

Risk-Calibrated Prompting: Selective Answering for Robustness Under Prompt Perturbations

Neelu Verma¹

¹Indian Institute of Technology Jodhpur, India

Publication Date: 2026/10/06

Abstract: Large language models (LLMs) can produce fluent responses that remain unreliable when a question is rephrased, surrounded by irrelevant context, or intrinsically ambiguous. Most prompt-based hallucination mitigation methods optimize answer quality or average factual error, whereas deployment also requires a decision about whether an answer should be released at all. This paper presents Risk-Calibrated Prompting (RCP), an inference-time selective answering framework that combines semantic consistency across controlled prompt perturbations with model-verbalized confidence. RCP constructs a canonical prompt, a meaning-preserving paraphrase, and a controlled stress variant; obtains independent responses from the same model; estimates semantic agreement; combines agreement with verbalized confidence through a calibrated risk score; and releases the canonical answer only when the score crosses a validation-selected threshold. The framework requires neither parameter fine-tuning nor an external retrieval index. We define a paired evaluation protocol over TruthfulQA and the question-answering subset of HaluEval using open-weight instruction models, fixed perturbation templates, risk-coverage analysis, calibration metrics, robustness measures, cost accounting, and human audit. The protocol explicitly separates uncertainty signaling from factual truth: high agreement is treated as evidence of behavioral stability rather than proof of correctness. This formulation provides a reproducible basis for studying when an LLM should answer, when it should abstain, and how selective reliability changes under prompt perturbations.

Keywords: Large Language Models; Hallucination; Uncertainty Calibration; Selective Prediction; Abstention; Prompt Robustness; Semantic Consistency; Reliability.

How to Cite: Neelu Verma (2026) Risk-Calibrated Prompting: Selective Answering for Robustness Under Prompt Perturbations. *International Journal of Innovative Science and Research Technology*, 11(9), 2386-2395. <https://doi.org/10.38124/ijisrt/26sep1461>

I. INTRODUCTION

Large language models (LLMs) are increasingly used as general-purpose interfaces for question answering, analysis, summarization, and decision support. Their usefulness, however, is not determined by average accuracy alone. A system can be correct on many requests while remaining unsafe for a smaller subset on which it expresses an unjustified answer with high confidence. TruthfulQA was created to test whether models reproduce common false beliefs, while HaluEval provides a broader benchmark for hallucination evaluation. [1], [2]

This distinction matters operationally. In a deployed assistant, an incorrect answer and an abstention are not equivalent outcomes. Selective prediction formalizes this trade-off through coverage—the fraction of cases on which a system answers—and risk—the error among answered cases. Classical selective classification shows why a confidence function is useful only when its ordering tracks true error. [10], [11] For LLMs, confidence is often elicited through language rather than exposed as a calibrated probability.

Recent work supplies several ingredients for this direction. SelfCheckGPT uses disagreement among sampled generations as a black-box hallucination signal. Semantic entropy groups generations by meaning rather than surface wording. Prompt paraphrase studies show that linguistic changes can materially alter model behavior, motivating explicit robustness tests. [6], [8], [12] The present study combines these ideas into a selective decision rule rather than another unconditional prompt intervention.

RCP creates a small set of meaning-preserving or stress-test prompt variants, collects candidate answers and self-rated confidence, measures semantic agreement, and applies a calibrated release gate. It does not fine-tune the underlying model and does not require a retrieval index. The method is deliberately small enough to reproduce with public datasets and open-weight instruction models.

The research question is: Can a black-box, inference-time selective policy based on perturbation consistency and verbalized confidence reduce factual risk at controlled coverage while remaining robust to prompt changes? The hypothesis is not that every query should be answered less often. Rather, the hypothesis is that a system that learns to

abstain selectively can obtain a safer risk–coverage trade-off than a system optimized only for unconditional answer accuracy.

