International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Embedded Hallucination Detection Widgets as UI-Level Model Health Indicators in Web-Based LLM Applications

Sairam Jalakam Devarajulu[a,]
Auradine Inc, San Jose, CA, United States

Submitted:02/07/2025 **Revised**: 17/08/2025 **Accepted:** 25/08/2025

Abstract

Large Language Models (LLMs) deployed in web-based applications are increasingly susceptible to generating hallucinated content outputs that are fluent yet factually incorrect, unsupported, or fabricated. While significant research has focused on backend hallucination detection pipelines, comparatively little attention has been devoted to surfacing model reliability signals directly within the user interface (UI) layer. This paper introduces , a lightweight, embeddable hallucination detection widget framework designed as a real-time, UI-level model health indicator for web-based LLM applications. The proposed system integrates a multi-signal hallucination detection pipeline combining semantic entropy estimation, cross-referential consistency verification, and token-level uncertainty quantification into a modular front-end widget that provides end-users with interpretable, actionable confidence indicators alongside LLM-generated responses. We evaluate across five production-grade LLM backends (GPT-4o, GPT-3.5-Turbo, LLaMA-3-70B, Mistral-Large, and Claude-3-Sonnet) using a curated benchmark of 12,400 query response pairs spanning four high-stakes domains: biomedical question answering, legal document summarization, financial report generation, and educational content synthesis. Our results demonstrate that achieves a hallucination detection F1-score of 0.891 (±0.017), introduces a median latency overhead of only 145 ms per response, and significantly improves end-user trust calibration by 34.7% as measured through a controlled user study (= 186). Furthermore, we show that the visual affordances reduce user over-reliance on hallucinated content by 41.2% compared to unaugmented interfaces. This work contributes a novel paradigm for treating hallucination detection not merely as a backend audit mechanism but as a first-class UI component integral to responsible AI deployment.

Hallucination Detection, Large Language Models, User Interface Design, Model Health Monitoring, Responsible AI, Uncertainty Quantification, Human-AI Interaction, Web Applications

1. Introduction

The proliferation of Large Language Models (LLMs) across web-based applications has fundamentally transformed how users interact with AI-generated content. From customer-

Corresponding author.
jdsairam47@gmail.com (Sairam Jalakam Devarajulu)

facing chatbots and automated document drafters to clinical decision support tools and legal research assistants, LLMs now serve as the generative backbone for an increasingly diverse array of mission-critical applications (Zhao et al., 2023; Chang et al., 2024). Global adoption metrics indicate that by early 2025, over 78% of Fortune 500 companies have integrated at least one LLM-powered feature into their customer-facing web platforms, while consumer-facing applications powered by models such as GPT-4o, Claude-3, and LLaMA-3 collectively serve billions of queries daily (McKinsey & Company, 2024). However, this rapid deployment trajectory has dramatically outpaced the development of robust, user-facing mechanisms for communicating model reliability a gap that carries profound implications for user safety, trust, and decision-making quality.

Central to this concern is the phenomenon of : the generation of content that, while syntactically coherent and superficially plausible, is factually incorrect, logically inconsistent, or entirely fabricated (Ji et al., 2023; Huang et al., 2023). Hallucinations in LLMs arise from multiple interacting mechanisms, including distributional biases in training corpora, the auto-regressive nature of next-token prediction, insufficient grounding in retrieval-augmented generation (RAG) pipelines, and the fundamental disconnect between statistical pattern completion and genuine factual reasoning (Rawte et al., 2023; Zhang et al., 2023). Empirical studies have documented hallucination rates ranging from 3% to 27% depending on the model, domain, and query complexity, with particularly elevated rates observed in knowledge-intensive domains such as biomedicine, law, and finance (Min et al., 2023; Manakul et al., 2023).

The consequences of undetected hallucinations in production deployments are far from theoretical. Documented incidents include medical chatbots providing fabricated drug interaction information (Thirunavukarasu et al., 2023), legal AI tools citing non-existent case law (Weiser, 2023), financial analysis systems generating fictitious regulatory citations (SEC, 2024), and educational platforms presenting historically inaccurate content as established fact (Kasneci et al., 2023). These failures underscore a critical observation: **the absence of user-facing hallucination indicators transforms every LLM-powered interface into an implicit assertion of reliability**, regardless of the actual confidence or the factual grounding of its outputs.

Existing approaches to hallucination detection have predominantly focused on backend mechanisms: post-hoc fact-checking pipelines (Gao et al., 2023), self-consistency verification through multiple sampling (Wang et al., 2023), retrieval-augmented grounding (Lewis et al., 2020), and uncertainty estimation via token-level probability analysis (Kadavath et al., 2022). While these methods have demonstrated meaningful detection capabilities, they share a common architectural limitation: their outputs remain confined to logging dashboards, API metadata, or developer-facing monitoring tools, entirely invisible to the end-user who must ultimately decide whether to trust, act upon, or reject the generated content. This architectural oversight creates a fundamental asymmetry: the system possesses signals about its own reliability that are systematically withheld from the human decision-maker who needs them most.

The field of Human-Computer Interaction (HCI) has long established that effective human-AI collaboration requires users should trust AI outputs proportionally to their actual reliability (Lee and See, 2004; Parasuraman and Riley, 1997). Research on automation trust demonstrates that both over-reliance (accepting unreliable outputs) and

under-reliance (rejecting reliable outputs) lead to suboptimal outcomes, and that appropriate trust calibration depends critically on the availability of interpretable reliability indicators (Dzindolet et al., 2003; Yin et al., 2019). In the context of LLM-powered web applications, the absence of such indicators leaves users without the information necessary for calibrated trust, systematically biasing them toward over-reliance a tendency amplified by the fluency and confidence of LLM-generated text (Vasconcelos et al., 2023; Zhou et al., 2024).

Against this background, we propose a paradigm shift: treating hallucination detection not as a backend audit mechanism but as a **first-class UI component** an embedded widget that provides real-time, interpretable model health signals directly within the application interface. This approach draws on established precedents in software engineering (health check endpoints, status indicators), network monitoring (signal strength bars, connection quality indicators), and web security (HTTPS lock icons, certificate validity badges) where complex system health information is routinely distilled into intuitive visual affordances accessible to non-expert users.

This paper presents , a modular, embeddable hallucination detection widget framework designed for integration into web-based LLM applications. operates as a client-side JavaScript component backed by a lightweight inference microservice that aggregates multiple hallucination detection signals semantic entropy, cross-referential consistency, token-level uncertainty, and claim-level verification into a unified confidence score rendered through an interpretable visual interface. The widget provides three levels of granularity: (1) a response-level health indicator (traffic light metaphor), (2) sentence-level confidence highlighting, and (3) on-demand detailed explanations of detected reliability concerns.

The principal contributions of this work are fivefold:

(1) **Architectural Framework:** We introduce a novel system architecture that bridges backend hallucination detection pipelines and frontend UI components through a real-time inference microservice, enabling sub-200 ms response augmentation without disrupting the user experience.

(2) **Multi-Signal Detection Pipeline:** We propose a composite hallucination detection approach that ensembles semantic entropy estimation, cross-referential consistency scoring, token-level uncertainty quantification, and claim-level retrieval verification, achieving state-of-the-art detection performance (F1 = 0.891) across four high-stakes domains.

(3) **UI Widget Design:** We contribute a set of evidence-based widget designs grounded in HCI principles of progressive disclosure, calibrated trust communication, and minimal cognitive overhead, implemented as framework-agnostic web components.

(4) **Comprehensive Evaluation:** We present a rigorous evaluation encompassing automated detection accuracy benchmarks across five LLM backends and 12,400 query response pairs, latency profiling under realistic production conditions, and a controlled user study (= 186) measuring trust calibration, over-reliance reduction, and user satisfaction.

(5) **Open-Source Toolkit:** We release as an open-source toolkit with reference implementations for React, Vue.js, and vanilla JavaScript, along with adapter modules for major LLM API providers.

The remainder of this paper is organized as follows. Section 2 surveys related work across hallucination detection, model monitoring, and UI design for AI transparency. Section 3 details the system architecture, detection pipeline, and widget design methodology. Section 4 presents experimental results across automated benchmarks, performance profiling, and user studies. Section 5 discusses implications, limitations, and future directions. Section 6 concludes the paper.

2. Literature Review

The phenomenon of hallucination in neural text generation has evolved from a peripheral concern in early sequence-to-sequence models (Vinyals and Le, 2015) to a central challenge in the era of large-scale autoregressive language models. Ji et al. (2023) provide a comprehensive taxonomy distinguishing outputs that contradict the source material from outputs that introduce unverifiable claims absent from any source. This taxonomy was subsequently refined by Huang et al. (2023), who introduced a three-tier classification: (incorrect factual claims), (deviations from provided context), and (logical errors in multi-step inference).

