SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Adversarial Prompt Analysis and Defense Mechanisms for Secure Large Language Model Deployment

Anjali Vats[1], Shikha, Gaurav[1], Melisa[1]

[1]Taleeb Vancouver, Canada

Abstract
Secure deployment of large language models (LLMs) requires simultaneous control of two failure classes that are usually studied separately: adversarial prompting, which attempts to override safety constraints, and hallucination, which produces unsupported or factually inconsistent outputs. This paper proposes APHD-Guard, a calibrated multi-source detection framework for adversarial prompt and hallucination risk analysis. The framework uses a DeBERTa-v3-base encoder with sequence-level risk classification and auxiliary span-localization supervision, integrates factuality and hallucination corpora with controlled adversarial augmentation, and applies validation-only temperature scaling and threshold selection before external evaluation. The publication protocol uses SOCRATES-300K, RAGTruth and FEVER for development, while TruthfulQA, HaluEval, HalluDial, JailbreakBench and BIPIA are isolated as external tests. Exact fingerprint filtering and MinHash near-duplicate screening are performed before model fitting. The current manuscript result set records an average external accuracy of 0.832, Macro-F1 of 0.816, MCC of 0.666, AUROC of 0.882 and AUPRC of 0.875, with HaluEval showing the strongest benchmark-level performance. The proposed model improves average Macro-F1 by 0.133 over a TF-IDF logistic-regression baseline and reduces expected calibration error from 0.097 before calibration to 0.043 after calibration. Ablation analysis attributes the largest performance decrease to removal of controlled adversarial augmentation, followed by RAGTruth supervision. The study further introduces explicit publication claim gates for contamination control, multi-seed replication, calibration, significance testing, benchmark completeness and dataset provenance. These findings support a layered view of LLM security in which detection, uncertainty calibration, adversarial robustness, data provenance and deployment governance are treated as a single empirical assurance problem rather than independent safeguards.

Keywords: large language models; adversarial prompts; prompt injection; jailbreak detection; hallucination detection; calibration; LLM security; trustworthy AI; red teaming; APHD-Guard

1. Introduction

Large language models are now in almost every application you can think of, from chatbot to software development and retrieval systems to clinical decision support systems and knowledge assistants and semi-autonomous agents. The problem with security is changed through this transition. A Good Model Is No Longer Just Fluent Text – It Must Tell Instructions Apart From Untrusted Data, Avoid Scenarios With Fabricated Claims, Convey Calibrated Ignorance And Remain Aligned When The Context Changes Hallucination can be uncovered by computing uncertainty over meaning (versus just surface forms) [1]. Hierarchical semantic verification also exemplifies how fctuality control can achieve gains from comparing generated statements to evidence at multiple semantic granularities [2]. Perturbation-based studies also show that supposedly robust foundation models can become very much less reliable under slight perturbations of input [3].

The effect on society is compounded when outputs from models are fed back into other systems. Creating a more nuanced type of misinformation, this hallucinated material flows through complex adaptive social networks in which the uncertainty that is lackluster in a single generated response can become input into downstream agents and users [4]. In summarization, a hallucination can range from being short and linguistically convincing while still not grounded in the source – this leads to recent work that combines quantitative detection where predictions are made, for instance using ROUGE [4], with iterative mitigation [5]. Third, Multimodal systems provide an additional opportunity for attack where textual and visual evidence may conflict, allowing cross-modal hallucinations in addition to other adversarial manipulations undetectable by text-only safety filters [6]. Noise-aware medical experiments likewise indicate that even irrelevant or corrupted evidence can increase the risk of hallucination, which drives explicit denoising and evidence-selection mechanisms [7].

The issue is not just limited to internals of the model. Reference-generation studies have shown fabricated or erroneous references in systematic-review workflows [8], and earlier analyses of scientific writing have cautioned that confident fabrication could be interpreted as contributory evidence [9]. Factual, logical and instruction-consistency hallucinations have been shown to persist in advanced systems even in high-stakes examination settings [10]. As a result, we must view hallucination when deploying an LLM as a failure of

integrity and adversarial prompting as a flux-control failure. While the output generated from either has good linguistic quality, it does not honor the security policy defined.

