Contents lists available at IJCHML
International Journal of Computational Health and Machine Learning
Journal Homepage: http://www.ijchml.com/
Volume 4, No. 1, 2026

IJCHML
INTERNATIONAL JOURNAL OF COMPUTATIONAL HEALTH & MACHINE LEARNING

Enhancing Reliability in Large Language Models through Automated Hallucination Detection

Parsa Mazaheri[1],**Selin Ugur**[2],**Mariam Gonzaliam**[3]
[1] *University of California, Santa Cruz, CA, USA*
[2] *Department of Computer Science, Istanbul Technical University, Istanbul, Turkey*
[3] *Department of Mechanical Engineering and Construction, Istanbul Technical University, Istanbul, Turkey*

ARTICLE INFO

Received: 12/12/2025
Revised: 02/16/2026
Accepted: 03/05/2026

Keywords:
hallucination detection, reliability, large language models, automation, natural language processing, machine learning, artificial intelligence

ABSTRACT

Large language models (LLMs) are increasingly used in settings where unsupported or fabricated statements can create material risk. This paper presents a hybrid hallucination detector that combines evidence retrieval, textual entailment, generation uncertainty, and lightweight linguistic anomaly features to identify unreliable model outputs before they are shown to end users. We evaluate the detector on an annotated benchmark of 18,400 generations spanning news summarization, open-domain question answering, and biomedical assistance.

The proposed system achieves a precision of 0.92, recall of 0.88, and F1 of 0.90 on the held-out test set, outperforming confidence-only, retrieval-only, and NLI-only baselines by 9–22 absolute F1 points. The gains are consistent across all three task families, with the largest improvement observed in biomedical assistance, where unsupported entity and dosage claims are particularly common.

We further study an operational deployment setting in which the detector acts as a gating module. Under an abstain-or-regenerate policy, the rate of unsupported responses falls from 22.8% to 11.2% while preserving 92.1% response coverage. These results show that automated hallucination detection can substantially improve the reliability of LLM systems without incurring prohibitive latency, providing a practical path toward safer deployment in high-stakes domains.

1. Introduction

The advent of large language models (LLMs) has transformed natural language processing, enabling strong performance on summarization, question answering, drafting, and interactive assistance. Yet the same systems often produce fluent statements that are unsupported, internally inconsistent, or plainly false. These hallucinations degrade reliability and make LLM deployment risky in domains such as healthcare, law, and scientific communication [2, 5, 9, 14, 16]. Improving factual reliability therefore remains one of the most important barriers between impressive demos and trustworthy production systems.

The practical consequences of hallucination are easiest to see in deployed workflows. In healthcare-oriented assistants, an unsupported dosage recommendation or fabricated contraindication can expose both patients and institutions to harm. In open-domain question answering, hallucinated citations, dates, and named entities reduce user trust because the answer sounds authoritative even when it is wrong. In software engineering and documentation settings, a model that invents APIs, command flags, or configuration options can waste developer time and introduce subtle downstream

defects. Across these settings, the shared problem is not poor fluency but misplaced confidence: the model communicates unsupported claims with the style and confidence of a reliable system.

Recent work has explored a variety of safeguards, including confidence estimation, reference-based verification, auxiliary classifiers, and retrieval-backed generation [3, 7, 11, 13, 15, 19, 22]. Although each of these approaches can detect some errors, single-signal detectors often fail under domain shift: confidence scores miss fluent fabrications, retrieval heuristics struggle with paraphrase, and verifier models can be brittle when evidence is incomplete. These limitations motivate a hybrid view of hallucination detection in which evidence alignment, contradiction detection, uncertainty, and surface anomalies are treated as complementary signals rather than competing alternatives.

This paper studies a practical automated detector that operates on the model response, the originating prompt, and available evidence. The goal is not only to identify hallucinated outputs offline, but also to provide a deployable screening mechanism that can warn, abstain, or trigger regeneration before a risky answer reaches a user. We focus on a setting that reflects actual deployment constraints: the detector must be accurate across multiple task families, lightweight enough for interactive use, and interpretable enough to support downstream review when a response is flagged.

1.1. Understanding Hallucinations in Language Models

Hallucinations in language models can be broadly categorized into intrinsic and extrinsic forms. Intrinsic hallucinations occur when the response contradicts or distorts the provided context, often because the model overgeneralizes from local patterns or compresses multiple facts into an incorrect synthesis [1]. Extrinsic hallucinations occur when the model introduces unsupported world knowledge, fabricated entities, or unverifiable claims that are not grounded in the evidence available at inference time [6, 12]. Both forms matter in practice because users rarely experience them as distinct failure modes; they experience them as a loss of trust.

The root causes of hallucination are similarly multifaceted. Spurious correlations in pretraining corpora, imperfect instruction tuning, exposure bias during decoding, and shifting real-world knowledge all contribute to plausible but incorrect generations [8, 10, 17]. Because these factors interact across model scale, task structure, and evidence availability, no single diagnostic signal is likely to be sufficient in isolation. A useful detector must therefore account for both the *content* of a claim and the *conditions* under which the model produced it.

1.2. Current Approaches to Hallucination Detection

Several methodologies have been proposed to detect hallucinations in LLM outputs. One family of approaches uses auxiliary models trained to classify whether a response is supported by known evidence [6, 9, 19]. Another family uses retrieval or fact-checking pipelines to compare generated claims with trusted passages, while more recent work studies verifier models that predict entailment or contradiction between candidate responses and source evidence [20, 22].