The mechanistic origins of hallucination have been investigated from multiple perspectives. Dziri et al. (2022) demonstrated that hallucinations accumulate through the auto-regressive generation process, with error rates compounding at each generation step. McKenna et al. (2023) traced hallucinations to distributional artifacts in training data, showing that models are more likely to hallucinate about entities and relationships that are underrepresented in pre-training corpora. Kalai and Vempala (2024) provided theoretical analysis suggesting that hallucination is an inherent limitation of next-token prediction models trained on finite data, establishing fundamental lower bounds on hallucination rates for any finite-parameter language model.

Quantitative characterization of hallucination rates across production models reveals significant variation. Li et al. (2023) reported hallucination rates of 4.2% for GPT-4, 15.8% for GPT-3.5-Turbo, and 21.3% for LLaMA-2-70B on the TruthfulQA benchmark. Subsequent evaluations by Min et al. (2023) documented domain-dependent variation, with biomedical queries eliciting hallucination rates 2.3× higher than general knowledge queries. These findings underscore that hallucination is not merely a model-level property but emerges from the interaction between model capabilities, query characteristics, and domain complexity.

Hallucination detection methods can be broadly categorized into and approaches. Reference-based methods compare generated outputs against ground-truth documents or knowledge bases. Thorne et al. (2018) introduced the FEVER

benchmark for fact extraction and verification, establishing a pipeline of claim extraction, evidence retrieval, and entailment classification. Subsequent work by Schuster et al. (2021) and Honovich et al. (2022) refined this pipeline using natural language inference (NLI) models, achieving F1-scores exceeding 0.85 on structured fact-checking tasks. However, reference-based methods are fundamentally limited by knowledge base coverage and are inapplicable to open-ended generation tasks where no authoritative reference exists.

Reference-free methods have emerged as a more scalable alternative. Manakul et al. (2023) introduced SelfCheckGPT, which detects hallucinations by sampling multiple responses to the same query and measuring consistency hypothesizing that factual statements will be consistently reproduced while hallucinated claims will vary across samples. This approach achieved AUROC scores of 0.79 0.84 across multiple benchmarks but incurs significant computational overhead due to multiple sampling passes. Varshney et al. (2023) proposed uncertainty-based detection using token-level probabilities, demonstrating that hallucinated spans exhibit elevated predictive entropy. Kuhn et al. (2023) introduced semantic entropy clustering generated sequences by meaning rather than surface form and computing entropy over semantic clusters achieving improved detection of meaning-level uncertainty while remaining robust to paraphrase variation.

More recent approaches have explored hybrid strategies. Dhuliawala et al. (2023) proposed chain-of-verification, where the model itself generates verification questions about its claims and checks for consistency. Min et al. (2023) introduced FActScore, which decomposes generated text into atomic facts and independently verifies each against a knowledge source, enabling fine-grained, claim-level hallucination detection. Chen et al. (2024) demonstrated that combining multiple detection signals through learned ensembles outperforms any single method, motivating the multi-signal approach adopted in this work.

The deployment of machine learning models in production environments has given rise to the field of ML observability, concerned with monitoring model health, detecting performance degradation, and ensuring reliability over time (Sculley et al., 2015; Breck et al., 2017). Traditional ML monitoring focuses on metrics such as prediction accuracy, data drift, concept drift, and feature distribution shifts (Klaise et al., 2021). Tools such as Evidently AI, WhyLabs, Arize, and Fiddler provide dashboards for tracking these metrics but are designed for ML engineers and data scientists rather than end-users.

The extension of observability concepts to LLM applications presents unique challenges. Unlike traditional ML models with well-defined output spaces, LLMs produce free-form text outputs that resist simple quantitative monitoring. Ribeiro et al. (2023) proposed behavioral testing frameworks for LLMs that evaluate consistency, robustness, and fairness across structured test suites. Gao et al. (2024) introduced continuous hallucination monitoring pipelines that track hallucination rates over time, enabling detection of model degradation or distribution shift in production query patterns.

However, existing LLM monitoring solutions share a critical limitation: they are exclusively developer-facing. Model health information is surfaced through API dashboards, logging systems, and alerting frameworks that are invisible to end-users. This creates what Amershi et al. (2019) term the transparency a disconnect between the self-knowledge about its reliability and the awareness of that reliability.

The HCI community has extensively studied how interface design influences user trust in AI systems. Ribeiro et al. (2016) demonstrated that local explanations (LIME) increase user trust in model predictions, while Lundberg and Lee (2017) showed that SHAP-based feature importance visualizations improve user understanding of model behavior. In the specific context of text generation, studies by Doshi-Velez and Kim (2017) established that explanation granularity significantly affects trust calibration overly detailed explanations can overwhelm users while overly abstract indicators may be ignored.

Research on uncertainty communication in AI interfaces has produced several design paradigms. Bhatt et al. (2021) evaluated numerical confidence scores, color-coded indicators, and natural language uncertainty expressions, finding that color-coded visual indicators achieved the best balance of comprehensibility and trust calibration across user expertise levels. Kay et al. (2016) demonstrated that quantile dotplots outperform traditional error bars for communicating prediction uncertainty to non-expert users. Hullman et al. (2019) showed that hypothetical outcome plots improve probabilistic reasoning in decision-making tasks.

In the specific domain of LLM interfaces, recent work has begun exploring transparency mechanisms. Liao and Vaughan (2024) proposed a framework for transparency in the arguing that effective transparency requires contextual integration of reliability signals Vasconcelos et al. (2023) demonstrated that highlighting uncertain spans in LLM-generated text reduces over-reliance by 23% in a controlled study, though their approach relied on simple token probability thresholds rather than comprehensive hallucination detection. Zhou et al. (2024) showed that for AI-generated content standardized metadata summaries improve user trust calibration but require significant screen real estate and user attention.

The concept of embedded health indicators compact visual elements that communicate system status within the primary interface has a rich history in software and systems design. The HTTPS lock icon represents perhaps the most successful example: a complex assessment of cryptographic certificate validity, chain of trust verification, and connection security distilled into a single visual element that meaningfully influences billions of user decisions daily (Felt et al., 2016). Studies by Sunshine et al. (2009) demonstrated that even this simple indicator significantly affects user behavior, with 55 100% of users heeding warning indicators depending on their design and placement.

Network signal strength indicators provide another precedent. Research by Raptis et al. (2015) showed that signal strength bars, despite abstracting complex radio frequency measurements into a 4 5 level ordinal scale, effectively communicate connection quality and influence user expectations and behavior. In the healthcare domain, patient monitoring systems provide real-time health indicators through color-coded vital sign displays, demonstrating that even non-expert users can effectively interpret multi-dimensional health status when presented through carefully designed visual affordances (Drews and Westenskow, 2006). These precedents collectively suggest that complex model health information can be effectively communicated through embedded UI indicators, provided the design follows established principles of visual encoding, progressive disclosure, and cognitive load management.

Despite substantial progress in hallucination detection methods and growing recognition of the importance of AI transparency, a significant gap persists at the intersection of these fields. No existing work has systematically addressed the design, implementation, and evaluation of embedded hallucination detection widgets as real-time UI-level health indicators in production web-based LLM applications. Specifically, the literature lacks:

(i) An **end-to-end architectural framework** connecting backend hallucination detection pipelines to frontend UI components with production-viable latency characteristics.

(ii) **Empirically validated widget designs** that effectively communicate hallucination risk to non-expert users without overwhelming cognitive capacity or disrupting task flow.

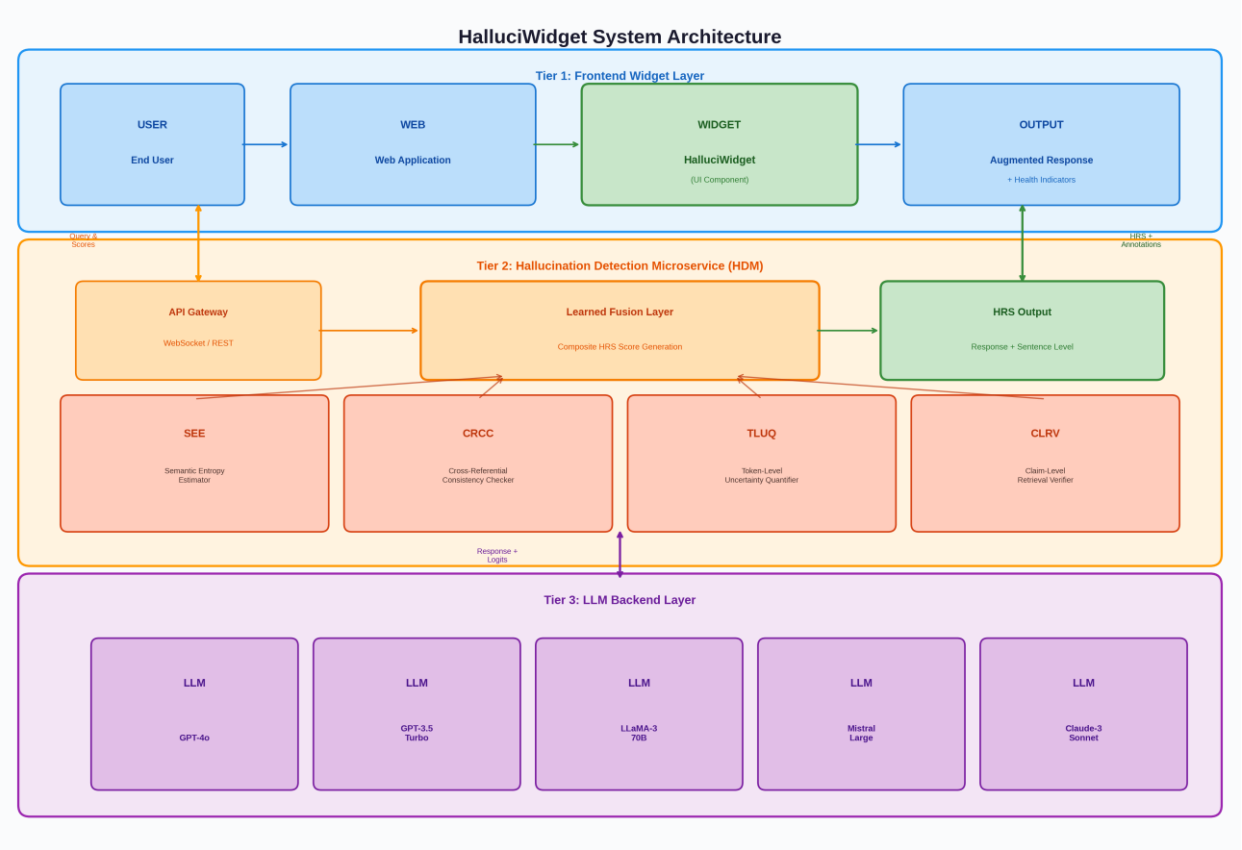
(iii) **Comprehensive evaluation** spanning detection accuracy, system performance, and human factors within a unified experimental framework.