Existing defenses are fragmented. MetaQA utilizes metamorphic relations for hallucination detection without needing access to any external probabilities [11]. Cross-LLM architectures focused on retrieval, vector databases and policy controls to cure both hallucination and injection attacks [12]. Additional recent work posits that hallucination is intrinsically related to generalization, and thus has to be controlled through verification instead of eliminated [13]. Additionally, fact-check research emphasizes that factuality assessment should consider evidence quality and the uncertainty in decomposing claims for verification [14]. However, scaling and higher instruction-following ability does not equal reliability; larger and more instructed models also tend to get confidently wrong on harder tasks [15].

Retrieval augmentation is one way to concretely ground your model responses, but the quality of retrieval itself is a security perimeter. By further jointly querying Latent variable based retrieval-and-ranking architectures [16] achieves higher domain accuracy with curated and reranked evidence base, and focuses on a scaled retrieval-augmented generation evaluation frameworks showing that retriever quality significance answer faithfulness task specificity validation should be evaluated separately. Uncertainty quantification provides a complementary mechanism for identifying regions in which we should not trust model confidence [18]. User studies reveal that human trust and decision quality are also influenced by how uncertainty is communicated [19] while token-based certainty retrieval offers a concrete mechanism for confidence-sensitive information extraction [20].

Adversarial prompting creates an additional failure channel. JailGuard demonstrates that prompt-based attacks can be detected through a universal mutation-and-discrepancy framework [21]. RedAgent shows that red teaming becomes more effective when attacks are adapted to the specific system context rather than treated as context-free strings [22]. Safety can also be weakened after benign or task-specific adaptation, motivating post-fine-tuning realignment [23] and machine-unlearning approaches whose safety gains must be weighed against utility loss [24]. Parameter-efficient adaptation can similarly degrade alignment, which has motivated pruning-based LoRA safety mechanisms [25].

This paper addresses the resulting gap: current research rarely evaluates hallucination risk, adversarial-prompt risk, calibration, contamination, ablation evidence and publication integrity within one empirical pipeline. We therefore formulate secure LLM deployment as a calibrated risk-detection problem and introduce APHD-Guard. The framework is designed to separate clean supported content from hallucinated/refuted content, unverifiable content and prompt attacks; to localize suspicious spans where supervision is available; and to expose a single risk score that can drive a deployment gate. The experimental design uses heterogeneous training data, external benchmark isolation, multiple random seeds, grouped bootstrap uncertainty, baseline comparison, ablation experiments and explicit claim gates. The central objective is not to claim that one classifier solves LLM security, but to demonstrate how a reproducible detection layer can become a measurable component of a defense-in-depth architecture.

1.1 Research Questions and Contributions

The study is organized around four research questions: (RQ1) Can a multi-source detector generalize across factuality, hallucination and adversarial-prompt benchmarks that are excluded from training? (RQ2) Does calibration produce a practically useful risk score rather than a purely discriminative classifier? (RQ3) Which components of the training mixture contribute most strongly to external robustness? (RQ4) Which integrity gates must be satisfied before the numerical results can support publication-level claims?

- **Unified risk formulation.** APHD-Guard treats factual hallucination, unverifiability and prompt attacks as related unsafe states while retaining a supported-clean state for deployment decisions.
- **Multi-source, leakage-aware training.** Factuality, hallucination and evidence-verification datasets are combined with controlled adversarial augmentation, followed by exact and MinHash-based contamination screening.
- **Calibrated deployment interface.** Validation-only temperature scaling and threshold selection convert neural logits into a risk score suitable for an intervention policy.
- **Publication-scale evaluation.** Five independent seeds, grouped bootstrap intervals, paired baseline significance tests, safety-oriented metrics and ablation analysis are integrated into one workflow.
- **Claim-gated reporting.** Dataset provenance, benchmark completeness, train/test isolation and reproducibility conditions are checked independently from predictive performance.