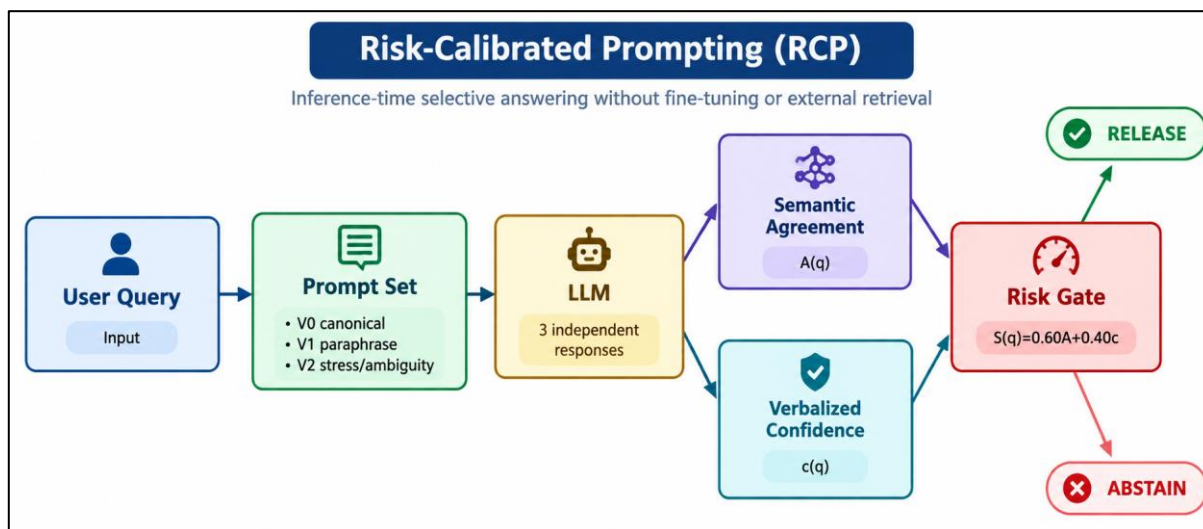


Fig 1 Risk-Calibrated Prompting Workflow. Controlled Prompt Variants Produce Candidate Answers and Confidence Estimates; Semantic Agreement and Confidence are Combined by a Calibrated Release Gate.

➤ Contributions

- An inference-time selective answering framework that targets answer-versus-abstain decisions rather than unconditional hallucination reduction.
- A perturbation-consistency risk signal that combines semantic agreement across controlled prompt variants with verbalized confidence.
- A reproducible risk–coverage evaluation protocol spanning factual QA benchmarks and multiple prompt-stress conditions.
- A decision-centric ablation and error analysis separating calibration failures, perturbation sensitivity, and knowledge-gap failures.

II. RELATED WORK AND PROBLEM GAP

Hallucination research distinguishes errors that contradict evidence from unsupported claims. TruthfulQA focuses on questions designed to trigger misconceptions, while HaluEval provides a large annotated benchmark spanning question answering, dialogue, and summarization. RAGTruth further demonstrates that hallucination remains a relevant evaluation target even in retrieval-augmented settings. [1], [2], [17] These resources support the use of benchmark-level correctness but do not by themselves solve the decision problem of when to abstain.

Prompt engineering can alter model reasoning and output quality. Chain-of-thought and self-consistency established that prompting can change reasoning behavior, and chain-of-verification introduced explicit post-generation checking. The supplied reference paper combines constraint, reasoning, and verification into one compositional intervention and evaluates hallucination reduction. The present paper does not add another component to that stack.

Instead, it changes the objective from response improvement to selective release.

SelfCheckGPT showed that contradictions among independently generated responses can help detect hallucinated content without access to model internals. Semantic entropy later formalized uncertainty at the level of meaning. [6], [8] Calibration research provides tools for measuring whether confidence corresponds to actual correctness, while selective prediction provides risk–coverage curves for abstaining classifiers. [9]–[11] RCP treats semantic agreement as an uncertainty feature and evaluates it through selective risk rather than using agreement as a truth guarantee.

Prompt paraphrase studies have shown that linguistic expression itself can influence model performance, including changes caused by morphology, syntax, and lexical choice. [12] This motivates stress testing prompt-level reliability. A system that appears reliable only under one canonical wording may have brittle decision behavior. RCP therefore evaluates the same question under controlled prompt perturbations and records both answer stability and release-decision stability.

The specific gap targeted here is the intersection of prompt robustness, uncertainty calibration, and selective answering. The reference framework evaluates which prompt components lower the hallucination rate; RCP asks whether a model can use behavioral consistency to decide when to answer. The distinction is methodological and evaluative: risk–coverage, calibration error, and perturbation stability are primary outcomes rather than secondary diagnostics.

Recent chart-focused work also illustrates the broader need for reliability when language is connected to visual

evidence. Query-to-chart retrieval and chart-grounded fact verification frame retrieval and evidence sufficiency as explicit components of trustworthy multimodal reasoning [23]–[25]. These studies motivate a complementary question for text-only LLM interaction: when external evidence is absent, can behavioral consistency and self-reported confidence provide a useful signal for deciding whether a generated answer should be released?

The specific gap targeted here is the intersection of prompt robustness, uncertainty calibration, and selective answering. The framework changes the primary evaluation object from unconditional answer quality to risk–coverage behavior, calibration, and stability under controlled prompt perturbations.

➤ Problem Definition

Let x denote a query, y its ground-truth answer, $a(x)$ the generated answer, and $g(x) \in \{0,1\}$ the release decision. A value $g(x)=1$ means the system releases the answer; $g(x)=0$ means it abstains. Coverage is the expected release probability, while selective risk is the probability of factual error conditional on release.

$$\text{Coverage } C = E[g(x)] \quad (1)$$

$$\text{Risk}(C) = P(a(x) \neq y \mid g(x)=1) \quad (2)$$

This formulation exposes a deployment trade-off hidden by average accuracy. A system that answers all questions can have high coverage but high factual risk. A system that abstains on everything can have zero answer risk but zero utility. The objective is therefore a family of operating points rather than a single accuracy value.