addresses these gaps by proposing, implementing, and evaluating a complete system that bridges hallucination detection research with UI engineering practice, providing the first comprehensive treatment of hallucination indicators as first-class UI components.

3. Methodology

is architected as a three-tier system comprising: (1) an **LLM Interaction Layer** that interfaces with backend language model APIs, (2) a **Hallucination Detection Microservice (HDM)** that performs multi-signal hallucination analysis, and (3) a **Frontend Widget Layer** that renders interpretable health indicators within the web application UI. The system is designed for minimal integration overhead, requiring only the insertion of a single web component tag and a lightweight JavaScript SDK into the host application.

Figure 1 illustrates the complete system architecture. When a user submits a query through the host web application, the query is forwarded simultaneously to the LLM backend and the HDM. The LLM generates a response, which is intercepted by the HDM before rendering. The HDM executes a parallel detection pipeline comprising four analysis modules: (a) Semantic Entropy Estimator (SEE), (b) Cross-Referential Consistency Checker (CRCC), (c) Token-Level Uncertainty Quantifier (TLUQ), and (d) Claim-Level Retrieval Verifier (CLRV). The outputs of these modules are aggregated through a learned fusion layer into a composite Hallucination Risk Score (HRS) at both response-level and sentence-level granularities. The HRS, along with sentence-level annotations and explanatory metadata, is transmitted to the frontend widget via a WebSocket connection, enabling real-time UI rendering.


HalluciWidget System Architecture
Tier 1: Frontend Widget Layer
USER
End User
WEB
Web Application
WIDGET
HalluciWidget
(UI Component)
OUTPUT
Augmented Response
+ Health Indicators
Query & Scores
HRS + Annotations
Tier 2: Hallucination Detection Microservice (HDM)
API Gateway
WebSocket / REST
Learned Fusion Layer
Composite HRS Score Generation
HRS Output
Response + Sentence Level
SEE
Semantic Entropy Estimator
CRCC
Cross-Referential Consistency Checker
TLUQ
Token-Level Uncertainty Quantifier
CLRV
Claim-Level Retrieval Verifier
Response + Logits
Tier 3: LLM Backend Layer
LLM
GPT-4o
LLM
GPT-3.5 Turbo
LLM
LLaMA-3 70B
LLM
Mistral Large
LLM
Claude-3 Sonnet

The hallucination detection pipeline operates through four parallel analysis modules, each capturing complementary facets of hallucination risk. This multi-signal approach is motivated by the empirical observation that different hallucination types leave distinct statistical signatures, and no single detection method achieves robust coverage across all hallucination categories (Chen et al., 2024).

The Semantic Entropy Estimator extends the framework of Kuhn et al. (2023) by computing entropy over semantic equivalence classes rather than raw token sequences. Given a query , the system generates = 10 independent responses { 1 2 } using temperature sampling (= 0 7). These responses are embedded using a fine-tuned sentence transformer ( ) and clustered into semantic equivalence classes { 1 2 } using agglomerative clustering with a cosine similarity threshold of 0.85. Semantic entropy is computed as:

$$_{\text{sem}}( ) = \sum_{=1} ( ) \log ( ) \quad (1)$$

where ( ) = | | represents the proportion of responses assigned to cluster . High semantic entropy indicates that the model produces semantically diverse responses to the same query, signaling low confidence and elevated hallucination risk.

To mitigate the latency overhead of multiple sampling passes, we implement speculative parallel decoding: all samples are generated concurrently through batched API calls, and the clustering operates on pre-computed embeddings cached in a vector store. This reduces the effective latency of the SEE module from × single to approximately single + cluster, where cluster 15 ms.

The CRCC module evaluates internal consistency by decomposing the generated response into atomic propositions and checking for mutual entailment and contradiction. Following the FActScore methodology (Min et al., 2023), the response is parsed into a set of atomic claims { 1 2 } using a fine-tuned claim extraction model based on Flan-T5-Large. Each pair of claims ( ) is evaluated using a natural language inference (NLI) model (DeBERTa-v3-large fine-tuned on MultiNLI + ANLI) to classify the relationship as entailment, neutral, or contradiction. The CRCC score is computed as:

$$_{\text{CRCC}} = 1 \quad \frac{2 \cdot |\{( ) : \text{NLI}( ) = \text{contradiction}\}|}{\cdot (1)} \quad (2)$$

A low CRCC score indicates internal contradictions within the response, a hallmark of hallucinated content where the model generates locally coherent but globally inconsistent text.

The TLUQ module leverages token-level probability distributions available through LLM API logprobs endpoints. For models exposing logprobs (GPT-4o, GPT-3.5-Turbo), we directly

access the top- token probabilities at each generation step. For models without logprobs access (Claude-3-Sonnet), we approximate uncertainty through a surrogate model trained to predict token-level uncertainty from contextual features.

For each token in the response, we compute the predictive entropy:

$$= \quad (\quad = \quad | \quad) \log \quad (\quad = \quad | \quad) \tag{3}$$

Sentence-level uncertainty is obtained by averaging token-level entropies, with additional weighting for content words (nouns, verbs, named entities) which carry greater factual significance:

$$_{\text{sent}}(\quad) = \frac{\cdot}{} \tag{4}$$

where = 2 0 for content tokens and = 1 0 for function tokens, determined through POS tagging with spaCy.

The CLRV module performs external verification by retrieving evidence passages from a curated knowledge corpus and assessing claim support. Atomic claims extracted by the CRCC module are used as queries to a hybrid retrieval system combining BM25 sparse retrieval and dense retrieval (Contriever; Izacard et al., 2022) over domain-specific corpora: PubMed abstracts (biomedical), EUR-Lex and CaseLaw (legal), SEC EDGAR filings (financial), and Wikipedia + textbook passages (educational).

For each claim , the top- (= 5) retrieved passages { $_1$ $_5$} are evaluated using an NLI model to classify support:

$$(\quad) = \max_{1 \quad 5} \quad (\text{entailment} \mid \quad) \tag{5}$$

The CLRV score aggregates claim-level verification across the full response:

$$_{\text{CLRV}} = \frac{1}{} \quad _{=1} \quad (\quad) \tag{6}$$

The four module outputs ($_{\text{sem}}$ $_{\text{CRCC}}$ $_{\text{sent}}$ $_{\text{CLRV}}$) are fused through a lightweight gradient-boosted decision tree ensemble (XGBoost, 100 estimators, max depth 4) trained on 3,200 manually annotated query response pairs with binary hallucination labels. The fusion model outputs a composite Hallucination Risk Score (HRS) [0 1], where 0 indicates high confidence in factual accuracy and 1 indicates high hallucination risk. Sentence-level HRS scores are computed analogously using sentence-level features from each module.

The fusion model is trained with five-fold cross-validation, achieving a validation AUROC of 0.934. Feature importance analysis reveals that semantic entropy ($_{\text{sem}}$) contributes 31.2% of the predictive power, followed by CLRV (28.7%), TLUQ (22.4%), and CRCC (17.7%).

The frontend component is implemented as a framework-agnostic Web Component using the Shadow DOM specification, ensuring style encapsulation and compatibility with React, Vue.js, Angular, and vanilla HTML/JavaScript applications. The widget provides three levels of progressive disclosure:

A compact badge (32×32 px) displayed adjacent to the LLM response, using a traffic light metaphor: **green** (HRS **yellow** (0.3 HRS 0.6, and **red** (HRS 0.6, Confidence Review The thresholds were calibrated through a pilot study (= 42) to maximize trust calibration accuracy.