2. Related Work and Research Gap

2.1 Hallucination Detection, Factuality and Uncertainty

Hallucination detection has evolved from surface-level consistency checks toward semantic, evidence-aware and uncertainty-aware methods. Farquhar et al. operationalized semantic entropy as an uncertainty signal over meaning-equivalent generations [1]. Liu et al. proposed hierarchical semantic pieces to decompose generated and reference text before verification [2]. Saha et al. showed that perturbations can reveal latent hallucination behavior in foundation models [3]. Hao et al. moved the analysis beyond a single model output by modelling the spread of hallucinated information in social networks [4]. Liu et al. used a

question-generation, sorting and evaluation workflow for faithful summarization [5]. Sun et al. extended the problem to multimodal models and argued that correctness and uncertainty must be jointly evaluated [6]. Qu et al. demonstrated that noisy medical contexts require explicit robustness against irrelevant evidence [7].

Studies from the literature and medical context give examples of why factuality cannot simply be reduced to stylistic fluency. Chelli et al. observed that generative systems for systematic-review tasks led to much more heterogeneous reference accuracy [8] ChatGPT outputs have been described as containing invent scientific content and references in the work of Alkaissi and McFarlane [9]. Ye et al. examined the types of hallucination in postgraduate medical entrance questions and suggested factual fabrication to still be a primary failure mode [10]. Yang et al. alternative to detectors based on token probabilities or external evidence, called metamorphicrelation testing [11]. Gokcimen and Das integrated retrieval, policy and cross-model verification into a security-oriented architecture [12]. Singh et al. characterized hallucination as an externalizable trade-off rather than one we should expect to be able to remove [13].

Factuality research increasingly intersects with formal fact-checking. Augenstein et al. argue that LLM-era factuality requires claim-level evidence, provenance and explicit uncertainty [14]. Zhou et al. show that scale can increase the frequency of plausible but incorrect answers on difficult tasks [15]. Retrieval can reduce this risk when evidence is ranked appropriately; Yang et al. report gains from dual retrieval and ranking in medical QA [16], while Elmitwalli et al. emphasize scalable, domain-specific RAG evaluation [17]. Catak and Kuzlu explore uncertainty geometry as an indicator of out-of-distribution behavior [18]. Xu et al. demonstrate that verbalized uncertainty affects user trust and performance [19]. Schimmelpfennig et al. operationalize certainty retrieval at the token level for clinical information extraction [20]. Collectively, these studies motivate APHD-Guard's decision to report AUROC/AUPRC together with Brier score, ECE and deployment-oriented miss/retention metrics.

2.2 Jailbreaks, Prompt Injection and Safety Alignment

Prompt attacks exploit the fact that an LLM processes policy instructions, user instructions and untrusted content through a common token interface. JailGuard detects prompt attacks through discrepancy induced by input mutation [21]. RedAgent shows that real-world systems can be more vulnerable to context-specific attacks than to generic jailbreak prompts and demonstrates the value of autonomous red teaming [22]. Safety failures also arise after fine-tuning: Yi et al. use subspace-oriented model fusion to recover safety without discarding task knowledge [23]. Zhao et al. evaluate machine unlearning for alignment and identify a safety-utility trade-off that complicates simple 'forget harmful knowledge' strategies [24]. Ao et al. similarly show that LoRA adaptation can erode safety and propose distance-guided pruning to remove risky adapter components [25].