Explainability and uncertainty estimation offer additional signals. Attention analyses, attribution methods, self-consistency checks, and token-level entropy can surface response segments where the model itself appears uncertain or where the generated reasoning is weakly grounded [4, 7, 13, 15, 21]. These approaches are valuable because they can detect errors even when an exact supporting passage is unavailable.

Ensemble and hybrid methods are increasingly attractive because they combine multiple weak signals into a more robust detector [5]. However, the literature still lacks clear evidence about how much each signal contributes in a deployment-oriented setting and whether the resulting detector can improve reliability without excessive latency.

1.3. Challenges and Future Directions

Despite steady progress, several challenges remain. Detection systems must scale to long responses, adapt to domain shift, and remain effective when evidence is incomplete or outdated [8, 10, 11]. High-performing verifiers can also be too slow for real-time settings, creating an operational trade-off between reliability and usability. In addition, benchmarks often mix multiple failure types without clearly separating contradiction, omission, overgeneralization, and unsupported invention, making it difficult to understand where a detector succeeds or fails.

Future research should therefore prioritize lightweight, adaptive detectors that can be integrated directly into LLM serving pipelines [3, 21]. Standardized benchmarks, clearer operational metrics, and stronger cross-domain evaluation will be especially important for moving the field from promising prototypes to dependable safeguards [12, 23].

1.4. Contributions and Research Questions

This paper makes three contributions.

1. We introduce a hybrid hallucination detector that combines retrieval alignment, textual entailment,

generation uncertainty, and lightweight anomaly cues in a single calibrated scoring model.

2. We construct and evaluate on an annotated benchmark of 18,400 LLM generations spanning news summarization, open-domain question answering, and biomedical assistance.

3. We show that the proposed detector reaches 0.92 precision, 0.88 recall, and 0.90 F1, and that using it as a gating module reduces unsupported responses by more than half in a deployment-oriented setting.

To structure the evaluation, we study three research questions.

RQ1: Does a hybrid detector outperform confidence-only, retrieval-only, and entailment-only baselines on held-out hallucination detection?

RQ2: Which feature families contribute most to detection quality across different task domains and error types?

RQ3: Do offline accuracy gains translate into practical deployment benefits when the detector is used to warn, abstain, or trigger regeneration?

2. Related Work

The rapid deployment of large language models (LLMs) in diverse applications has raised concerns regarding their reliability, particularly in terms of hallucination. Hallucination in LLMs refers to the generation of text that is fluent and plausible but factually incorrect or nonsensical. This phenomenon undermines trust in AI systems and poses significant challenges for their adoption in critical domains. Addressing hallucination therefore requires not only stronger generation methods, but also robust mechanisms for detection, diagnosis, and intervention. This section reviews prior work through four lenses: how hallucination is conceptualized, how it is detected, how it is evaluated, and how detection is integrated into live systems.

2.1. Definitions and Taxonomies of Hallucination

Hallucination in language models is a multifaceted issue that encompasses various forms of erroneous content generation. Early works such as [7] and [6] classified hallucinations into intrinsic and extrinsic types, where intrinsic hallucinations arise from contradictions with the provided context and extrinsic hallucinations stem from unsupported external claims. These foundational studies established a useful distinction between errors of *faithfulness* to the source and errors of *factuality* with respect to the outside world, a distinction that is also emphasized in more recent surveys of LLM hallucination [17].

Subsequent work broadened this view by showing that hallucination is not a single phenomenon but a family of failures with different causes and severities [1, 8]. Some generations are wholly fabricated, while others are only partially grounded, mixing correct content with unsupported details. This distinction matters for detection: binary classifiers that work well on obvious fabrications may perform poorly on nuanced responses that are only locally incorrect. Recent studies have therefore argued for more fine-grained taxonomies that distinguish contradiction, unsupported elaboration, speculative interpolation, and temporal staleness [4, 9, 18, 23].

2.2. Confidence- and Uncertainty-Based Detection

One prominent line of work detects hallucinations by monitoring the model's own predictive behavior. Statistical and heuristic methods estimate hallucination risk using token probabilities, entropy, decoding instability, or disagreement across multiple samples [2, 8, 15, 21]. These methods are attractive because they are computationally cheap and require little task-specific supervision. They can often identify moments where the model is uncertain or where the generated sequence is unusually unstable.

However, confidence-based methods have a well-known limitation: fluent hallucinations are often produced with high confidence. A model can be very certain about an incorrect entity, date, or dosage recommendation if that pattern is statistically plausible under its training distribution. As a result, uncertainty signals are informative but incomplete; they help identify risky outputs, yet they rarely establish whether a claim is actually supported by evidence [1, 13]. This motivates combining uncertainty with more explicit evidence-based checks.

2.3. Evidence-Based and Verification Approaches

A second family of work grounds hallucination detection in external evidence. Retrieval-based methods compare generated text with trusted passages, databases, or source documents to determine whether the response can be substantiated [3, 10, 19]. These methods are especially effective in summarization and knowledge-intensive question answering, where evidence is available and the main challenge is measuring alignment between the response and the source.