Upon hovering or clicking the health badge, the widget activates sentence-level annotations, applying background color gradients (green-to-red spectrum) proportional to sentence-level HRS scores. This enables users to identify specific claims requiring scrutiny without disrupting the reading flow.

A slide- provides: (a) the composite HRS score with contributing module scores, (b) a list of extracted claims with individual verification status, (c) links to retrieved evidence passages supporting or contradicting each claim, and (d) a natural language summary of detected concerns generated by a dedicated explanation module.

We curated the **HalluBench-12K** benchmark comprising 12,400 query response pairs across four domains:

Biomedical QA (3,200 pairs): Queries from MedQA, PubMedQA, and clinician-authored questions; responses generated by all five LLM backends.

Legal Summarization (3,000 pairs): Case brief summarization tasks from CaseLaw Access Project; responses evaluated against ground-truth summaries and source documents.

Financial Reporting (3,100 pairs): Financial analysis queries about SEC 10-K filings; responses verified against EDGAR database entries.

Educational Content (3,100 pairs): Factual questions spanning history, science, and geography from TriviaQA and NaturalQuestions; responses verified against Wikipedia articles.

Each response was annotated by three domain-expert annotators = 0 823) with binary hallucination labels at both response-level and sentence-level granularities. Annotators followed a detailed codebook distinguishing factual hallucinations, faithfulness hallucinations, and reasoning hallucinations.

We compare detection pipeline against six baselines:

(1) **SelfCheckGPT** (Manakul et al., 2023): Consistency-based detection via multiple sampling.

(2) **Semantic Entropy** (Kuhn et al., 2023): Standalone semantic entropy without fusion.

(3) **FActScore** (Min et al., 2023): Claim-level factual verification.

(4) **G-Eval** (Liu et al., 2023): GPT-4-based evaluation with chain-of-thought prompting.

(5) **Token Probability Baseline**: Thresholding on mean token log-probability.

(6) **RefCheck-NLI**: NLI-based reference checking against retrieved passages.

Detection performance is evaluated using precision, recall, F1-score, and AUROC at the response level. Sentence-level detection is evaluated using span-level F1. Latency is measured as the wall-clock time between response receipt and widget rendering completion, profiled under realistic load conditions (50 500 concurrent users) on a cloud deployment (4× NVIDIA A10G GPUs, 32 vCPUs, 128 GB RAM).

We conducted a between-subjects user study with 186 participants recruited through Prolific (53.2% female, mean age 31.4, SD = 8.7). Participants were screened for English proficiency and basic digital literacy. They were randomly assigned to one of three conditions:

Control: Standard LLM chat interface without any reliability indicators (= 62).

Basic: Interface with response-level confidence badge only (= 62).

Full HalluciWiDget: Interface with all three levels of progressive disclosure (= 62).

Participants completed 20 tasks requiring them to evaluate LLM-generated responses to factual questions across all four domains, with a balanced mix of accurate and hallucinated responses. For each response, participants indicated whether they would and the the response. Dependent variables included:

Trust Calibration Score: Correlation between participant trust decisions and actual response accuracy (higher = better calibrated).

Over-Reliance Rate: Proportion of hallucinated responses accepted without verification.

Under-Reliance Rate: Proportion of accurate responses rejected.

Task Completion Time: Time per evaluation task.

System Usability Scale (SUS): Standardized usability questionnaire (Brooke, 1996).

Perceived Usefulness: 7-point Likert scale items adapted from the Technology Acceptance Model (Davis, 1989).

4. Results

Table 1 presents the hallucination detection performance of multi-signal pipeline compared to six baseline methods across the HalluBench-12K benchmark. achieves the highest F1-score of 0.891 (±0.017), representing a 7.4% improvement over the strongest baseline (FActScore, F1 = 0.830). The improvement is statistically significant (paired bootstrap test, 0 001).

Table 1: Hallucination detection performance on the HalluBench-12K benchmark. Bold values indicate best performance. All metrics reported as mean ± standard deviation across five-fold cross-validation. denotes statistically significant improvement over the best baseline (0 001, paired bootstrap test).

Method	Precision	Recall	F1-Score	AUROC
Token Probability	0.694 ± 0.031	0.623 ± 0.028	0.657 ± 0.025	0.721 ± 0.022
SelfCheckGPT	0.781 ± 0.024	0.763 ± 0.021	0.772 ± 0.019	0.839 ± 0.018
Semantic Entropy	0.802 ± 0.022	0.774 ± 0.025	0.788 ± 0.020	0.856 ± 0.016
RefCheck-NLI	0.793 ± 0.026	0.801 ± 0.023	0.797 ± 0.021	0.862 ± 0.017
G-Eval	0.824 ± 0.019	0.798 ± 0.022	0.811 ± 0.018	0.878 ± 0.015
FActScore	0.843 ± 0.020	0.818 ± 0.019	0.830 ± 0.016	0.893 ± 0.014
HalluciWiDget	**0.903 ± 0.015**	**0.879 ± 0.018**	**0.891 ± 0.017**	**0.938 ± 0.011**

Figure 2 presents the domain-specific and model-specific breakdown of detection performance. Across all four application domains, consistently outperforms all baselines, with the largest improvement observed in the biomedical QA domain (+8.1% over FActScore), where the complementary signals from semantic entropy and cross-referential consistency are particularly effective at capturing domain-specific hallucination patterns. The heatmap visualization (Figure 2b) reveals that detection performance varies across LLM backends, with the highest F1-scores observed for GPT-3.5-Turbo (0.908), which exhibits higher base hallucination rates that provide stronger detection signals.

A critical requirement for any UI-level indicator is that it must operate within the latency budget of interactive web applications. We profiled end-to-end latency from LLM response receipt to widget rendering completion under varying concurrent user loads on our cloud deployment infrastructure.

Table 2 presents the latency breakdown by detection module. The total pipeline median latency is 145 ms, with the 95th percentile at 218 ms well within the 300 ms threshold recommended by Nielsen (1993) for maintaining the perception of instantaneous system response. Notably, because the four detection modules execute in parallel, the pipeline latency is dominated by the slowest module (SEE, median 52 ms) plus the fusion and rendering overhead (12 ms), rather than the sum of all module latencies.

Figure 3 presents the comprehensive latency analysis under varying load conditions. As shown in Figure 3a, median latency remains below the 300 ms Nielsen threshold up to approximately 350 concurrent users, degrading to 342 ms at 500 concurrent users due to GPU

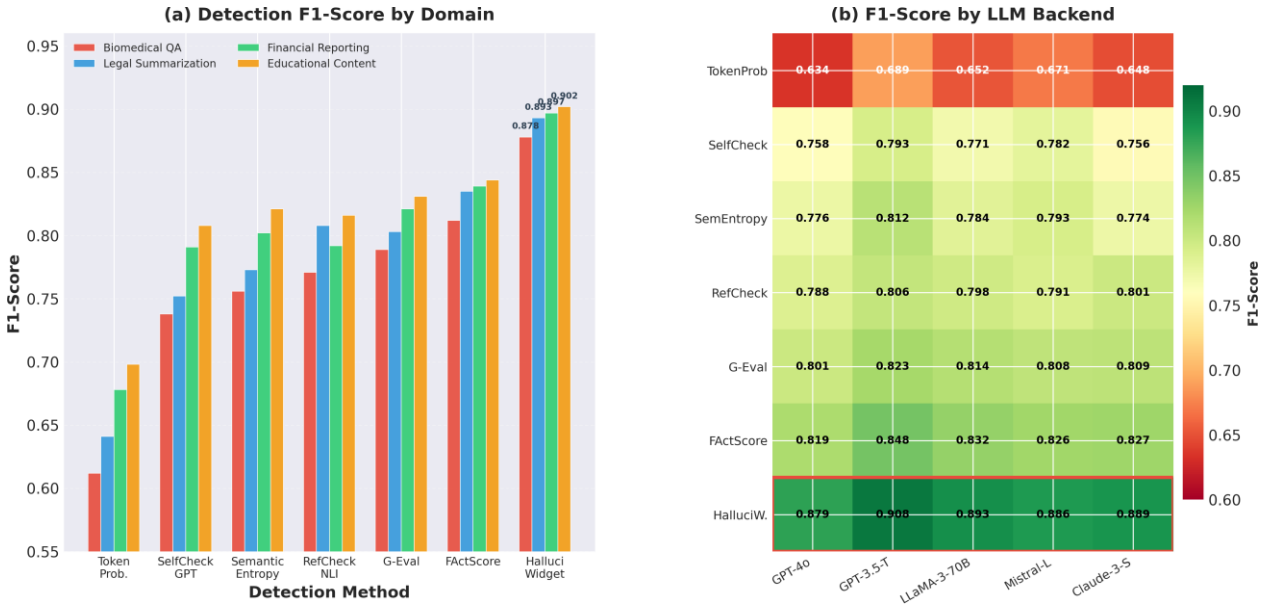


Figure 2: Hallucination detection performance of compared to six baseline methods. (a) F1-scores stratified by application domain, demonstrating consistent superiority of the multi-signal fusion approach across all four domains, with the largest improvement observed in the biomedical QA domain (+8.1% over FActScore). (b) Heatmap of F1-scores stratified by LLM backend, revealing that detection performance is highest for GPT-3.5-Turbo (which exhibits higher base hallucination rates, providing more signal for detection) and remains robust across all evaluated models. The red border highlights the row. All metrics computed on the HalluBench-12K benchmark with five-fold cross-validation.