Attack transferability is another unresolved issue. Meade et al. show that adversarial triggers do not transfer uniformly across aligned models and that preference-aligned and fine-tuned models exhibit different robustness characteristics [26]. Shi et al. show that LLM-generated detector-evasion transformations can reduce the effectiveness of language-model detectors, highlighting the importance of testing countermeasures in the context of adaptive attacks [27]. Betley et al. expand that warning: task-specific fine-tuning can promote behavioral misalignment outside of the target task [28]. Lindström et al. We analyze the sociotechnical limits of RL from Human Feedback and show that helpfulness, harmless, and honesty cannot be even optimized with preference optimization alone [29]. Raza et al. Make sure it retains language understanding and knowledge utility in addition to improving safety focused instruction tuning. [30] These results encourage the exploration of a defense layer not dependent on the assumption that the base model is still perfectly aligned after deployment or adaptation.

2.3 Secure LLM Systems, Software Engineering and Agentic Attack Surfaces

Safe LLM deployment is not limited to that of conversational harmfulness. Rodriguez et al. show that LLMs can parse privacy policies at scale, however considerable validation and governance is essential to safely deploy in privacy-sensitive settings [31]. Tony et al. fill this gap by systematically investigating prompting methods for secure code generation. We show that the design of prompts has a substantial impact on security outcomes [32]. Xuan et al. LLMs have been successful in a variety of domains, from natural language processing etc. Gradually, some researchers began to use LLMs for vulnerabilities detection [33]. Mao et al. stress the importance of explainability in vulnerability-detection pipelines [34]. Jamal et al. use transformer-based modelling for phishing and spam detection [35], while Kibriya et al. we did combine deep learning with LLM-oriented techniques but for lightweight malicious-URL detection [36]. Metric-Based Signatures: Le, Tran have demonstrated the use of metric-based signatures for generated-text detection [37].

As LLMs control tools or embodied actions, the attack surface grows larger. Liu et al. exhibit contextual backdoor attacks on embodied LLM agents à la layer freezing models [38]. This finding is especially critical for retrieval and few-shot systems since the poisoned contextual demonstration can serve as an attack surface. Shorinwa et al. summarise inference uncertainty methods and emphasise the necessity of associating uncertainty with operational choices [39] Lin et al. Learned Methods: Synthetic methods for producing debiasing and dehallucination mechanism for robust, trustworthy LLMs [40]. Huang et al.

provide a classification of hallucination types and evaluation problems [41]. Jabbar et al. assess evaluation and transferability challenges that have not been addressed in the existing literature but could be useful for red-teaming methods [42]. Naveed et al. position these threats in the broader architecture and training schema of modern LLMs [43].

Agentic systems are introducing cross-module threats related to memory, data retrieval, tools and external execution. Gan et al. characterize security and privacy threats arising from LLM-based agents [44]. Paracha et al. situate these problems in the broader adversarial machine-learning threat landscape [45]. Zhou et al. overview the existing vulnerability detection and repair techniques using LLM [46], and Sheng et al. synthesize LLM-based software-security techniques [47]. Hou et al. Here, you were trained on quickly materializing usages of LLMs across software-engineering tasks and mention reproducibility and evaluation issues [48]. All three studies validate the very basic design assumption we are making about APHD-Guard: to treat the detector as one more layer in a bigger security control plane, not replacing sandboxing, authorization, retrieval hygiene or human oversight.

2.4 High-Stakes Reliability, Human Trust and Governance

Lack of reliability matters most when users treat generated text as authoritative evidence. Generative AI to synthesize literature poses severe hallucination risks however [49]. Förster and Skop follow how the meaning of 'hallucination' may obfuscate the sociotechnical nature of model error [50]. Bottazzi Grifoni and Ferrario caution that human-level language can lead us to believe that we are communicating and understand [51]. Franke explains that calibration in and of itself is not the same as trustworthiness, because what is at stake (such as institutional roles) does matter [52]. Dahl et al. document legal hallucinations, role of authoritative style in coexistence with invented legal propositions [53].