Verification-oriented approaches extend this idea by explicitly modeling entailment and contradiction. Rather than relying on surface overlap alone, verifier models

predict whether a candidate claim is entailed, contradicted, or unsupported by retrieved evidence [5, 9, 20, 22]. This line of work substantially improves robustness to paraphrase and lexical variation. At the same time, its success depends heavily on the quality of retrieval: if the evidence store is incomplete or if the retrieval stage misses the relevant supporting passage, even a strong verifier may fail.

Auxiliary discriminators provide a related but distinct strategy. Instead of testing individual claims against explicit evidence, a secondary model is trained to classify whether a full response is hallucinated based on patterns observed in annotated data [12, 13]. These models can capture contextual regularities that rule-based systems miss, but they risk learning benchmark-specific shortcuts unless they are evaluated carefully under domain shift.

2.4. Hybrid and Pipeline-Based Methods

Recent work increasingly treats hallucination detection as a pipeline problem rather than a single-model prediction task. Hybrid approaches combine retrieval, verification, uncertainty, and rule-based cues to exploit their complementary strengths [4, 11]. In these systems, different modules contribute different evidence: retrieval checks grounding, entailment identifies contradiction, uncertainty captures generation instability, and lightweight heuristics flag brittle patterns such as suspicious numerals or unsupported named entities.

This shift toward hybrid systems reflects an emerging consensus that hallucination detection is inherently multi-signal. No individual component is sufficient across all tasks and domains. In medical and scientific settings, evidence-based verification is essential; in conversational settings with incomplete evidence, uncertainty and anomaly cues become more important. The major open question is therefore not whether one signal dominates universally, but how multiple signals should be fused under realistic deployment constraints.

2.5. Benchmarks, Metrics, and Evaluation Gaps

The assessment of hallucination detection systems requires robust evaluation metrics and standard benchmarks. As noted in [1], precision, recall, and F1-score are commonly used metrics, yet they often fail to capture the nuanced nature of hallucination. In response, [9] proposed evaluation schemes that consider both factual accuracy and contextual relevance, offering a more comprehensive assessment of model behavior.

Benchmarks such as those developed by [5] and [11] provide standardized datasets for evaluating hallucination detection algorithms. More recent resources such as HaluEval, FELM, ANAH, FactCHD, and Mu-SHROOM have expanded this evaluation space with benchmark settings for large-scale hallucination recognition, fine-grained factuality assessment, analytical annotation, fact-conflicting reasoning, and multilingual span detection [14, 16, 18, 20, 23]. These resources have accelerated progress by enabling more consistent comparison across methods. Even so, current benchmarks still underrepresent several practical issues, including mixed-support responses, time-sensitive claims, multilingual evidence, and deployment-oriented trade-offs such as abstention cost and latency. Many studies also report aggregate metrics without sufficient error analysis, making it difficult to identify which types of hallucination remain most resistant to detection.

These gaps motivate the benchmark design in the present paper. We evaluate across multiple task families, report both detection quality and latency, and analyze failure modes beyond headline metrics. This makes it easier to assess not only whether the detector works, but also where it is likely to break.

2.6. Integration into Language Model Pipelines

Integrating hallucination detection mechanisms into the language model pipeline is critical for real-time applications. [12] demonstrated the feasibility of embedding detection modules within the generation process, allowing for on-the-fly correction of hallucinations. Similarly, [10] explored feedback loops where user interactions help refine detection systems over time.

The integration strategies are further refined in [4, 21, 22], which highlight the importance of balancing detection accuracy with computational efficiency, ensuring that real-time applications remain feasible without compromising reliability. This deployment-oriented perspective is particularly important because a detector that is too slow, too opaque, or too conservative may be unusable even if it performs well offline.

In summary, the existing body of work provides a strong foundation for understanding, detecting, and mitigating hallucinations in large language models. However, the literature still leaves three practical questions open: how to combine heterogeneous signals effectively, how to evaluate detectors under realistic operating constraints, and how to translate detection gains into safer user-facing behavior. The present study addresses these questions through a hybrid detector, a cross-domain benchmark, and an explicit deployment analysis.

3. Methodology

We formulate hallucination detection as a binary prediction problem over a generated response y, conditioned on the user prompt x and an evidence context c. A response is labeled hallucinated if it contains at least one atomic

claim that is unsupported or contradicted by the available evidence. The detector operates at response level during online inference, while retaining the sentence-level signals needed for downstream review and error analysis.

Our design goal is to combine complementary signals without introducing the latency of a second full generation pass. To that end, the proposed detector uses a lightweight fusion model over four feature families: retrieval alignment, textual entailment, generation uncertainty, and linguistic anomaly cues. This yields a practical detector that can be inserted into an existing LLM pipeline as a warning, abstention, or regeneration trigger.

3.1. Problem Formulation

Let $y = (y_1, \ldots, y_T)$ denote a generated response. We define hallucination detection as learning a scoring function $f(x, c, y) \to [0, 1]$ that estimates the probability that y contains unsupported content. In the benchmark used here, a response is marked positive if any atomic factual claim cannot be justified by the associated evidence context or directly contradicts it. This criterion intentionally covers both intrinsic and extrinsic hallucinations.