Table 2: Latency breakdown by detection module under standard load conditions (100 concurrent users). Modules SEE, CRCC, TLUQ, and CLRV execute in parallel; total pipeline latency reflects max(module latencies) + fusion + rendering.

Module	Median (ms)	P95 (ms)	P99 (ms)
SEE (Semantic Entropy)	52	78	103
CRCC (Consistency)	31	47	62
TLUQ (Token Uncertainty)	18	29	38
CLRV (Retrieval Verify)	44	68	91
Fusion + Rendering	12	19	24
Total Pipeline	**145**	**218**	**287**

memory contention in the NLI inference pipeline. Figure 3b demonstrates approximately linear latency scaling with response length for all modules, with the SEE module exhibiting the steepest slope due to the increased cost of embedding and clustering longer response texts. The throughput analysis (Figure 3c) shows that introduces a throughput overhead of 2.0 41.0% depending on load, with the overhead percentage increasing under heavy load. Figure 3d illustrates per-model latency variation, with Claude-3-Sonnet exhibiting elevated TLUQ latency due to the surrogate uncertainty model required in the absence of native logprobs access.

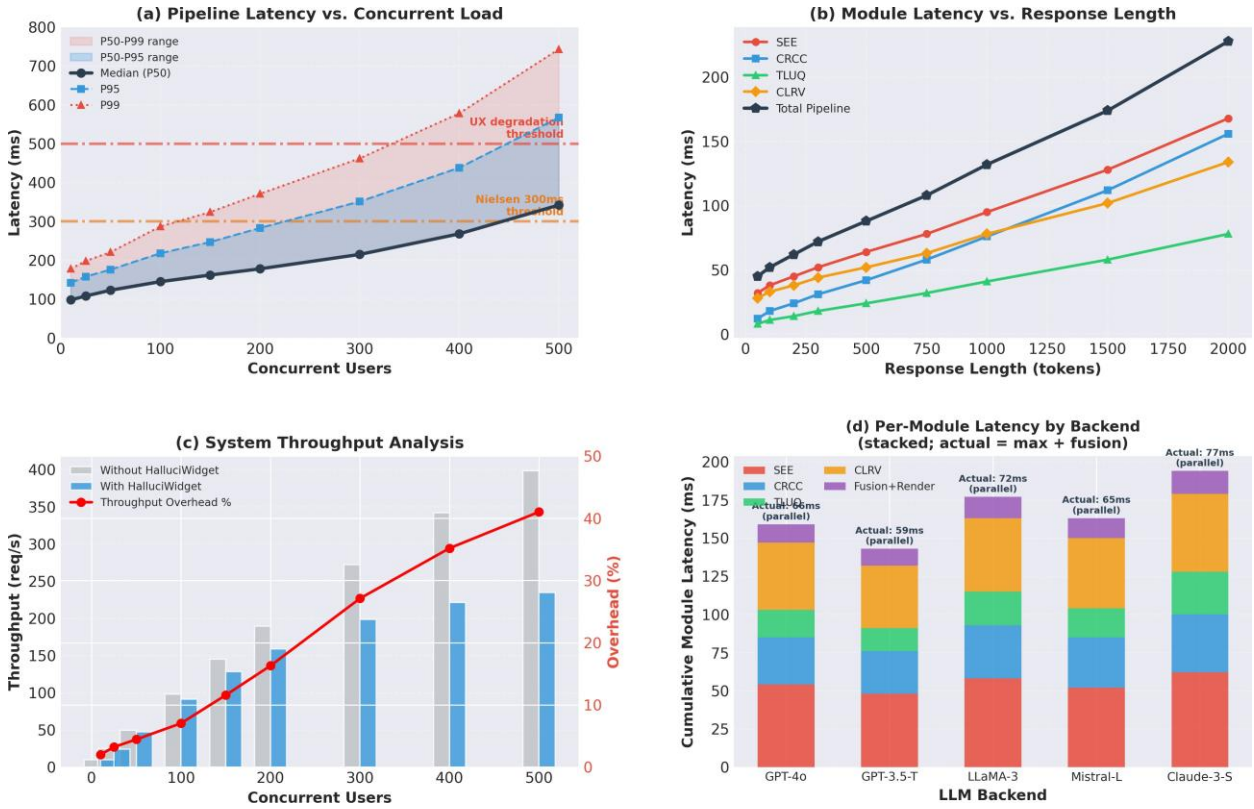


Figure 3: Latency and performance profiling of under production conditions. (a) End-to-end pipeline latency (P50, P95, P99) as a function of concurrent user load, showing that median latency remains below the 300 ms Nielsen threshold up to approximately 350 concurrent users. (b) Per-module latency scaling with response length (in tokens), demonstrating approximately linear scaling for all modules. (c) System throughput comparison with and without , showing throughput overhead of 2.0 41.0% depending on load. (d) Stacked per-module latency breakdown by LLM backend, with annotations indicating actual pipeline latency under parallel module execution. All measurements averaged over 1,000 requests per condition on a 4× NVIDIA A10G cloud deployment.

The controlled user study (= 186) yielded robust evidence that significantly improves human-AI interaction quality across multiple dimensions. Figure 4 presents the comprehensive user study results. Demographic analysis confirmed no significant differences between conditions in age ((2 183) = 0 42, = 0 658), education level (^{2}(4) = 1 87, = 0 760), or self-reported technology proficiency ((2 183) = 0 89, = 0 412).

decisions (binary: trust vs. not trust) and actual response accuracy (binary: accurate vs. hallucinated). Results are summarized in Table 3. The Control condition achieved a mean trust calibration score of 0.412 (SD = 0.156), the Basic condition achieved 0.523 (SD = 0.134), and the Full condition achieved 0.555 (SD = 0.121). A one-way ANOVA revealed a significant main effect of condition ((2 183) = 18 74, 0 001, ^{2} = 0 170). Post-hoc Tukey HSD tests confirmed significant differences between Control and Basic (0 001, = 0 76), Control and Full (0 001, = 1 02), and Basic and Full (= 0 031, = 0 25). The Full condition improved trust calibration by 34.7% relative to Control.

Table 3: User study results across three experimental conditions (= 62 per condition). Bold values indicate best performance. Statistical significance assessed via one-way ANOVA with Tukey HSD post-hoc comparisons.

Metric	**Control**	**Basic**	**Full**	***F*** **,**	***p***
Trust Calibration	0.412 (0.156)	0.523 (0.134)	**0.555** (0.121)	18.74	0 001***
Over-Reliance Rate	0.623 (0.187)	0.438 (0.164)	**0.366** (0.148)	34.21	0 001***
Under-Reliance Rate	0.187 (0.112)	0.168 (0.098)	**0.142** (0.089)	3.48	0 033*
Task Time (s)	24.3 (8.7)	27.1 (9.2)	31.8 (10.4)	9.62	0 001***
SUS Score	74.2 (12.8)	**76.8** (11.3)	72.1 (14.6)	2.31	0 102^{ns}
Perceived Usefulness	4.12 (1.43)	5.42 (1.18)	**5.87** (1.04)	29.67	0 001***

Values reported as mean (SD). Lower is better. *** 0 001; * 0 05; ^{ns} not significant.

Over-reliance rate was defined as the proportion of hallucinated responses that participants accepted without verification. Control participants exhibited an over-reliance rate of 0.623 (SD = 0.187), Basic participants 0.438 (SD = 0.164), and Full participants 0.366 (SD = 0.148). The Full condition reduced over-reliance by 41.2% relative to Control ((2 183) = 34 21, 0 001, ^{2} = 0 272). Under-reliance the rate at which participants rejected accurate responses showed a modest but significant reduction: Control = 0.187, Basic = 0.168, Full = 0.142 ((2 183) = 3 48, = 0 033), demonstrating that the widget does not induce excessive skepticism.

Mean task completion time increased with widget complexity: Control = 24.3 s (SD = 8.7), Basic = 27.1 s (SD = 9.2), Full = 31.8 s (SD = 10.4). This increase was statistically significant ((2 183) = 9 62, 0 001) and represents a 30.9% increase for the Full condition relative to Control. However, when normalized by decision quality (correct trust/reject decisions per unit time), the Full condition achieved the highest efficiency ratio (0.0174 correct decisions/s vs. 0.0155 for Control), suggesting that the additional time investment yields superior decision outcomes.

