings, majority voting can recover correct answers even under moderate uncertainty, compressing the entropy gap between correct and incorrect outputs.

Together, these distributions suggest that the informativeness of the instability signal is closely tied to output space structure. Tasks with hard, unambiguous correctness criteria produce cleaner entropy separation, while tasks that allow partial recovery through aggregation yield noisier but still meaningful signals.

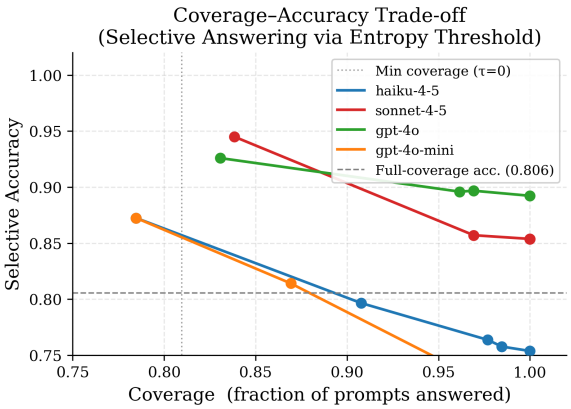

Figure 5: **Coverage–accuracy trade-off.** Lower thresholds (near dotted line $\tau = 0$) yield higher accuracy but reduced coverage.

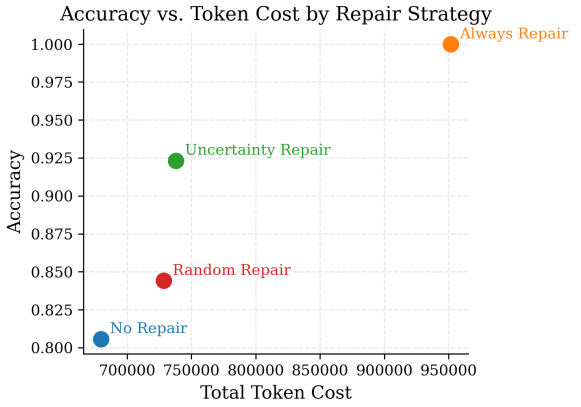

Figure 6: **Repair Efficiency.** Accuracy versus token cost for different repair strategies.

5.3 Error Rates and Failure Modes

The distributional asymmetry observed in Figure 3 is further supported by a direct comparison of error rates across entropy regimes. Among stable outputs where entropy equals zero (n=421), only 40 errors were observed (9.5%), indicating that perfectly consistent answers are highly reliable. Among unstable outputs with non-zero entropy (n=99), 61 errors were observed (61.6%) — a gap of more than 50 percentage points. This stratification confirms that the entropy signal is not merely a continuous discriminator but identifies a practically meaningful binary threshold: any disagreement among sampled responses is a strong indicator of elevated error risk. Among the 40 errors in the stable group, the model produced the same incorrect answer across all five samples, representing high-confidence failures that instability cannot detect by design. These cases represent the primary failure mode of the signal — systematic model misconceptions rather than genuine uncertainty — and are consistent with known miscalibration patterns in instruction-tuned models.

5.4 Selective Answering

We evaluate answer instability in a selective prediction setting, where the model abstains when entropy exceeds a threshold τ, inducing a trade-off between coverage and accuracy. As shown in Figure 5, reducing τ improves accuracy at the cost of coverage: at moderate selectivity (coverage $\approx$ 0.82–0.85), stronger models such as `claude-sonnet-4-5` and `gpt-4o` exceed 0.92 accuracy, well above their full-coverage baseline. As τ increases toward full coverage, accuracy converges to the unfiltered baseline (dashed line), confirming that abstained predictions are disproportionately incorrect. These results demonstrate that answer instability functions not only as a diagnostic signal but as an actionable decision rule for safer LLM deployment.

To assess robustness to threshold choice, we evaluate the selective answering policy across $\tau \in [0.0, 1.0]$. Coverage and accuracy remain stable at 81.0% and 90.5%, respectively, for all $\tau \leq 0.5$, indicating that the policy is insensitive to threshold choice in this range. At $\tau = 0.6$, coverage increases to 94.2% while accuracy drops to 84.2%, reflecting the inclusion of more uncertain outputs. For $\tau \geq 0.7$, the system approaches full coverage at the unfiltered baseline accuracy of approximately 80.9%. This plateau structure suggests a natural operating point at any $\tau < 0.6$, where the abstention policy can be applied without careful threshold tuning.

5.5 Uncertainty-Guided Repair Strategies

Beyond predicting errors, we evaluate whether instability can guide decision-making strategies that improve performance while controlling computational cost across all models and tasks. Figure 6 compares four repair policies.

Always repairing outputs yields the highest accuracy but incurs the largest token cost. In contrast, uncertainty-triggered repair achieves most of the accuracy improvement while requiring substantially fewer tokens. Uncertainty-triggered repair achieves 92.3% accuracy (vs. 100.0% for always-repair and 80.6% for no-repair) at a token cost of 737,865 tokens, compared to 951,624 tokens for always-repair — a reduction of approximately 22.5%. Random repair provides only modest gains

despite similar computational cost, indicating that targeted repair is more effective than naive strategies. These results highlight a practical advantage of instability-based uncertainty: it enables systems to allocate additional computation selectively. Instead of uniformly applying repair procedures to all outputs, the system can focus resources on outputs that exhibit instability, which are significantly more likely to contain errors. This leads to a favorable trade-off between accuracy and computational efficiency.