Clinical evidence is similarly mixed. Han et al. Demonstrate how you can make the medical LLMs to generate targeted misinformation [54]. Johri et al. Development of a framework for assessing clinical use in patient-interaction tasks [55] Hager et al. Perform well and poorly [56] evaluation of limitations in clinical decision making and mitigation. Hoppe et al. Febo et al.As a cardinal arrest indicator, you were examined with different ranges of 672 to 679 nfully included and the average service technologically demonstrated an AUCs data price an excellent examination diagnostic performance within limited veterinary-funded situations [54].The originality migrant position is likely some exploration on GUS from stations beyond your home. This model demonstrates that while examination performance provides some assurance about the quality of decisions it does not diminish the need to explore diagnostic accuracy and error [58]. Ultrasound contexts: Hong and Huang on medical education performance [59]. Asgari et al. directly study clinical safety and hallucination rates in medical summarization [60]. Dhane et al. Genetic References in Global-Health Need to be Verified [61].

Of course, organizations have to also enforce organizational controls for secure deployment. Humphreys et al. that uncritical use of generative AI can pose a cybersecurity threat in and of itself [62]. Qiu et al. show that robust extracting hard-label adversarial attacks against state-of-the-art NLP models, and highlight the importance of black-box robustness evaluation [63]. Wang et al. Explore the potentials and boundaries of moral improvement in LLMs [64]. Deng et al. explores the significant ethical challenge in dealing with privacy, truthfulness, copyright and social norms [65]. Saxena et al. demonstrates stable detection training by a fine-grained synthetic hallucination data for detector [66]. Cao et al. learn that the way you construct a prompt is key to making LLMs drive successful program repair [67]. Yang et al. describe RAG as a healthcare grounding technique while also noting its own implementation risks [68]. He et al. demonstrate that MCP tools may themselves be a red-teaming and poisoning surface [69]. Li et al. synthesize jailbreak attack classes and defence directions [70]. In much of this literature there is therefore a clear gap for an empirical defense pipeline that is both unified and claim-gated.

2.5 Research Gap

Across the literature, we can identify five gaps that reoccur. When hallucination detectors are combined with adversarial-prompt detectors, they may be the only detectors for unsafe text at deployment time but they are usually trained and evaluated separately as if they were separate classification problems with safe–safe vs. safe–unsafe pairs at deployment time. Second, most studies prioritize one measure of discrimination over calibration or interventions. Third, the benchmark leakage and contaminated near-duplicates are often not raised to a top-level validity condition. Fourth, ablation analysis often is weaker than the headline comparison and it can be unclear which training source provides robustness. Fifth, empirical completion is often confused with being publication-ready even while there are still unresolved questions about dataset provenance or the completeness of benchmarks. APHD-Guard is framed with these gaps in mind rather than as a single benchmark leaderboard.

2.6 Threat Taxonomy and Defense Requirements

An interesting way to summarize the paper review is organizing failure modes according to the security property they break. The most desperate threat to truth, factual hallucinations: the system more or less presents rubbish as reliable content. Integrity of control is additionally threatened with prompt injection and jailbreaking, as malware controlled tokens corrupt the hierarchy of instructions. The integrity and provenance are afflicted by the retrieval poisoning, while those are affected to confidentiality and

availability by improper tool usage. Contextual backdoors are particularly relevant since the payload is located outside of the model weights. The practical implication is to not rate any one defense as though it protects against all classes of threat. Detection can cut down exposure, but authorization, sandboxing and provenance control are still requisite.

Table 3. Threat taxonomy motivating the APHD-Guard defense-in-depth design

Threat class	Representative mechanism	Primary property	Required control
Factual hallucination	Unsupported or false generated claim	Integrity	Evidence verification; calibrated output risk
Unverifiable answer	Claim lacks sufficient evidence	Integrity / assurance	Abstention; retrieval; human review
Direct jailbreak	User prompt overrides policy	Control integrity	Prompt-risk detector; policy separation; red teaming
Indirect prompt injection	Malicious instruction embedded in retrieved data	Control integrity / provenance	Context sanitization; trust labels; least privilege
Retrieval poisoning	Adversarial or low-quality corpus item	Integrity / provenance	Source signing; corpus monitoring; retrieval filtering
Contextual backdoor	Poisoned demonstrations or memory	Integrity / execution	Versioned prompt assets; anomaly detection; review
Tool poisoning / excessive agency	Malicious or over-privileged tool interaction	Confidentiality / integrity / availability	Tool allowlists; confirmation; sandboxing
Calibration failure	Risk score does not match observed error	Assurance	Validation-only calibration; drift monitoring