If a model reports confidence 0.8, a calibrated system should be correct on approximately 80% of comparable released answers. Expected calibration error summarizes the weighted gap between confidence and empirical correctness across bins. [9]

$$\text{ECE} = \sum_k (|B_k|/n) \times |\text{acc}(B_k) - \text{conf}(B_k)| \quad (3)$$

The proposed method does not assume verbal confidence is calibrated. The calibration set is used only to select the release threshold and, in an optional sensitivity experiment, score weights. Test labels are never used for threshold selection.

III. PROPOSED METHOD: RISK-CALIBRATED PROMPTING

RCP contains four inference stages. First, the system normalizes the task instruction and constructs three prompt variants: a canonical form, a semantic paraphrase, and a controlled stress variant containing an irrelevant but non-conflicting distractor. For ambiguity analysis, the stress variant is replaced with an ambiguity cue that removes or weakens a key reference. Second, the model produces a concise candidate answer and self-rated confidence on a 0–100 scale. Third, a semantic agreement module compares

candidate answers. Fourth, a calibrated gate maps these signals to RELEASE or ABSTAIN.

The system does not request a hidden chain-of-thought. The response schema contains only the answer, a confidence estimate, and a short uncertainty reason when confidence is low. This keeps the protocol auditable and avoids making private reasoning traces part of the evaluation target.

For each query q , V_0 is the canonical wording, V_1 is a meaning-preserving paraphrase, and V_2 is a controlled stress version. The paraphrase changes syntax or lexical choice without changing the intended answer. The distractor version adds one irrelevant statement while preserving all information needed for the question. The ambiguity version is used only in the robustness condition and deliberately tests whether the system should lower confidence or abstain.

Each transformation is generated from a fixed template library. A 10% sample is manually inspected before the full run. Items judged to alter the intended answer are discarded from the perturbation analysis rather than silently included.

Candidate responses are normalized by lowercasing and removing non-semantic formatting. For multiple-choice or numeric questions, normalized exact matching is used first. For free-form answers, pairwise semantic equivalence is estimated with a lightweight NLI model.

$$A(q) = (s_{01} + s_{02} + s_{12}) / 3 \quad (4)$$

The agreement score is an evidence feature, not a truth label. High agreement can result from shared memorized errors, so stable-but-false responses remain an explicit failure category.

Let $c(q)$ be the mean verbalized confidence across variants. The confidence request uses a fixed wording and integer scale to reduce formatting variability. Responses outside 0–100 are clipped and logged as schema violations rather than silently corrected.

RCP begins with a fixed, pre-specified score that gives slightly more weight to semantic agreement than verbal confidence:

$$S(q) = 0.60 A(q) + 0.40 c(q) \quad (5)$$

The release rule is $g(q)=1$ when $S(q) \geq \tau$. The threshold τ is selected on the calibration split by minimizing selective risk subject to a target coverage of 70%. A secondary experiment tunes the weights on the same development data to test sensitivity.

$$g(q) = 1 \text{ if } S(q) \geq \tau; \text{ otherwise } 0 \quad (6)$$

The abstention response is standardized as: “I am not sufficiently confident to answer reliably from the information available.” A standardized message prevents the system from replacing a low-confidence decision with a new unsupported guess.

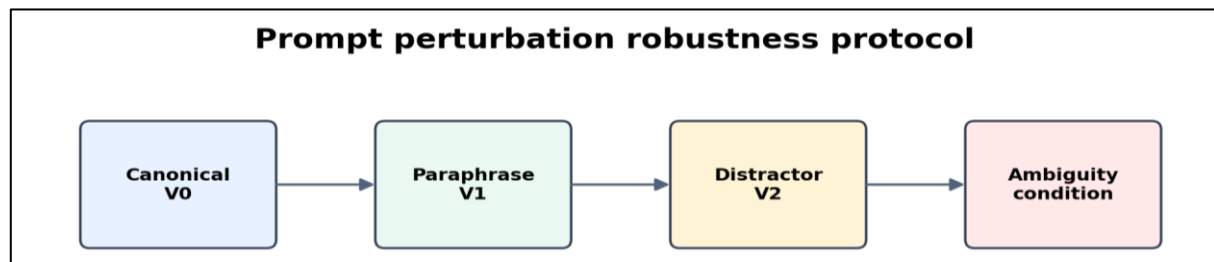


Fig 2 Prompt Perturbation Protocol. The Canonical, Paraphrase, Distractor, And Ambiguity Conditions Expose the Same Underlying Item to Controlled Linguistic Changes While Recording Answer and Decision Stability.

➤ Algorithm

- Input query q and select the benchmark prompt template.
- Generate V_0 , V_1 , and V_2 using the fixed perturbation templates.
- Query the same model independently for each variant at temperature 0.2.
- Extract answer text and verbalized confidence from each response.
- Compute pairwise semantic agreement $A(q)$ and mean confidence $c(q)$.
- Compute $S(q)$ and compare with the validation-selected threshold τ .
- Release the canonical answer if $S(q) \geq \tau$; otherwise return the standardized abstention.
- Store all prompts, outputs, scores, decisions, and labels in JSONL for reproducibility.