In interpreting these results, an important methodological consideration arises. The comparison between *no repair*, *targeted repair*, and *always repair* involves two interacting factors: (i) the use of chain-of-thought (CoT) prompting and (ii) the selection mechanism that determines when CoT is applied. In our current setup, the always-repair condition applies CoT universally, whereas the no-repair condition relies on majority voting without CoT, meaning that part of the observed gains may reflect the general benefits of CoT prompting in addition to the effectiveness of instability as a selection signal. A more controlled comparison would include a baseline that applies CoT uniformly across all prompts to isolate the marginal benefit of targeted selection, which we leave for future work. We further note that the reported repair accuracies are aggregated across models and task types; the 100% accuracy under always-repair reflects the inclusion of near-ceiling settings (e.g., reasoning tasks for stronger models), which can elevate aggregate results and obscure variation across conditions. Despite these considerations, the key finding remains that instability-based selection achieves competitive accuracy while reducing token usage, supporting its role as a practical heuristic for adaptive compute allocation in LLM systems.

5.6 Performance Across Models and Tasks

To better understand how model performance varies across tasks, Table 1 summarizes accuracy by model and task category.

As shown in Table 1, performance differs substantially across task types. Reasoning tasks consistently achieve the highest accuracy across all models, whereas multiple-choice (MCQ) and constraint-following tasks are more challenging. Notably, even the strongest models—`claude-sonnet-4-5-20250929` (1.00) and `gpt-4o` (0.96) on reasoning—experience significant drops in performance on MCQ and constraint tasks. This indicates that no model is uniformly reliable across task formats.

Model	Reasoning	MCQ	Constraint
GPT-4o	0.96	0.76	1.00
GPT-4o-mini	0.88	0.60	0.67
Claude Sonnet	1.00	0.82	0.67
Claude Haiku	0.92	0.64	0.67

Table 1: Accuracy across models and task types.

These disparities highlight the need for prompt-level uncertainty estimation. While aggregate accuracy may appear high, performance degrades in more structured or constrained settings, where errors are less predictable. In such cases, instability-based entropy provides a mechanism for identifying difficult or error-prone prompts without requiring labeled data or task-specific calibration.

6 Conclusion

We presented answer instability—the variability of model outputs under repeated stochastic decoding—as a simple, label-free, and black-box proxy for uncertainty in LLMs. Our experiments show that entropy over sampled answer distributions reliably predicts model errors across task types and model families, without requiring access to internal probabilities or additional training. Notably, answer instability achieves performance competitive with semantic entropy across all evaluated models and tasks, demonstrating that semantic normalization provides little additional benefit in structured output settings where answers naturally collapse to discrete spaces. This signal is also actionable: selective answering improves reliability by abstaining on uncertain cases, while uncertainty-triggered repair achieves favorable accuracy–efficiency trade-offs by targeting additional computation only where instability is high.

Future work should extend evaluation to larger and more diverse prompt sets, open-source model families, and more complex generation settings such as long-form or dialogue tasks. Incorporating semantic equivalence into the answer distribution—treating paraphrases as identical answers—would improve instability estimation in open-ended settings. More broadly, answer instability offers a lightweight alternative to traditional calibration methods and a practical foundation for building more reliable LLM-based systems.

7 Limitations

Our evaluation is limited to three task types and 520 prompt-model pairs, which may not fully represent real-world applications such as long-form generation, dialogue, or multimodal reasoning. Answer instability also incurs additional inference cost through repeated sampling, which may be prohibitive in latency-sensitive settings; adaptive or early-stopping strategies could mitigate this. Our formulation operates on surface-level final answers and does not account for semantic equivalence, potentially overestimating uncertainty in open-ended tasks where multiple phrasings are correct. For MCQ and numerical reasoning tasks, surface-level answer matching is standard and this limitation is minimal. For constraint-following and open-ended tasks, semantic equivalence may inflate instability estimates, a direction we leave for future work. We do not evaluate against white-box methods that use token probabilities or internal representations, as our focus is strictly on black-box applicability. Finally, further validation on larger datasets, open-source model families, and real-world deployment scenarios is needed to fully establish the generality of answer instability as an uncertainty signal.

8 Ethical Considerations

This work proposes a black-box uncertainty metric evaluated on publicly available benchmarks (GSM8K, MMLU) and author-constructed prompts; it does not involve human subjects, personal data, or sensitive content. Model outputs are used solely for measuring answer variability and do not inform decisions about individuals. We do not make claims about model cognition or internal reasoning, and we caution against interpreting low instability as a guarantee of correctness or high instability as evidence of model failure. Answer instability is an empirical signal over observable outputs and should not be overgeneralized as a definitive measure of reliability or trustworthiness in deployment.

Acknowledgments

This work was supported by the Department of Science and Technology – Science Education Institute (DOST-SEI) under the ASTHRDP Graduate Scholarship Program, and the Asian Institute of Management. Special thanks to Adamson University for its institutional support.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396, Singapore. Association for Computational Linguistics.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Kristine Ann M. Carandang, Jasper Meynard Arana, Ethan Robert Casin, Christopher Monterola, Daniel Stanley Tan, Jesus Felix B. Valenzuela, and Christian Alis. 2025. Are LLMs reliable? an exploration of the reliability of large language models in clinical note generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1413–1422, Vienna, Austria. Association for Computational Linguistics.

Jiefeng Chen, Jinsung Yoon, Sayna Ebrahimi, Sercan Arik, Tomas Pfister, and Somesh Jha. 2023. Adaptation with self-evaluation to improve selective prediction in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5190–5213, Singapore. Association for Computational Linguistics.

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

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *ArXiv*, abs/1705.08500.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1326–1340, Online. Association for Computational Linguistics.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: fine-grained uncertainty quantification for llms from semantic similarities. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *ArXiv*, abs/2408.03314.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025. Benchmarking uncertainty quantification methods for large language models with LM-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on*

Neural Information Processing Systems, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051, Online. Association for Computational Linguistics.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *ArXiv*, abs/2306.13063.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.