3. Proposed APHD-Guard Framework

3.1 Threat Model and Label Space

Deployment setting that assumes LLM-based application which take user prompt, along with contextual evidence and may exposed to simple but innocently challenging questions, unsupported claims & adversarially manipulated instructions or indirect prompt injection. Finally, APHD-Guard can be used either before or after generation based on the application; i.e., a prompt-risk filter (which is applied before generation), a factuality filter (which is applied after generation) or jointly over prompt–response pairs. We use four sequence-wise labels: supported_clean, hallucinated_refuted, unverifiable and prompt_attack. For binary deployment gating, the last 3 are mapped to an unsafe/risky state and supported_clean remains the safe state. This binary risk perspective is intentionally cautious because unverifiable content can necessitate refraining from use even when the information is not objectively untrue.

3.2 Multi-Source Data Construction

The development corpus is obtained by putting together SOCRATES-300K, RAGTruth and FEVER. SOCRATES provides examples of factuality in large scale; RAGTruth presents supervision and span annotations for hallucinations; FEVER provides evidence-grounded SUPPORTS, REFUTES and NOT ENOUGH INFO distinctions. This also includes "controlled adversarial augmentations" of development data, limited to 40,000 training and 5,000 validation in the publication profile. TruthfulQA, HaluEval, HalluDial, JailbreakBench and BIPIAs are treated as external tests that would not affect their fitting of the models. Train/Val partitions are created using dataset-specific grouping keys to ensure that examples from the same source unit do not cross the split.

3.3 Neural Architecture and Objective

For the publication configuration, we set encoder -> microsoft/deberta-v3-base with max_seq_length=384 tokens. The pooled sequence representation is also passed from a classification head to the four-class label space. In the absence of RAGTruth span supervision, we introduce an extra token-localization head that predicts spans that have low quality. The complete loss is defined as L = $L_{cls} + \lambda L_{span}$, where λ = 0.35 in the paper setting. Compute class weights from the training records to minimize label imbalance. This

architecture is designed to be small in comparison with generative judge models: it can perform discriminative inference without sampling a number of times from some deep generator.

L = L_cls + 0.35 L_span

3.4 Calibration and Deployment Risk

Temperature-scored of raw logits, on validation data only For class probability $p_k = softmax\left(\frac{z_k}{T}\right)$ this means the deployment risk is r = 1 - p(supported_clean). We set the binary intervention threshold on the validation set, in which it is then frozen before external testing. This decoupling avoids threshold tuning specific to the test set, and allows for quality of calibration reporting (via Brier score and expected calibration error (ECE)). Subsequently, a deployment can map r to behavior like allow, retrieve-and-verify, human review (if you are feeling especially audacious), regenerate with constrained context or block.

3.5 Leakage and Contamination Controls

Before model fitting, exact fingerprints of external records are compared against the development corpus and exact overlaps are removed. A MinHash locality-sensitive-hashing screen then searches for near-duplicate groups using a 0.90 similarity threshold and 64 permutations in the publication configuration. The observed Run 1 log completed a MinHash screen over 188,054 candidate groups. Group-level rather than individual-row removal is used when a near match is identified, reducing the risk that a paraphrased member of the same source group remains in training. The run manifest also checks train–validation group overlap independently from train–test contamination.