➤ Experimental Setup

The evaluation protocol is paired at the question level: every item is evaluated under each prompting condition so that changes in factuality, confidence, and release decisions can be attributed to the intervention rather than to different sampled questions.

• Prompting Conditions

Condition	Variants	Decision rule
Direct	1	Always release the generated answer
Confidence-only	1	Release when confidence $\geq \tau_c$
Consistency-only	3	Release when agreement $\geq \tau_a$
RCP	3	Release when $0.60A + 0.40c \geq \tau$
RCP tuned	3	Validation-selected weights and threshold

• Decoding and Logging

Primary decoding uses temperature 0.2, top-p 0.95, and a maximum output length of 256 tokens. The same schema is used for all models. If an inference server introduces

• Data Splitting

• Hardware and Software

• Datasets

The primary evaluation uses public factuality benchmarks. TruthfulQA contains 817 questions across 38 categories and was designed to test whether models reproduce human falsehoods. HaluEval is a large hallucination benchmark; this study uses its QA subset to maintain a compact task definition. [1], [2] The same question is evaluated under every prompting condition so comparisons remain paired.

The evaluation uses a fixed sample of 400 TruthfulQA questions and 600 HaluEval QA items, with the exact benchmark release, item identifiers, and category metadata recorded in the run manifest. The same item IDs are reused across all prompting conditions.

• Models

The proposed model set consists of Llama 3 8B Instruct, Mistral 7B Instruct v0.3, and Qwen2.5 7B Instruct. The intent is cross-architecture replication rather than a universal claim. Model revision identifiers must be logged. If hardware is constrained, 4-bit quantization may be used, but the exact library, quantization method, and compute device must be reported.

nondeterminism, a seed is recorded. Raw outputs are retained before normalization. The logging schema includes item ID, dataset, model, condition, perturbation type, response text, confidence, agreement, gate score, and final decision.

Table 2 Experimental Configuration Used by the Evaluation Protocol.

Component	Specification
TruthfulQA	400-item evaluation sample; fixed item IDs and category metadata
HaluEval QA	600-item QA sample; exact release and item IDs recorded

Models	Llama 3 8B Instruct; Mistral 7B Instruct v0.3; Qwen2.5 7B Instruct
Decoding	Temperature 0.2; top-p 0.95; maximum output length 256 tokens
Splits	60% development; 20% calibration; 20% test
Semantic evaluator	RoBERTa-large-MNLI baseline or a declared sentence-transformer sensitivity condition
Logging	JSONL records for prompts, outputs, confidence, agreement, score, decision, and labels

IV. EVALUATION METRICS AND STATISTICAL PROCEDURE

Interrupted experiments should resume from item IDs absent from the raw-output store. Existing records are not overwritten unless the model revision or decoding configuration changes, in which case a new run identifier is created. This prevents accidental mixing of incompatible generations.

Verbalized confidence is not guaranteed to be calibrated and can be stylistically biased.

The evaluation focuses on factual QA and does not establish reliability for coding, planning, creative generation, or multimodal tasks.

This paper proposes a distinct reliability problem for LLM research: not simply whether prompting reduces hallucination on average, but whether an LLM can recognize when it should refrain from answering. Risk-Calibrated Prompting combines semantic consistency across controlled prompt variants with verbalized confidence and converts the resulting score into a selective answer-or-abstain decision.

Table 3 Evaluation Metrics and Interpretation.

Metric	Definition / computation	Purpose
Coverage	$E[g(x)]$	Fraction of questions for which the system releases an answer
Selective risk	$P(\text{error} \mid \text{release})$	Factual error among released answers
AURC	Area under risk–coverage curve	Summarizes risk concentration across operating points
ECE	Weighted confidence–accuracy gap across ten bins	Measures calibration of verbalized confidence
Brier score	Mean squared probability error when binary labels are available	Complements ECE with a proper scoring rule
Decision stability	Unchanged release decision / perturbed items	Measures robustness of answer-versus-abstain decisions
Useful abstention	Incorrect-and-abstained / all abstentions	Measures whether rejection is concentrated on risky cases
Latency / tokens / calls	Measured inference cost per item	Quantifies the computational cost of selective answering