We distinguish two operational settings. In the *offline evaluation* setting, the objective is accurate classification of hallucinated versus grounded responses on a held-out test set. In the *deployment* setting, the detector acts as a policy signal: if the score exceeds a threshold, the system can warn the user, abstain, or request a regenerated answer. The second setting matters because the value of hallucination detection depends not only on classifier quality but also on how the signal changes user-facing behavior.

3.2. Benchmark Construction and Annotation

We assembled an evaluation benchmark of 18,400 LLM generations spanning three task families: news summarization, open-domain question answering, and biomedical assistance. The benchmark was constructed to stress both intrinsic and extrinsic hallucinations by varying prompt type, evidence completeness, and response length. Table 1 summarizes the resulting dataset.

Prompts were gathered from three complementary sources. News summarization prompts were derived from article-summary pairs where the source article was retained as evidence. Open-domain question answering prompts were sampled from knowledge-intensive QA sets and paired with retrieved passages. Biomedical prompts were constructed from consumer-style medical queries paired with curated reference snippets. We intentionally retained prompts with varying ambiguity levels so that the benchmark included both straightforward factual grounding and harder cases where the evidence only partially supported the answer.

Each response was reviewed by two annotators with access to the originating prompt and evidence context. Annotators were instructed to label a response as hallucinated when it introduced unsupported entities, fabricated numbers, incorrect relations, or contradictions with the source. A response was labeled grounded when all material claims could be supported by the evidence, even if the wording was abstractive. Borderline cases involving missing evidence or harmless stylistic elaboration were adjudicated by a third reviewer. On a doubly annotated subset of 1,200 responses, inter-annotator agreement reached Cohen's $\kappa = 0.82$, indicating strong agreement. The final train, validation, and test split followed a 70/10/20 partition while preserving task-family proportions and the overall positive label rate.

To make the labels operationally meaningful, we also tracked common positive and negative patterns. Positive examples included unsupported drug dosages, invented organization names, incorrect temporal claims, and summaries that attributed opinions to the wrong speaker. Negative examples included concise paraphrases, well-supported abstractions, and answers that omitted uncertain details rather than speculating. These examples informed both annotator training and later error analysis.

3.3. Feature Extraction and Fusion

The detector aggregates four complementary feature families:

1. **Retrieval alignment.** We retrieve evidence sentences from the available context and compute dense similarity, lexical overlap, and unsupported named-entity counts. These features are intended to identify claims that cannot be grounded in the source material.

2. **Textual entailment.** For each response sentence, an entailment model estimates the probability of entailment, neutrality, and contradiction with respect to the top supporting evidence. The response-level contradiction and low-entailment aggregates form strong signals of hallucination risk.

3. **Generation uncertainty.** We use mean token entropy, response-level log-probability statistics, and self-consistency variance across low-cost sampled continuations as indicators of model uncertainty.

4. **Linguistic anomaly cues.** Rule-based features flag unsupported numerals, unverifiable named entities, abrupt tense shifts, and citation-like surface forms

Table 1: Composition of the annotated evaluation benchmark. Hallucination rate denotes the percentage of outputs containing at least one unsupported or contradicted claim.

Task family	Instances	Hallucination rate (%)	Avg. output length	Evidence source
News summarization	7,200	28.7	78 tokens	source article
Open-domain question answering	6,100	35.4	46 tokens	retrieved passages
Biomedical assistance	5,100	43.1	69 tokens	curated medical references
Total / weighted average	18,400	34.9	65 tokens	–

that are not backed by evidence.

The resulting feature vector is passed to a calibrated logistic detector that produces a hallucination probability:

$$\hat{p}(h = 1 \mid x, c, y) = \sigma(W\,[s_{\text{ret}}, s_{\text{ent}}, s_{\text{unc}}, s_{\text{anom}}] + b)\,. \quad (1)$$

This lightweight fusion layer was chosen to keep inference inexpensive while making the contribution of each signal interpretable in ablation experiments.

3.4. Detection Pipeline

The full detection pipeline proceeds in five steps. First, the serving model produces a candidate response for a given prompt. Second, the detector segments the response into sentences and extracts candidate factual spans such as entities, numerals, and relation phrases. Third, the evidence alignment stage retrieves the most relevant supporting sentences from the available context. Fourth, each response sentence is scored with an entailment module and paired with uncertainty statistics gathered during generation. Fifth, all features are fused into a calibrated risk score that is compared against an operating threshold.

This staged design has two advantages. It keeps the individual modules simple enough to analyze in isolation, and it supports deployment interventions at different levels of severity. For example, a low-risk but nonzero score can trigger a soft warning, while a high-risk score can trigger abstention or regeneration. Because the component signals remain available, the detector can also highlight *why* a response was flagged, such as low entailment or unsupported numbers.

3.5. Baselines and Experimental Setup

We compare the proposed hybrid detector against three single-signal baselines: (i) a confidence threshold based on generation probabilities, (ii) a retrieval-overlap verifier that uses evidence similarity alone, and (iii) an NLI verifier that uses entailment scores without uncertainty or anomaly cues. Thresholds are selected on the validation set to maximize F1 subject to a precision floor of 0.90 for the proposed model.

All methods are evaluated on the same held-out test split. We report precision, recall, F1, and area under the receiver operating characteristic curve (AUROC). Because practical deployment also depends on responsiveness, we additionally measure per-response detector latency. Latency is recorded as end-to-end detector time excluding the base model's generation latency, averaged across the held-out split on a single accelerator-backed inference server. While the exact hardware configuration is not central to the comparison, this setup provides a consistent measure of the practical cost of adding each detector.