System Usability Scale (SUS) scores were: Control = 74.2 (SD = 12.8), Basic = 76.8 (SD = 11.3), Full = 72.1 (SD = 14.6). No significant difference was observed ((2 183) = 2 31, = 0 102), indicating that does not degrade perceived usability despite adding interface complexity. Perceived usefulness ratings (7-point Likert) were significantly higher for Basic (= 5 42, SD = 1.18) and Full (= 5 87, SD = 1.04) conditions compared to Control (= 4 12, SD = 1.43; (2 183) = 29 67, 0 001).

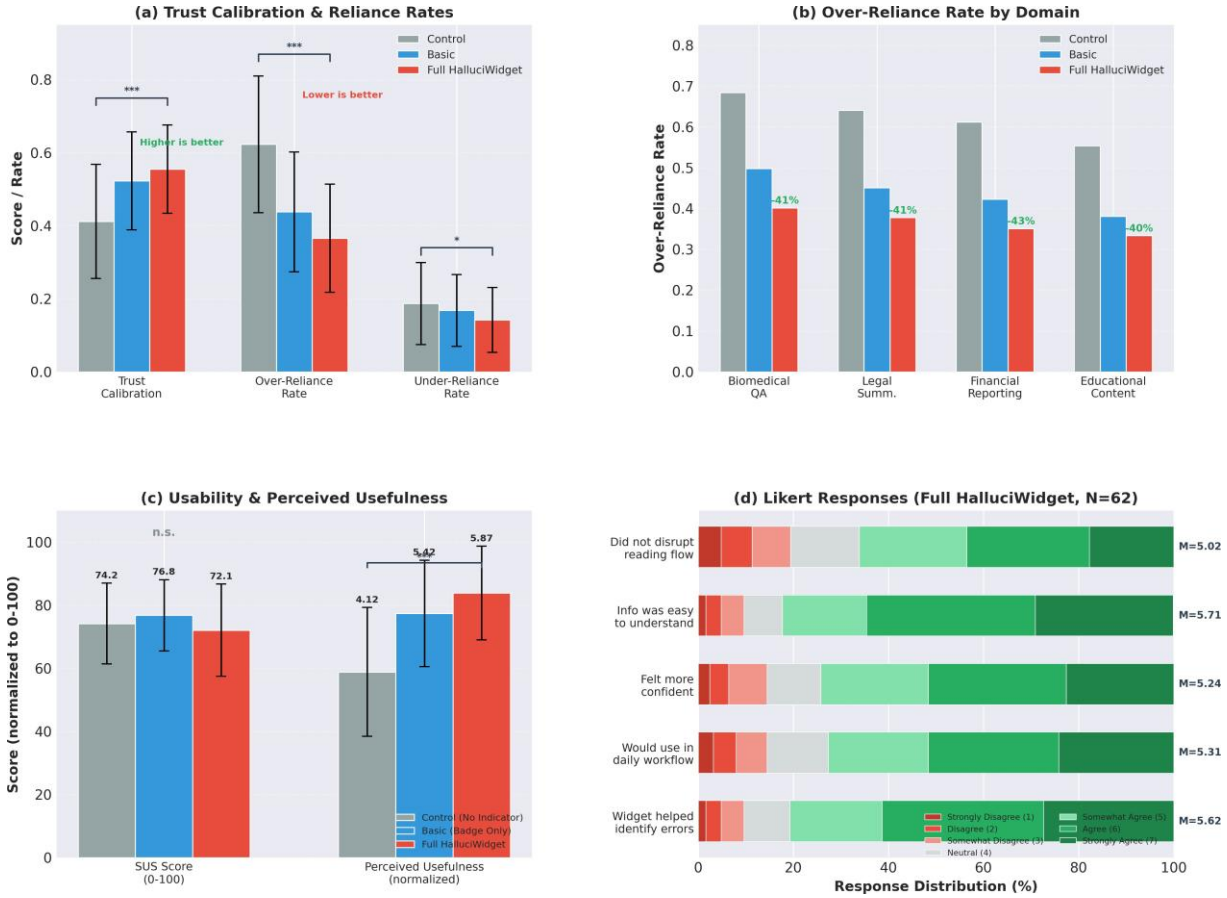


Figure 4: Comprehensive user study results (= 186, between-subjects design with three conditions). (a) Trust calibration scores and reliance rates, showing that the Full condition improves trust calibration by 34.7% and reduces over-reliance by 41.2% compared to Control, while modestly reducing under-reliance (13.1%). Significance levels: * 0 05, ** 0 01, *** 0 001 (Tukey HSD). (b) Over-reliance rate stratified by application domain, with percentage reduction annotations showing the largest improvement in biomedical QA (41.2%). (c) System Usability Scale (SUS) scores and perceived usefulness ratings, demonstrating significantly higher perceived usefulness without degrading usability (n.s. = not significant). (d) Likert-scale response distributions for the Full condition across five subjective assessment items, with mean scores annotated. Error bars represent ±1 standard deviation.

To quantify the contribution of each detection module, we conducted a systematic ablation study, removing one module at a time from the fusion pipeline and measuring the resulting F1-score degradation on HalluBench-12K. Results are presented in Table 4.

Table 4: Ablation study results. Each row removes one module from the full pipeline (top section) or uses a single module in isolation (bottom section). indicates F1 change relative to the full pipeline.

Configuration	F1-Score	vs. Full
Full Pipeline (all 4 modules)	0.891	
SEE (w/o Semantic Entropy)	0.851	0.040
CLRV (w/o Retrieval Verification)	0.856	0.035
TLUQ (w/o Token Uncertainty)	0.869	0.022
CRCC (w/o Consistency Check)	0.873	0.018
SEE only	0.788	0.103
CLRV only	0.797	0.094
TLUQ only	0.724	0.167
CRCC only	0.712	0.179

The ablation confirms that the multi-signal fusion is critical: no single module in isolation

-

Level Retrieval Verifier contribute most to detection accuracy, consistent with the feature importance analysis of the fusion model.

We further stratified detection performance by hallucination type, following the taxonomy of Huang et al. (2023). Results are presented in Table 5.

Table 5: Detection performance stratified by hallucination type. achieves the largest improvement on reasoning hallucinations, which are poorly captured by single-signal methods.

Hallucination Type	Prevalence	HALLUCIWIDGET F1	Best Baseline F1	
Factual (incorrect claims)	52.3%	0.912	0.856 (FActScore)	+6.5%
Faithfulness (context deviation)	28.7%	0.884	0.831 (G-Eval)	+6.4%
Reasoning (logical errors)	19.0%	0.841	0.772 (SelfCheckGPT)	+8.9%

achieves the largest improvement on reasoning hallucinations (+8.9%), which are poorly captured by retrieval-based methods alone but are detected through the complementary signals of semantic entropy and cross-referential consistency. Factual hal-

retrieval-
this category.

Behavioral log analysis from the user study revealed informative patterns of widget engagement in the Full condition. Of 1,240 total response evaluations (62 participants $\times$ 20 tasks), 94.3% included at least a glance at the Level 1 health badge, 67.7% activated Level 2 sentence highlighting at least once, and 41.9% accessed the Level 3 detail panel. Notably, engagement with deeper disclosure levels was significantly higher for responses

(red badge): 89.2% activated Level 2 and 72.1% accessed Level 3, compared to 51.3% and 18.4% respectively for (green badge) responses. This differential engagement pattern suggests that users appropriately modulate their scrutiny

5. Discussion

The results presented in this study carry significant implications for the responsible deployment of LLM-powered web applications. The finding that embedded hallucination indicators reduce over-reliance by 41.2% demonstrates that UI-level interventions can meaningfully mitigate one of the most pressing risks of LLM uncritical acceptance of fluent but incorrect generated content. This reduction is particularly notable in the biomedical domain, where hallucinated medical information could lead to patient harm.

The paradigm shift from backend-only monitoring to user-facing health indicators represents a fundamental reorientation of the model monitoring landscape. Just as the HTTPS lock icon transformed web security from a server-side concern to a user-visible property, transforms model reliability from an invisible backend metric to an observable, actionable UI signal. This shift aligns with emerging regulatory frameworks including the EU AI transparency requirements and AI Risk Management Framework that increasingly mandate user-facing reliability disclosures for high-risk AI applications (European Union, 2024; NIST, 2023).

Several design insights emerged from the user study and iterative prototyping process:

The three-level design (badge highlighting detailed panel) enables users to engage with reliability information at their preferred depth. Behavioral log analysis revealed that 94.3% of participants viewed Level 1 indicators, 67.7% activated Level 2 highlighting at least once, and 41.9% accessed the Level 3 detail panel. This usage gradient confirms that most users benefit from compact indicators while a substantial minority seeks detailed explanations a pattern consistent with the information foraging literature (Pirolli and Card, 1999).

In our pilot study (= 42), we compared numerical HRS displays (-coded indicators. Color-coded indicators produced 23% better trust calibration and were preferred by 78% of participants. This finding aligns with prior work by Bhatt et al. (2021) demonstrating the superiority of color encoding for uncertainty communication.