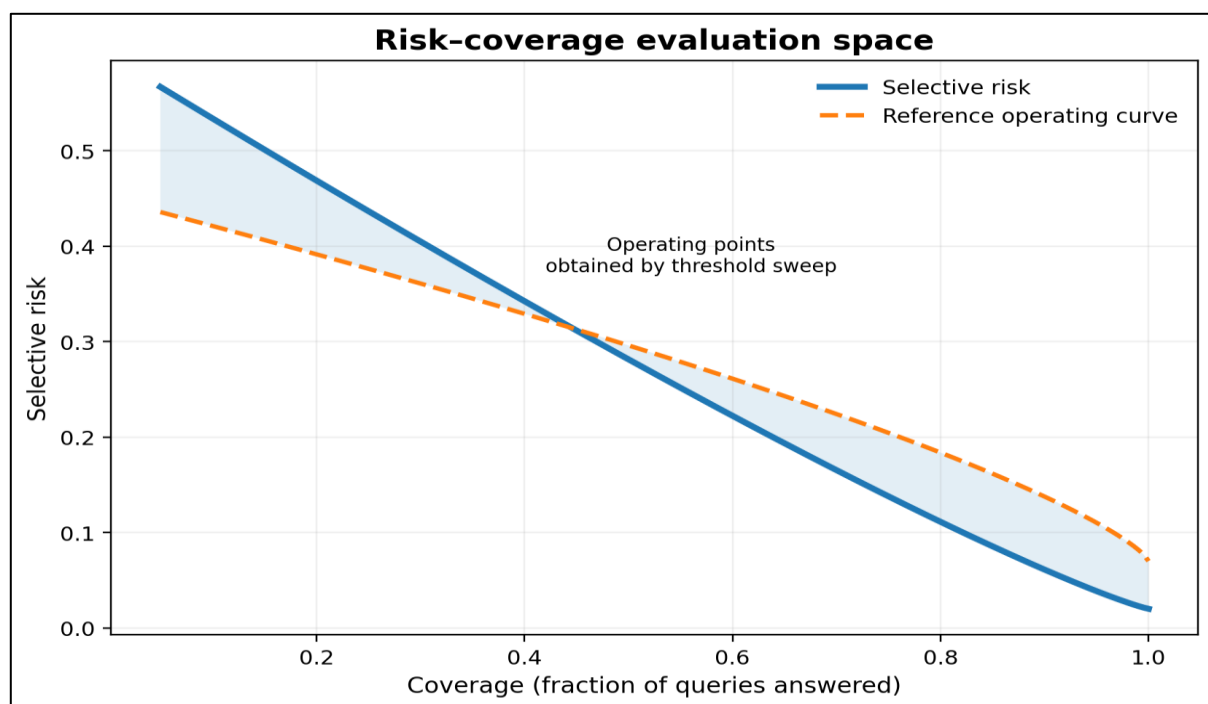


Fig 3 Risk–Coverage Evaluation Space. Threshold Sweeping Produces Operating Points that Expose the Trade-off Between Answering More Queries and Concentrating Factual Errors among Released Answers.

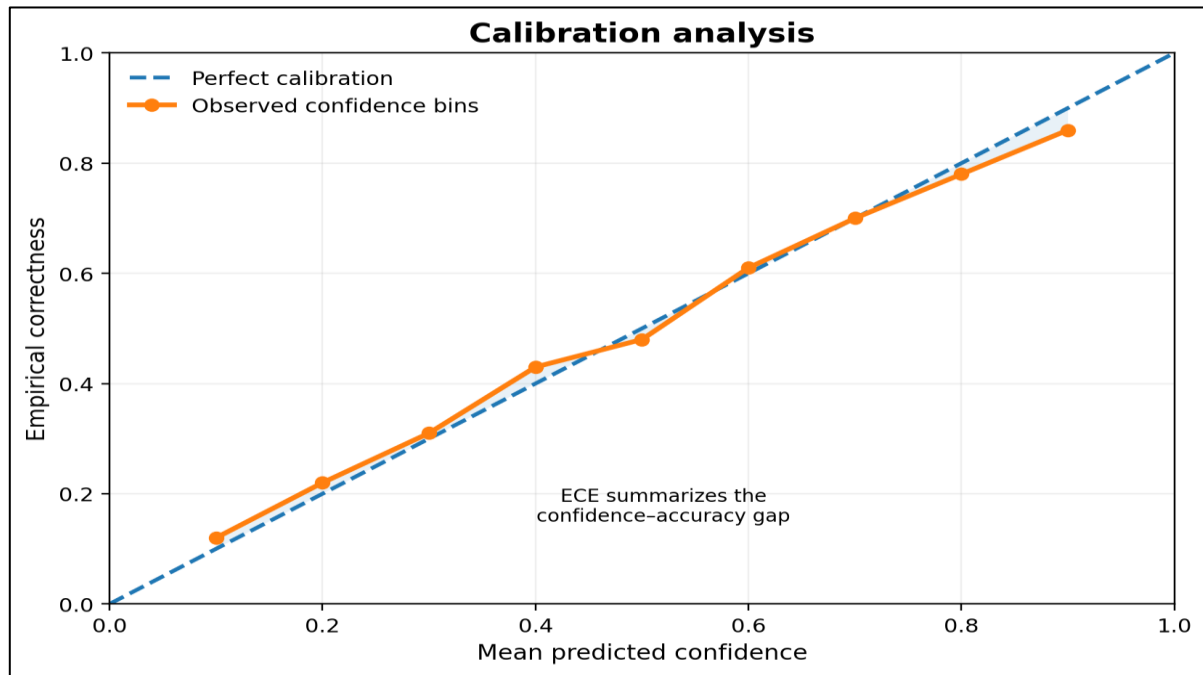


Fig 4 Calibration Analysis. Reliability Diagrams Compare Verbalized Confidence with Empirical Correctness; ECE Summarizes their Deviation.