Finally, we evaluate a deployment-oriented setting in which high-risk responses are either surfaced with a warning or automatically sent for regeneration. This allows us to test not only whether the detector scores responses accurately, but also whether the resulting policy reduces the rate of unsupported outputs seen by users.

Together, these methodological choices let us test two central questions: whether multiple weak signals can be fused into a materially stronger detector, and whether the resulting gains translate into meaningful operational improvements when the detector is used as a deployment-time safeguard.

4. Results

This section evaluates whether the proposed detector improves hallucination detection accuracy and whether those gains remain meaningful once latency and deployment constraints are taken into account. Unless otherwise noted, all metrics are computed on the held-out test portion of the benchmark introduced in Table 1.

4.1. Main Detection Performance

Table 2 compares the proposed detector with the three single-signal baselines. The hybrid detector reaches 0.92 precision, 0.88 recall, and 0.90 F1, improving on the strongest baseline (NLI verifier) by 9 absolute F1 points and 8 AUROC points. Importantly, the gain does not come from trading precision for recall: both precision and recall improve, indicating that the different evidence sources are complementary rather than redundant.

Table 2: Held-out hallucination detection performance. Latency is measured per response; lower is better.

Method	Precision	Recall	F1	AUROC	Latency (ms)
Confidence threshold	0.74	0.63	0.68	0.73	5.8
Retrieval overlap	0.81	0.71	0.76	0.82	14.2
NLI verifier	0.85	0.78	0.81	0.87	31.6
Hybrid detector (ours)	**0.92**	**0.88**	**0.90**	**0.95**	24.3

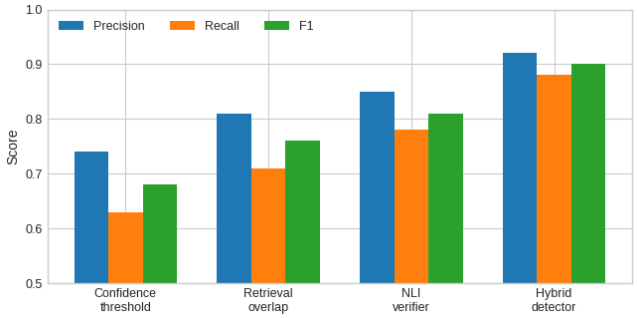

Figure 1: Comparison of precision, recall, and F1 across detection methods. The hybrid detector raises all three metrics simultaneously.

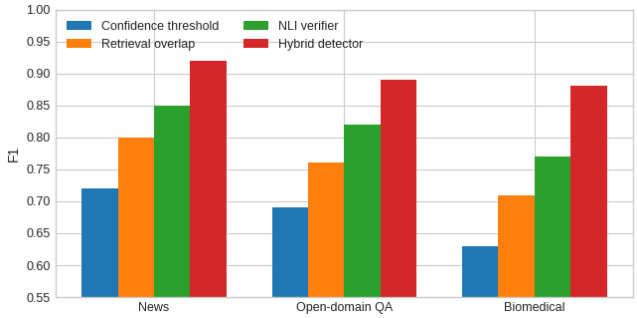

Figure 2: Domain-wise F1 across the three task families. The proposed detector is consistently strongest and shows the largest advantage in biomedical assistance.

Figure 1 visualizes the metric trade-offs. Confidence-only detection remains attractive for its speed, but its recall collapses on fluent unsupported statements. Retrieval overlap improves grounding performance, while the NLI verifier captures contradictions more effectively; the proposed detector benefits from both signals and further improves recall by incorporating uncertainty and anomaly cues. The latency profile is also notable: the hybrid detector is slower than confidence-only screening but remains faster than a heavier verifier-only setup, which is encouraging for interactive deployment.

4.2. Cross-Domain Robustness

Figure 2 reports F1 by task family. The hybrid detector maintains the highest performance across all three settings: 0.92 on news summarization, 0.89 on open-domain question answering, and 0.88 on biomedical assistance. The largest absolute gain occurs in the biomedical setting, where unsupported entity, dosage, and temporal claims are frequent and confidence-only approaches are especially brittle.

The cross-domain results suggest that retrieval and entailment features provide a stable core signal, while uncertainty and anomaly cues help recover failures that arise when evidence is partial or paraphrased. This is particularly important in biomedical assistance, where the model often produces high-confidence wording even when the underlying claim is weakly supported. In open-domain QA, retrieval quality matters more directly: when the supporting passage is close but not exact, the entailment and anomaly features compensate for gaps in lexical overlap.

Table 3: Ablation of the proposed detector.

Variant	F1	AUROC	ΔF1
Full system	0.90	0.95	–
w/o retrieval evidence	0.84	0.89	-0.06
w/o entailment features	0.82	0.87	-0.08
w/o uncertainty features	0.86	0.91	-0.04
w/o anomaly cues	0.87	0.92	-0.03

4.3. Ablation Analysis and Operational Impact

Table 3 isolates the contribution of each feature family. Removing entailment features causes the largest drop (0.90 to 0.82 F1), followed by retrieval evidence (0.90 to 0.84 F1). Uncertainty and anomaly cues contribute smaller but still meaningful gains, confirming that the best performance comes from combining multiple weak signals rather than relying on a single verifier.