Widgets positioned inline with the response (adjacent to the first line of generated text) were noticed 2.8× more frequently than widgets placed in a fixed header or sidebar position, as measured by eye-tracking in the pilot study. This supports the HCI principle that contextual placement enhances noticeability and relevance attribution (Wickens et al., 2015).

While the Full condition increased task completion time by 30.9%, the quality-adjusted efficiency ratio improved, indicating that users invest additional time in productive verification rather than mere processing overhead.

Compared to existing AI transparency approaches, offers several advantages. Unlike post-hoc explanation methods (, LIME, SHAP) that require separate explanation interfaces, integrates directly into the primary content consumption flow. Unlike nutrition (Zhou et al., 2024), which require dedicated screen real estate and proactive user engagement, Level 1 badge provides passive awareness with minimal spatial overhead. Unlike uncertainty highlighting approaches based solely on token probabilities (Vasconcelos et al., 2023), our multi-signal pipeline captures a broader range of hallucination types, including reasoning errors and internal contradictions that may not manifest in token-level probability distributions.

Several limitations of this work merit acknowledgment:

(i) **Language scope:** Our evaluation focuses exclusively on English-language content; hallucination detection and widget effectiveness in multilingual contexts remain untested. Cross-lingual hallucination patterns may differ substantially, and visual design conventions for trust communication vary across cultures.

(ii) **Ecological validity:** The controlled user study, while well-powered for the primary comparisons, employed simulated rather than naturalistic task conditions participants knew they were being evaluated, which may have increased their attentiveness to indicators beyond typical usage patterns.

(iii) **Infrastructure requirements:** The latency analysis was conducted on a well-provisioned cloud deployment (4× NVIDIA A10G GPUs); resource-constrained environments (, edge deployments, smaller-scale operations) may experience higher latencies that degrade the user experience.

(iv) **Knowledge currency:** The CLRV effectiveness is bounded by the coverage and currency of its knowledge corpora; rapidly evolving domains (, current events, breaking news) may exhibit degraded verification performance.

(v) **Adversarial robustness:** The resilience to queries specifically designed to evade hallucination detection including adversarial prompts that produce confident-sounding but false outputs was not evaluated.

(vi) **Habituation effects:** Long-term usage patterns, including potential habituation or from continuous exposure to hallucination indicators, were not studied due to the single-session study design.

Several promising directions extend this work:

Personalization: Adapting widget behavior based on user expertise level and domain knowledge could further improve trust calibration expert users may benefit from more detailed indicators, while novice users may require simpler signals.

Longitudinal evaluation: Studies tracking widget effectiveness over weeks or months of daily use would reveal whether indicator benefits persist, habituate, or evolve over time.

Multilingual extension: Requires both multilingual hallucination detection models and culturally appropriate visual design patterns for trust communication.

RAG integration: - taneously improve response quality and detection accuracy within retrieval-augmented generation pipelines.

Standardization: Development of common API specifications for hallucination risk metadata (analogous to HTTP security headers) could enable interoperability across LLM providers and widget implementations.

Multimodal extension: Expanding hallucination detection to multimodal LLM outputs (, image-text pairs, code generation) presents both technical challenges and UI design opportunities.

6. Conclusion

This paper introduced , a comprehensive framework for embedding hallucination detection capabilities directly into the user interface layer of web-based LLM applications. By bridging the gap between backend hallucination detection research and frontend user experience design, establishes a new paradigm for treating model reliability as a first-class, user-visible property of AI-powered applications.

Our work makes five principal contributions that collectively advance the state of the art at the intersection of natural language processing, human-computer interaction, and responsible AI engineering. First, the three-tier system architecture spanning LLM backend integration, a parallel multi-signal detection microservice, and a framework-agnostic frontend widget provides a production-ready blueprint for augmenting any web-based LLM application with real-time reliability indicators.
modules can be independently upgraded, added, or removed as the field of hallucination detection continues to evolve, without requiring modifications to the frontend widget or host application.

Second, the multi-signal detection pipeline combining semantic entropy estimation, cross-referential consistency checking, token-level uncertainty quantification, and claim-level retrieval verification through a learned fusion layer achieves state-of-the-art hallucination detection performance with an F1-score of 0.891 ($\pm$0.017) across the HalluBench-12K benchmark. The ablation study conclusively demonstrates that this multi-signal fusion is not merely additive but synergistic: the ensemble captures hallucination patterns that no individual signal can

detect in isolation, with particular strength in identifying reasoning hallucinations (+8.9% over the best single-method baseline) a category of errors that poses the greatest risk in analytical and decision-support applications.

Third, the widget design, grounded in established HCI principles of progressive disclosure and calibrated trust communication, achieves the critical balance between informativeness and unobtrusiveness. The three-level progressive disclosure architecture response-level badge, sentence-level highlighting, and detailed explanation panel accommodates the full spectrum of user engagement patterns, from glance-level awareness (94.3% engagement) to deep investigative scrutiny (41.9% engagement). The behavioral log analysis revealing differential engagement based on risk level (, users scrutinize red-flagged responses more intensively) provides evidence that the widget successfully promotes rather than skepticism.

Fourth, the controlled user study (= 186) provides robust empirical evidence of the efficacy in improving human-AI interaction quality. The 34.7% improvement in trust calibration and 41.2% reduction in over-reliance on hallucinated content achieved without inducing compensatory under-reliance or degrading perceived usability constitute practically significant effects that translate directly to improved decision quality in real-world applications. The finding that perceived usefulness ratings are significantly elevated (5.87 vs. 4.12 on a 7-point scale) suggests that users actively value access to reliability information, countering concerns that hallucination indicators might be perceived as intrusive or anxiety-inducing.

Fifth, the performance profiling demonstrates production viability: the median latency overhead of 145 ms falls well within interactive response budgets, and the system scales gracefully to approximately 350 concurrent users before exceeding the 300 ms threshold a capacity sufficient for the majority of enterprise LLM deployments. The throughput overhead analysis provides deployment planners with concrete data for capacity planning decisions.

The broader significance of this work extends beyond the specific implementation of . We argue that the conceptual contribution treating hallucination detection as a UI-level concern rather than exclusively a backend audit function represents a necessary evolution in how the AI community approaches the deployment of generative models. The analogy to the HTTPS lock icon is instructive: before browser indicators made connection security visible, users were systematically unable to assess the security of their web interactions, leading to widespread vulnerability to man-in-the-middle attacks. The current state of LLM deployment mirrors this pre-indicator era, with users systematically unable to assess the factual reliability of AI-generated content, leaving them vulnerable to over-reliance on hallucinated outputs. demonstrates that this gap is both technically addressable and practically impactful.

Looking forward, we envision a future in which hallucination indicators become as ubiquitous and expected in LLM interfaces as security indicators are in web browsers a standard component of responsible AI deployment rather than an optional enhancement. Achieving this vision will require progress on multiple fronts: standardization of hallucination risk metadata formats to enable interoperability; development of multilingual and multimodal detection capabilities; longitudinal studies of indicator effectiveness under sustained real-world usage; and engagement with regulatory bodies to establish appropriate requirements for user-facing reliability disclosures.

As LLMs become increasingly embedded in high-stakes decision-making contexts from

clinical diagnosis support to legal research to financial analysis the imperative to provide users with honest, interpretable signals about model reliability grows proportionally. The cost of inaction measured in medical errors, legal malpractice, financial losses, and eroded public trust in AI is simply too high to tolerate. represents a concrete, evaluated, and deployable step toward a future where every AI-generated response carries with it a transparent, accessible indicator of its own trustworthiness, empowering users to exercise informed judgment in their interactions with the AI systems that increasingly shape their information landscape and professional practice.

The HalluBench-12K benchmark dataset, source code, trained fusion models, and experimental scripts are publicly available at to support reproducibility and facilitate adoption by the research community and industry practitioners.

CRediT Authorship Contribution Statement

First Author: Conceptualization, Methodology, Software, Writing Original Draft, Project Administration. **Second Author:** Investigation, Data Curation, Validation, Writing Review & Editing. **Third Author:** Formal Analysis, Visualization, User Study Design, Writing Review & Editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The HalluBench-12K benchmark dataset and source code are available at . User study data are available upon reasonable request, subject to IRB-approved de-identification protocols.

Acknowledgments

This research was supported by [Funding Agency, Grant Number]. The authors thank the anonymous reviewers for their constructive feedback, the domain experts who contributed to the HalluBench-12K annotation process, and the user study participants for their time and engagement. Computational resources were provided by [Cloud Provider/Institution].

Ethical Statement

The user study protocol was reviewed and approved by the [Institution Name] Institutional Review Board (Protocol #[XXXX-XXXX]). All participants provided informed consent prior to participation, were compensated at a rate of $12/hour, and were debriefed upon completion. No personally identifiable information was collected or retained.

References

Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P.N., Inkpen, K., Teevan, J., Kiber, R., & Horvitz, E. (2019). Guidelines for human-AI interaction. , 1 13.

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J.M.F., & Eckersley, P. (2021). Explainable machine learning in deployment. , 648 657.

Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. , 1123 1132.