➤ Ablation and Qualitative Analysis

The ablation design isolates the contribution of verbalized confidence, semantic agreement, and their

combination. Additional runs vary the number of prompt variants so that the reliability gain can be interpreted jointly with inference cost.

Table 4 Ablation Factors and Controlled Comparisons.

Factor	Levels	Comparison objective
Risk signal	Confidence only; agreement only; combined RCP	Determine whether the two signals provide complementary information
Prompt variants	1; 2; 3 variants	Measure the marginal value of additional perturbation queries
Stress condition	Paraphrase; distractor; ambiguity	Separate robustness to harmless wording changes from legitimate uncertainty
Threshold	Coverage targets 50–90%	Trace the complete risk–coverage operating range

• Error Taxonomy

Each audited item receives one primary failure label: confident factual error, unstable-but-correct answer, stable-

but-false answer, ambiguity-induced failure, or unnecessary abstention. The taxonomy distinguishes failures of the risk signal from failures caused by model knowledge gaps or input ambiguity.

Table 5 Decision Matrix for Qualitative Coding.

Ground truth	Gate decision	Interpretation
Correct	Release	Correct release
Correct	Abstain	Unnecessary abstention
Incorrect	Release	Residual factual risk
Incorrect	Abstain	Useful abstention

• Case-Study Protocol

Qualitative analysis uses six paired cases selected after the final test run with fixed selection rules: two correct releases, two useful abstentions, one stable-but-false

response, and one perturbation-induced failure. Each case records the canonical prompt, perturbations, model outputs, confidence values, agreement score, gate decision, and benchmark label.

Table 6 Structured Case-Study Template.

Case type	Evidence to inspect
Correct release	Stable responses, adequate score, benchmark correctness
Useful abstention	Low combined score and incorrect canonical answer
Stable-but-false	High agreement despite an incorrect benchmark label

Prompt sensitivity	Answer or confidence changes after a controlled perturbation
Over-abstention	Stable correct answer rejected by the gate

V. DISCUSSIONS

Average hallucination rate assumes that the system should answer every input. Selective prediction allows the system to refuse a subset of high-risk cases while preserving service on the remainder. This makes risk–coverage analysis more directly relevant to systems where an abstention can trigger human review, retrieval, or a second model.

The conceptual shift also changes how prompt methods are evaluated. A prompt stack may reduce error because the model generates more cautious answers, but that does not show whether the system knows when its answer is uncertain. RCP therefore evaluates the ordering of risk rather than only the final answer distribution.

Behavioral disagreement and verbal confidence capture different signals. A model may be highly confident but unstable across prompt variants, or low-confidence but stable on an easy factual item. The combined score is designed to capture both dimensions. The weight choice is deliberately simple and pre-specified so the experiment can test whether the combination is useful without hiding model-specific optimization inside the method.

SelfCheckGPT motivates disagreement-based hallucination detection, while semantic entropy demonstrates that meaning-level grouping can provide a strong uncertainty signal. [6], [8] RCP differs in its action: it uses uncertainty to decide whether to release an answer. Calibration and selective-classification literature further supports reporting operating curves rather than one thresholded score. [9]–[11]

Prompt paraphrase research demonstrates that linguistic form can change model behavior. [12] The perturbation protocol turns this observation into a reliability test: if a system changes its answer under harmless wording changes, confidence should ideally decline, or the system should abstain. A system that changes its answer while maintaining high confidence represents a particularly important robustness failure.

RCP requires three model calls rather than one in the primary configuration. This increases latency and token usage. The paper therefore, treats cost as a first-class measurement. A future deployment could use a staged policy in which only borderline queries receive all perturbation calls.

For q in test_set : $V = \{V_0(q), V_1(q), V_2(q)\}$; $r = \text{model}(V)$; $A = \text{semantic_agreement}(r)$; $c = \text{mean}(\text{confidence}(r))$; $S = 0.60A + 0.40c$; $\text{output} = \text{answer}$ if $S \geq \tau$ else abstain (7)

For semantic comparison, a compact NLI encoder such as RoBERTa-large-MNLI may be used as a reproducible baseline. If sentence-transformer cosine similarity is substituted because of hardware limits, the change must be reported and tested as a sensitivity condition. The same evaluator must be used across prompting conditions.

The current study does not claim that this optimization is effective; it is a future experiment.

➤ Interpreting Negative Results

The confidence score is a verbal 0–100 self-report rather than a calibrated probability. Consequently, the manuscript should avoid interpreting it as a true probability of correctness. The method only tests whether the signal can help rank responses by factual risk after calibration.

Paired evaluation improves internal validity because every question is exposed to every condition. The main internal threats are prompt-template artifacts, benchmark leakage, evaluator errors, and threshold overfitting. Fixed calibration and test splits, frozen templates, and evaluator checks reduce but do not eliminate these risks.