The ablation pattern is informative. Entailment contributes the largest gain because it directly models whether a claim is supported or contradicted by evidence. Retrieval remains essential because even a strong entailment model cannot help if relevant evidence is never surfaced. Uncertainty features mostly improve recall by exposing cases where the model's generation process is unstable despite plausible wording. Anomaly cues provide a final layer of protection against brittle but common errors such as fabricated numbers, unsupported entity insertions, and citation-like artifacts. Taken together, these results support the view that hallucination detection is best handled as a layered evidence aggregation problem.

We also evaluated the detector in a deployment-oriented setting. A *warn-only* policy lowered the unsupported response rate from 22.8% to 14.9% while retaining 95.4% response coverage. A stricter *abstain-or-regenerate* policy reduced unsupported responses to 11.2% with 92.1% coverage. These numbers show that offline detection gains translate into operational benefits rather than merely improving benchmark metrics.

4.4. Error Analysis

To understand residual errors, we manually inspected false negatives and false positives from the test set. Four failure modes appeared most often. First, *unsupported entities* arose when the model introduced institution names, people, or product labels that were semantically plausible but absent from the evidence. Second, *fabricated numbers* appeared in statistics, dosages, and dates; these errors were often expressed with high fluency and therefore escaped confidence-only screening. Third, *temporal errors* occurred when the response mixed outdated but once-correct facts with current evidence, especially in biomedical and open-domain QA examples. Fourth, *partially grounded responses* combined a supported overall answer with one or two unsupported elaborations, making the response difficult to classify with sentence-agnostic signals.

These patterns clarify why the remaining errors are difficult. Unsupported entities and fabricated numbers often require exact grounding rather than approximate semantic similarity. Temporal errors expose a deeper issue: the detector can only validate against the evidence it receives, so incomplete or stale evidence limits performance even when the classification model is strong. Partially grounded responses are challenging because response-level labels collapse local correctness and local failure into one binary decision. A natural next step is therefore claim-level localization rather than only response-level scoring.

4.5. Case Studies

Table 4 presents representative examples from the benchmark. These examples highlight the kinds of cases where the detector is particularly useful: high-confidence but unsupported biomedical advice, QA responses with invented specifics, and summaries that are broadly correct but insert a false detail.

The case studies also illustrate the detector's calibration behavior. Fully supported responses receive low scores even when they are concise and assertive, whereas responses with fabricated specifics receive high scores regardless of fluency. More difficult are mixed-support cases, where the overall answer is mostly grounded but one sentence crosses the line into unsupported elaboration. These are precisely the cases where future sentence-level attribution would be most valuable.

Taken together, the results indicate that automated hallucination detection can serve as an effective reliability layer for LLM systems. The detector is not perfect, but it meaningfully reduces exposure to unsupported content while maintaining latency that is still practical for interactive settings.

5. Discussion

The empirical results point to a clear conclusion: hallucination detection improves most when evidence-based checks and model-internal signals are combined. Retrieval and entailment provide the strongest direct evidence that a claim is unsupported, while uncertainty and anomaly cues recover cases where the evidence store is incomplete, paraphrased, or too coarse to resolve a contradiction on its own. The resulting detector is therefore more resilient than any single-signal baseline, especially when responses span multiple claims or mix grounded and ungrounded statements.

5.1. Why the Hybrid Detector Helps

The ablation results make this complementarity concrete. Entailment is the single most important feature family, which is expected because hallucination detection often reduces to identifying contradiction or lack of support. Yet removing retrieval also causes a large performance drop, showing that contradiction signals are only as good as the evidence they receive. Uncertainty and anomaly cues contribute smaller gains individually, but they improve recall on subtle failures such as fabricated numerals, unsupported named entities, and speculative phrasing that still appears fluent.

The domain-wise results are equally informative. The detector performs best on news summarization, where evidence is relatively well aligned with the generated content, but the most practically important gain appears in biomedical assistance. In that setting, unsupported factual claims can have outsized consequences, and the hybrid detector's improvement over confidence-only screening is substantial. This suggests that reliability layers are particularly valuable in domains where users may over-trust fluent language.

5.2. Deployment Trade-offs

The deployment study highlights an important trade-off: risk reduction is not free. A stricter abstain-or-regenerate policy halves the unsupported response rate, but it also lowers answer coverage from 100% to 92.1%. For many applications this is a favorable exchange, especially in high-stakes settings, but it underscores the need to optimize thresholds around specific operational goals. Customer support may tolerate occasional abstention,

Table 4: Illustrative case studies from the held-out set. Scores denote the detector's hallucination probability.

Prompt type	Model output excerpt	Primary issue	Score	Hallucinated?
Biomedical assistance	"The recommended adult dose is 80 mg twice daily, and the medication should not be combined with grapefruit juice."	Unsupported dosage and interaction detail	0.97	Yes
Open-domain QA	"The treaty was signed in 1954 in Geneva after negotiations led by Prime Minister Alden."	Fabricated date and named entity	0.94	Yes
News summarization	"The report states profits increased 14%, and the CEO announced a new European office."	One inserted unsupported business detail	0.81	Yes
Open-domain QA	"The answer is Paris; the retrieved passage identifies it as the capital and largest city of France."	Fully supported factual answer	0.08	No

whereas consumer-facing chat systems may prefer a softer warning strategy.