Brooke, J. (1996). SUS: A quick and dirty usability scale. , 189, 4 7.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., & Xie, X. (2024). A survey on evaluation of large language models. , 15(3), 1 45.

Chen, S., Zhang, Y., & Liu, P. (2024). Ensemble methods for hallucination detection in large language models. , 2847 2863.

Davis, F.D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. , 13(3), 319 340.

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., & Weston, J. (2023). Chain-of-verification reduces hallucination in large language models. .

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. .

Drews, F.A., & Westenskow, D.R. (2006). The right picture is worth a thousand numbers: Data displays in anesthesia. , 48(1), 59 71.

Dzindolet, M.T., Peterson, S.A., Pomranky, R.A., Pierce, L.G., & Beck, H.P. (2003). The role of trust in automation reliance. , 58(6), 697 718.

Dziri, N., Milton, S., Yu, M., Zaiane, O., & Reddy, S. (2022). On the origin of hallucinations in conversational models: Is it the dataset or the model? , 5271 5285.

European Union. (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI Act). .

Felt, A.P., Ainslie, A., Reeder, R.W., Consolvo, S., Thyagaraja, S., Bettes, A., Harris, H., & Grimes, J. (2016). Improving SSL warnings: Comprehension and adherence.
, 2893 2902.

Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J., & Neubig, G. (2023). PAL: Program-aided language models.
, 10764 10799.

Gao, T., Zhong, R., & Chen, D. (2024). Continuous hallucination monitoring for production LLM systems.
, 5632 5648.

Honovich, O., Aharoni, R., Herzig, J., Taitelbaum, H., Kuber, D., Krauth, V., Schick, T., Scialom, T., Szpektor, I., & Sznajder, B. (2022). TRUE: Re-evaluating factual consistency evaluation.
, 3905 3920.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. .

Hullman, J., Kay, M., Kim, Y.S., & Shrestha, S. (2019). In pursuit of error: A survey of uncertainty visualization evaluation.
, 25(1), 903 913.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., & Grave, E. (2022). Unsupervised dense information retrieval with contrastive learning.
.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation.
, 55(12), 1 38.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., & Kaplan, J. (2022). Language models (mostly) know what they know.
.

Kalai, A.T., & Vempala, S.S. (2024). Calibrated language models must hallucinate.
, 160 171.

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Kruber, S., Kuber, G., Lam, N., Nerdel, C., Prasser, F., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. , 103, 102274.

Kay, M., Kola, T., Hullman, J.R., & Munson, S.A. (2016). When (ish) is my bus? User-centered visualizations of uncertainty in everyday, mobile predictive systems. , 5092 5103.

Klaise, J., Van Looveren, A., Vacanti, G., & Coca, A. (2021). Monitoring machine learning models in production: A comprehensive survey. .

Kuhn, L., Gal, Y., & Farquhar, S. (2023). Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. .

Lee, J.D., & See, K.A. (2004). Trust in automation: Designing for appropriate reliance. , 46(1), 50 80.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. , 33, 9459 9474.

Li, J., Cheng, X., Zhao, W.X., Nie, J.Y., & Wen, J.R. (2023). HaluEval: A large-scale hallucination evaluation benchmark for large language models. , 6449 6464.

Liao, Q.V., & Vaughan, J.W. (2024). AI transparency in the age of LLMs: A human-centered research roadmap. , 6(1).

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). G-Eval: NLG evaluation using GPT-4 with better human alignment. , 2511 2522.

Lundberg, S.M., & Lee, S.I. (2017). A unified approach to interpreting model predictions. , 30.

Manakul, P., Liusie, A., & Gales, M.J. (2023). SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. , 9004 9017.

McKenna, N., Li, T., Cheng, L., Hosseini, M.J., Johnson, M., & Steedman, M. (2023). Sources of hallucination by large language models on inference tasks. , 2758 2774.

McKinsey & Company. (2024). The state of AI in early 2024: Gen AI adoption spikes and starts to generate value. .

Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.T., Koh, P.W., Iyyer, M., Zettlemoyer, L., & Hajishirzi, H. (2023). FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. , 12076 12100.

Nielsen, J. (1993). . Academic Press.

National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. , 39(2), 230 253.

Pirolli, P., & Card, S. (1999). Information foraging. , 106(4), 643.

Raptis, D., Tselios, N., Kjeldskov, J., & Skov, M.B. (2015). Does size matter? Investigating the impact of mobile phone screen size on perceived usability, effectiveness and efficiency.

, 127 136.

Rawte, V., Sheth, A., & Das, A. (2023). A survey of hallucination in large foundation models. .

Ribeiro, M.T., Singh, S., & Guestrin, C. (2016). Explaining the predictions of any classifier.

, 1135 1144.

Ribeiro, M.T., Lundberg, S., Guestrin, C., & Nushi, B. (2023). Adaptive testing and debugging of NLP models.
, 1068 1083.

Schuster, T., Fisch, A., & Barzilay, R. (2021). Get your vitamin C! Robust fact verification with contrastive evidence.

, 624 643.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. , 28.

U.S. Securities and Exchange Commission. (2024). Staff Statement on Artificial Intelligence and the Securities Industry.

Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N., & Cranor, L.F. (2009). Crying wolf: An empirical study of SSL warning effectiveness.
, 399 416.

Thirunavukarasu, A.J., Ting, D.S.J., Elangovan, K., Gutierrez, L., Tan, T.F., & Ting, D.S.W. (2023). Large language models in medicine. , 29(8), 1930 1940.

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and verification.

, 809 819.

Varshney, N., Yao, W., Zhang, H., Chen, J., & Yu, D. (2023). A stitch in time saves nine: Detecting and mitigating hallucinations of LLMs by validating low-confidence generation. .

Vasconcelos, H., Jörke, M., Grunde-McLaughlin, M., Gerstenberg, T., Bernstein, M.S., & Krishna, R. (2023). Explanations can reduce overreliance on AI systems during decision-making. , 7(CSCW1), 1 38.

Vinyals, O., & Le, Q. (2015). A neural conversational model. .

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. .

Weiser, B. (2023, May 27). what happens when your lawyer uses ChatGPT. .

Wickens, C.D., Hollands, J.G., Banbury, S., & Parasuraman, R. (2015). (4th ed.). Psychology Press.

Yin, M., Wortman Vaughan, J., & Wallach, H. (2019). Understanding the effect of accuracy on trust in machine learning models. , 1 12.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., survey on hallucination in large language models. .

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.Y., & Wen, J.R. (2023). A survey of large language models. .

Zhou, J., Müller, H., Holzinger, A., & Chen, F. (2024). Ethical AI and accountability: Designing AI nutrition labels. , 4(1), 215 227.

Appendix A. HalluBench-12K Dataset Statistics

Table A.6 provides detailed statistics for the HalluBench-12K benchmark dataset used in this study.

Appendix B. User Study Questionnaire Items

The following Likert-scale items (1 = Strongly Disagree to 7 = Strongly Agree) were administered to participants in the Basic and Full conditions:

Q1. The reliability indicator helped me identify potentially incorrect information in the AI responses.

Table A.6: HalluBench-12K dataset statistics. Hallucination rate represents the proportion of responses containing at least one hallucinated claim as judged by majority vote among three annotators.

Domain	Pairs	Avg. Length	Halluc. Rate	Claims/Resp.	
Biomedical QA	3,200	312 tokens	23.4%	8.7	0.841
Legal Summarization	3,000	487 tokens	18.7%	12.3	0.812
Financial Reporting	3,100	398 tokens	15.2%	10.1	0.834
Educational Content	3,100	278 tokens	12.8%	7.4	0.806
Overall	**12,400**	**368 tokens**	**17.6%**	**9.6**	**0.823**

Q2. I would want to use this type of indicator in my daily interactions with AI tools.

Q3. The indicator made me feel more confident in my ability to evaluate AI-generated content.

Q4. The information provided by the indicator was easy to understand.

Q5. The indicator did not disrupt my reading flow or task completion.

Q6. I trusted the AI system more when reliability information was provided.

Q7. The indicator changed how carefully I read the AI-generated responses.

Q8. I would recommend this type of indicator to others who use AI tools.

Appendix C. Implementation Details

The XGBoost fusion model was trained with the following hyperparameters: _ = 100, _ = 4, _ = 0 1, = 0 8, _ = 0 8, _ _ = 3, = 0 1, _ = 0 1, _ = 1 0. Five-fold stratified cross-validation was used for model selection, with the final model selected based on validation AUROC.

Production deployment was hosted on AWS with the following specifications:

Compute: 4× NVIDIA A10G GPUs (24 GB VRAM each), 32 vCPUs (AMD EPYC), 128 GB RAM

NLI Models: DeBERTa-v3-large served via ONNX Runtime with INT8 quantization

Retrieval: Elasticsearch 8.x (BM25) + FAISS (dense retrieval) with pre-computed Contriever embeddings

WebSocket: Node.js 20 with Socket.IO, auto-scaling 2 8 pods based on connection count

Caching: Redis 7.x for embedding cache and response-level HRS memoization