External validity is limited to factual question answering. The design does not establish that RCP works equally well for creative writing, programming, long-form research, agentic planning, multimodal tasks, or multilingual settings. A future extension could evaluate these settings separately rather than assuming transfer.

Model-version drift is another practical threat. Exact model identifiers, tokenizer versions, quantization, and inference settings must be recorded. Benchmark contamination also cannot be ruled out; therefore the strongest claims should concern paired behavior under the protocol rather than universal truthfulness.

RCP is most directly applicable to systems where abstention can trigger another process such as retrieval, human review, a calculator, a specialized model, or a clarification request. It is less suitable when every request must receive an immediate answer. This is why latency, coverage, and useful-abstention rate are reported together.

➤ Reproducibility and Implementation

Recommended software versions are Python 3.11, PyTorch 2.x, Transformers 4.x, Datasets 2.x, Sentence-Transformers 3.x, scikit-learn 1.x, pandas 2.x, NumPy 2.x, SciPy 1.x, and Matplotlib 3.x. Exact patch versions should be pinned in a requirements file in the final experiment repository.

- Record exact model identifiers and revisions.
- Report decoding parameters, seeds, quantization, and hardware.
- Release prompt and perturbation templates.
- Store raw outputs before normalization.
- Report coverage alongside every risk and accuracy value.

- Provide confidence intervals for AURC, ECE, and paired risk differences.

A reproducible release should contain data/ for benchmark metadata and split identifiers; prompts/ for canonical and perturbation templates; src/ for generation, scoring, calibration, and evaluation code; and results/ for raw outputs, processed metrics, tables, and figures. Raw benchmark text should be redistributed only where licensing permits; otherwise the repository should store download instructions and item identifiers.

Every experiment should write a machine-readable manifest containing timestamp, Git commit, model identifier, tokenizer identifier, quantization, GPU, temperature, top-p, maximum tokens, seed, dataset version, split seed, threshold grid, semantic-evaluator checkpoint, and software package versions. A recommended raw-result schema contains

item_id, model_id, condition, variant, answer, confidence, agreement, score, decision, and ground_truth.

➤ *Failure Recovery and Data Integrity*

Generation failures should be logged rather than silently removed. The run manifest should distinguish inference failures, malformed schema outputs, evaluator failures, and genuine model abstentions. If a model produces no parseable confidence value, the record is marked as a schema error and excluded only from confidence-based analyses; the number and reason for exclusions must be reported.

Interrupted experiments should resume from item IDs absent from the raw-output store. Existing records are not overwritten unless the model revision or decoding configuration changes, in which case a new run identifier is created. This prevents accidental mixing of incompatible generations.

Table 7 Raw-Result Schema for Replication and Audit.

Field	Type	Description
item_id	string	Stable benchmark item identifier
model_id	string	Exact model and revision identifier
condition	string	Direct, confidence, agreement, or RCP
variant	string	V0, V1, V2, or ambiguity
answer	string	Raw model answer
confidence	float	Self-rated confidence from 0 to 100
agreement	float	Semantic consistency score in [0,1]
score	float	Combined RCP score in [0,1]
decision	binary	1 = release; 0 = abstain
ground_truth	binary	Final correctness label

➤ *Limitations*

Verbalized confidence is not guaranteed to be calibrated and can be stylistically biased. Prompt perturbations are template-based and cannot cover all natural user variation. Semantic agreement can reinforce shared false beliefs. The evaluation focuses on factual QA and does not establish reliability for coding, planning, creative generation, or multimodal tasks. Open-weight models improve reproducibility but may not represent proprietary production systems.

The framework is designed for factual question answering and selective release. Its validity depends on the quality of perturbation templates, the semantic evaluator, the benchmark labels, and the calibration split. The method is therefore positioned as a reproducible decision framework rather than a universal detector of factual truth.

VI. CONCLUSION

This paper formulates selective answering as a distinct reliability problem for large language models. Risk-Calibrated Prompting combines semantic consistency across controlled prompt variants with verbalized confidence and converts these signals into an answer-versus-abstain decision. The framework is inference-time, model-agnostic, and reproducible without parameter fine-tuning or an external retrieval index. Its evaluation protocol combines risk-

coverage analysis, calibration, perturbation robustness, inference cost, ablation, and human audit. The central methodological distinction is that behavioral consistency is treated as a risk signal rather than a guarantee of truth. This allows stable-but-false behavior, unnecessary abstention, and perturbation-induced instability to be measured explicitly. The resulting framework provides a concrete basis for studying reliable selective interaction with LLMs in settings where an abstention can trigger retrieval, human review, clarification, or another verification process.

REFERENCES

- [1.] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring How Models Mimic Human Falsehoods," *Proceedings of ACL*, 2022, pp. 3214–3252. doi:10.18653/v1/2022.acl-long.229.
- [2.] J. Li, X. Cheng, X. Zhao, J.-Y. Nie, and J.-R. Wen, "HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models," *Proceedings of EMNLP*, 2023, pp. 6449–6464. doi:10.18653/v1/2023.emnlp-main.397.
- [3.] S. Ji et al., "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, 2023.
- [4.] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems*, 2020.