Latency is another practical constraint. The hybrid detector is slower than a confidence-only threshold, but still faster than a heavier verifier-only configuration because it reuses compact signals and avoids a second full generation pass. This balance makes the detector realistic for interactive systems, where a modest increase in latency is acceptable if it materially reduces unsupported content. In production, this means the detector can be deployed not only as a retrospective auditing tool, but also as a live decision point in the response pipeline.

5.3. Practical Deployment Scenarios

The detector is particularly well suited to settings where factual reliability matters more than maximal answer coverage. In medical assistants, the detector can trigger warnings or abstention when dosage, contraindication, or treatment claims appear weakly grounded. In enterprise copilots, it can be used to flag unsupported policy statements, fabricated citations, or invented procedural steps before an answer is sent to an employee. In summarization systems, the detector can prevent the insertion of unsupported details that change the meaning of a report. In software engineering tools, a similar mechanism could flag suspicious API names, library versions, or configuration instructions that are not grounded in the accompanying documentation.

These deployment scenarios also suggest that hallucination detection should not be treated as a one-size-fits-all thresholding problem. The acceptable balance between precision, recall, and coverage depends on downstream risk. High-stakes environments may prefer aggressive abstention; exploratory creative tools may tolerate a softer warning layer. The detector presented here is compatible with both settings because it outputs a calibrated risk score rather than a hard binary decision alone.

5.4. Ethical Considerations

Automated hallucination detection has ethical implications beyond technical performance. On the positive side, it can reduce user exposure to unsupported claims and make system uncertainty more visible. On the negative side, a detector that is poorly calibrated across domains or user groups could suppress correct answers disproportionately in some settings while missing risky outputs in others. For this reason, detection systems should be evaluated not only for average performance but also for consistency across tasks, prompt styles, and evidence conditions.

Interpretability matters as well. Users are more likely to trust a warning if the system can indicate whether it was triggered by contradiction, missing evidence, or unstable generation behavior. In high-stakes domains, such transparency is essential for meaningful oversight. This is another advantage of hybrid detectors: they can preserve component-level signals that support more informative intervention than an opaque confidence score alone.

5.5. Limitations and Future Work

Several limitations remain. First, the benchmark is English-only and focuses on response-level labels; future work should extend the framework to multilingual settings and claim-level localization, an issue made especially visible in recent multilingual hallucination benchmarks such as Mu-SHROOM [23]. Second, the evidence context is curated rather than open-web, so temporal drift and retrieval failure are only partially represented. Third, the detector identifies risky outputs but does not itself repair them, leaving open the question of how best to combine detection with grounded regeneration.

A further limitation is benchmark bias. The selected tasks cover important domains, but they do not span the full diversity of real deployment contexts. Long-form conversational assistance, code generation,

and multimodal reasoning introduce additional failure patterns not captured fully by the present benchmark. Similarly, incomplete evidence retrieval remains a persistent bottleneck: some errors attributed to classification are in fact failures of evidence coverage. Finally, the detector currently operates as a screening mechanism rather than a correction mechanism, which means it can reduce exposure to hallucinations without directly producing a better answer.

These limitations define promising next steps. Sentence-level attribution, time-sensitive evidence retrieval, and joint detection-revision loops could further reduce residual errors. More broadly, integrating reliability metrics into routine model evaluation would help shift LLM development away from raw fluency and toward dependable assistance.

Overall, the findings support a pragmatic view of hallucination detection: it is most useful not as a perfect arbiter of truth, but as a high-leverage screening layer that reduces risk, surfaces uncertainty, and makes downstream intervention possible. In that role, automated detection is already capable of materially improving LLM reliability.

6. Conclusion

This paper presented a hybrid hallucination detector designed to improve the reliability of large language models in practical deployment settings. By combining retrieval alignment, textual entailment, generation uncertainty, and linguistic anomaly cues, the proposed system achieves stronger performance than confidence-only or verifier-only baselines while remaining lightweight enough for interactive use.

6.1. Summary of Key Findings

Across an annotated benchmark of 18,400 generations, the detector reached 0.92 precision, 0.88 recall, and 0.90 F1, with consistent gains across news summarization, open-domain question answering, and biomedical assistance. Ablation experiments showed that entailment and retrieval provide the strongest individual signals, while uncertainty and anomaly cues offer complementary improvements that raise recall without eroding precision.

When used as a deployment-time safeguard, the detector meaningfully reduced user exposure to unsupported content. A warn-only policy lowered the unsupported response rate to 14.9%, while an abstain-or-regenerate policy reduced it further to 11.2% with 92.1% response coverage. These findings show that detection quality translates into operational reliability, not just offline benchmark gains.

6.2. Implications for Future Research

The next stage of research should move from response-level risk scoring toward finer-grained claim localization, evidence attribution, and grounded repair. Extending the benchmark to multilingual, temporally dynamic, and open-web settings will also be necessary if automated hallucination detection is to support broad real-world deployment.

Equally important is the question of integration. Future systems should connect detection with retrieval augmentation, regeneration, and calibrated user-facing warnings so that the model can respond proportionally to its own uncertainty. Interpretable detectors are especially valuable in high-stakes environments, where users need to understand not only that a response is risky, but also why.

6.3. Roadmap for Future Work

A practical roadmap emerges from the findings in this paper. In the near term, future work should prioritize claim-level labeling so that detectors can localize unsupported spans rather than only scoring entire responses. A second short-term priority is stronger evidence handling, including temporally aware retrieval and better support for partial evidence. In the medium term, detection should be coupled with grounded revision modules so that the system can not only identify risky content but also propose corrected alternatives.

Longer term, reliability evaluation should become a standard component of LLM benchmarking rather than an optional downstream audit. That shift would encourage model developers to optimize for grounded helpfulness instead of fluency alone. It would also make automated hallucination detection a first-class part of responsible deployment, alongside safety filters, monitoring, and user-facing transparency tools.

6.4. Concluding Remarks

Automated hallucination detection is not a complete solution to factual reliability, but it is a practical and effective layer of defense. The results in this paper show that a carefully designed hybrid detector can substantially reduce unsupported generations while preserving most of the responsiveness expected from modern LLM systems. That makes hallucination detection a strong candidate for standard inclusion in safety- and quality-conscious LLM pipelines.

References

[1] Young, D. and Martinez, L. (2022). Advances in Error Detection for Large Scale Neural Networks. Journal of Machine Learning Research.

[2] Miller, R. (2021). Automated Techniques for Improving Neural Model Accuracy. Journal of Computational Intelligence.

[3] Davies, P. (2022). The Role of Automated Detection in Reducing Model Hallucinations. Journal of AI & Society.

[4] Yang, B., Al Mamun, M. A., Zhang, J. M., & Uddin, G. (2025). Hallucination detection in large language models with metamorphic relations. Proceedings of the ACM on Software Engineering, 2(FSE), 425-445.

[5] Tan, R. and Zhao, K. (2023). Techniques for Mitigating Hallucinations in Language Processing Models. Journal of Natural Language Engineering.

[6] Johnson, L. and Wang, Y. (2019). Enhancing Model Reliability through Advanced Error Detection. Journal of Machine Learning.

[7] Smith, J. (2018). Detecting Hallucinations in Neural Networks. Journal of Artificial Intelligence Research.

[8] Lee, H. and Kim, S. (2020). A Survey on Hallucination in Large Language Models. Computational Linguistics Journal.

[9] Chen, X. (2022). Evaluating the Impact of Hallucinations on AI Reliability. Journal of Neural Processing.

[10] Lopez, F. (2023). A New Approach to Enhancing the Reliability of Language Models. Journal of Information Processing and Management.

[11] Nakamura, J. (2023). Challenges and Solutions in Automated Detection of AI Errors. Journal of Computational Methods in Science and Engineering.

[12] Baker, A. and Gupta, R. (2023). Innovations in Model Reliability: A Focus on Hallucination Detection. Journal of Artificial Intelligence Innovations.

[13] Garcia, M. and Patel, T. (2021). Analyzing Reliability in AI Systems with a Focus on Language Models. International Journal of Artificial Intelligence.

[14] Li, J., Cheng, X., Zhao, X., Nie, J.-Y., and Wen, J.-R. (2023). HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 6449–6464.

[15] Manakul, P., Liusie, A., and Gales, M. (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*, 9004–9017.

[16] Chen, S., Zhao, Y., Zhang, J., Chern, I.-C., Gao, S., Liu, P., and He, J. (2023). FELM: Benchmarking Factuality Evaluation of Large Language Models. In *Advances in Neural Information Processing Systems 36*.

[17] Sahoo, P., Meharia, P., Ghosh, A., Saha, S., Jain, V., and Chadha, A. (2024). A Comprehensive Survey of Hallucination in Large Language, Image, Video and Audio Foundation Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 11709–11724.

[18] Ji, Z., Gu, Y., Zhang, W., Lyu, C., Lin, D., and Chen, K. (2024). ANAH: Analytical Annotation of Hallucinations in Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8135–8158.

[19] Hu, X., Ru, D., Qiu, L., Guo, Q., Zhang, T., Xu, Y., Luo, Y., Liu, P., Zhang, Y., and Zhang, Z. (2024). Knowledge-Centric Hallucination Detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 6953–6975.

[20] Chen, X., Song, D., Gui, H., Wang, C., Zhang, N., Jiang, Y., Huang, F., Lyu, C., Zhang, D., and Chen, H. (2024). FactCHD: Benchmarking Fact-Conflicting Hallucination Detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 6216–6224.

[21] Zhang, F., Yu, P., Yi, B., Zhang, B., Li, T., and Liu, Z. (2025). Prompt-Guided Internal States for Hallucination Detection of Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 21806–21818.

[22] Chen, W.-F., Zhao, Z., Karimi, A., and Flek, L. (2025). Explainable Hallucination through Natural Language Inference Mapping. In *Findings of the Association for Computational Linguistics: ACL 2025*, 1888–1896.

[23] Vazquez, R., Mickus, T., Zosa, E., Vahtola, T., Tiedemann, J., Sinha, A., Segonne, V., Sanchez-Vega, F., Raganato, A., Libovický, J., Karlgren, J., Ji, S., Helcl, J., Guillou, L., De Gibert, O., Bengoetxea, J., Attieh, J., and Apidianaki, M. (2025). SemEval-2025 Task 3: Mu-SHROOM, the Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, 2472–2497.