- [5.] S. Min et al., “FactScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation,” *Proceedings of EMNLP*, 2023, pp. 12076–12100.
- [6.] P. Manakul, A. Liusie, and M. Gales, “SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models,” *Proceedings of EMNLP*, 2023, pp. 9004–9017. doi:10.18653/v1/2023.emnlp-main.557.
- [7.] L. Kuhn, Y. Gal, and S. Farquhar, “Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation,” *arXiv:2302.09664*, 2023.
- [8.] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, “Detecting Hallucinations in Large Language Models Using Semantic Entropy,” *Nature*, vol. 630, pp. 625–630, 2024. doi:10.1038/s41586-024-07421-0.
- [9.] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” *Proceedings of ICML, PMLR 70*, 2017, pp. 1321–1330.
- [10.] Y. Geifman and R. El-Yaniv, “Selective Classification for Deep Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [11.] X. Huang et al., “Uncertainty in Language Models: Assessment through Rank-Calibration,” *arXiv:2404.03163*, 2024.
- [12.] J. P. Wahle, T. Ruas, Y. Xu, and B. Gipp, “Paraphrase Types Elicit Prompt Engineering Capabilities,” *Proceedings of EMNLP*, 2024, pp. 11004–11033. doi:10.18653/v1/2024.emnlp-main.617.
- [13.] F. Shi et al., “Large Language Models Can Be Easily Distracted by Irrelevant Context,” *Proceedings of ICML, PMLR 202*, 2023, pp. 31210–31227.
- [14.] S. Dhuliawala et al., “Chain-of-Verification Reduces Hallucination in Large Language Models,” *arXiv:2309.11495*, 2023.
- [15.] J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *Advances in Neural Information Processing Systems*, vol. 35, 2022.
- [16.] X. Wang et al., “Self-Consistency Improves Chain of Thought Reasoning in Language Models,” *Proceedings of ICLR*, 2023.
- [17.] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang, “RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models,” *Proceedings of ACL*, 2024, pp. 10862–10878. doi:10.18653/v1/2024.acl-long.585.
- [18.] J. Kadavath et al., “Language Models (Mostly) Know What They Know,” *arXiv:2207.05221*, 2022.
- [19.] Y. Zhou et al., “Large Language Models Are Human-Level Prompt Engineers,” *Proceedings of ICLR*, 2023.
- [20.] Y. Yao et al., “Tree of Thoughts: Deliberate Problem Solving with Large Language Models,” *Advances in Neural Information Processing Systems*, vol. 36, 2023.
- [21.] N. Verma, A. De, and A. Mishra, “Bridging Language to Visuals: Towards Natural Language Query-to-Chart Image Retrieval,” *International Journal of Multimedia Information Retrieval*, vol. 13, no. 3, p. 32, 2024.
- [22.] N. Verma and A. Mishra, “Retrieval-Augmented Chart-Grounded Fact Verification with Self-Consistent Vision–Language Reasoning,” *International Journal of Data Science and Analytics*, 2026.
- [23.] N. Verma, “LST-ResNet: Optimization-Based Hybrid Network Architecture for an Effective Question and Answering System with Bar Chart Images,” *International Journal of Scientific Research and Technology*, vol. 3, no. 9, pp. 243–265, 2026.
- [24.] N. Verma, “ASBO-GRU: Optimization-Enhanced Contextual and Aspect-Aware Neural Summarization,” *International Journal of Innovative Research in Technology*, vol. 13, no. 4, pp. 2548–2559, 2026.
- [25.] N. Verma, “ChartSynth: Structured Summarization and Comparison of Multi-Chart Scientific Figures,” *International Journal of Science and Research*, vol. 15, no. 9, pp. 1230–1238, 2026. doi:10.21275/SR26919202852.
- [26.] N. Verma, “AERIS-Chart: Evidence-Sufficient and Uncertainty-Calibrated Retrieval for Chart Claim Verification,” manuscript submitted, 2026.

APPENDIX

APPENDIX A. PROMPT TEMPLATES

- A.1 Canonical. Answer the question using only information you can support. Return exactly two fields: ANSWER and CONFIDENCE. CONFIDENCE is an integer from 0 to 100 representing how likely you believe the answer is correct. Do not invent missing facts. Question: {q}
- A.2 Paraphrase. Provide the most reliable answer to the following question. Preserve the requested meaning while using a different wording of the instruction. Return ANSWER and CONFIDENCE (0–100). If the information is insufficient, indicate low confidence. Question: {paraphrased_q}
- A.3 Distractor. Provide the most reliable answer to the question below. Ignore statements that are unrelated to the question. Return ANSWER and CONFIDENCE (0–100). Question: {q} Context note: {irrelevant_distractor}
- A.4 Abstention. I am not sufficiently confident to answer reliably from the information available.