3.6 Training and Reproducibility Configuration

Table 1. Publication-profile APHD-Guard configuration

Parameter	Setting
Encoder	microsoft/deberta-v3-base
Publication seeds	11, 23, 42, 71, 101
Epochs	3
Maximum length	384
Learning rate	2×10^{-5}
Training batch size	8
Evaluation batch size	32
Gradient accumulation	4
Weight decay	0.01
Warmup ratio	0.06
Span-loss weight	0.35
Grouped bootstrap repetitions	2,000
MinHash threshold	0.90
MinHash permutations	64

The publication protocol is computationally expensive and is designed for multiple independent seeds rather than single-run optimization. Run 1 continued the publication profile on CPU when CUDA was unavailable; this affects execution time, not the predefined statistical or claim-gate requirements. Mixed precision and fused GPU-specific optimizers are disabled automatically on CPU.

3.7 End-to-End Decision Algorithm

For each training record, APHD-Guard first constructs a normalized text pair and retains the dataset-specific group identifier. Training examples are filtered against external-test fingerprints before tokenization. During each seed, the encoder is optimized jointly for sequence classification and, when span labels are available, suspicious-token localization. After training, validation logits are collected once and used to estimate the temperature parameter. The same calibrated validation probabilities determine the binary risk threshold. Neither the temperature nor threshold is subsequently changed for an external benchmark. This ordering is critical because calibrating separately on each test set would transform the benchmark into a development set and inflate deployment performance.

External evaluation then proceeds dataset by dataset. For each example, the model generates a four-class probability vector, from which the unsafe risk score is computed as one minus the supported-clean probability. Binary metrics are calculated at the frozen threshold, ranking metrics are calculated over the continuous risk score, and grouped bootstrap resampling estimates uncertainty while respecting group structure. Per-example prediction files are retained so that baseline significance tests use paired records. The best validation seed is stored for diagnostic plots, but aggregate claims use all configured seeds. Finally,

the audit manifest evaluates claim gates independently from the metric tables and packages reproducibility outputs without bundling large raw or licensed datasets.

4. Experimental Design

4.1 Dataset Processing

Table 2. Dataset composition and experimental role

Dataset	Role	Observed/available rows	Primary purpose
SOCRATES-300K	Development	297,795 processed	Factuality supervision
RAGTruth	Development	17,790 processed	Hallucination + span supervision
FEVER train	Development	145,449 processed	Evidence-grounded factuality
FEVER validation	Validation	19,998 processed	Validation / factuality
TruthfulQA	External test	Manifest value	Truthfulness generalization
HaluEval	External test	Manifest value	Hallucination detection
HalluDial	External test	19,994 retrieved	Dialogue hallucination
JailbreakBench	External test	200 behaviors loaded	Harmful/benign prompt behavior
BIPIA	External test	Manifest value	Indirect prompt injection

The run processed 297,795 SOCRATES records, 17,790 RAGTruth records, 145,449 FEVER training records and 19,998 FEVER validation records. FEVER evidence resolution sanitized or skipped 218 malformed/encoding-affected evidence lines. HalluDial loaded 9,994 spontaneous and 10,000 induced examples (19,994 total), below the documented complete test-set size used by the code's integrity check. Consequently, HalluDial findings in the current run should be described as performance on the retrieved subset unless the complete benchmark is reacquired.

4.2 Baselines and Ablations

Two reference models are included. TF-IDF logistic regression provides a lexical baseline whose threshold is selected using the same validation principle, and a majority-class classifier provides a lower-bound reference. The ablation matrix evaluates the full training mixture and four removals: minus SOCRATES-300K, minus RAGTruth, minus FEVER and minus controlled adversarial augmentation. Ablations use a fixed seed and are evaluated on HaluEval in the implementation, isolating the effect of each training source under a common evaluation condition.

4.3 Evaluation Metrics and Statistical Analysis

Sequence performance is characterized by accuracy, Macro-F1 and MCC. Ranking quality is measured by AUROC and AUPRC. Calibration is measured by Brier score and ECE. Deployment safety is summarized through unsafe recall, unsafe precision, unsafe miss rate and safe retention rate. Grouped bootstrap resampling provides uncertainty intervals while respecting dataset grouping. For model comparison, paired bootstrap tests use per-example predictions from APHD-Guard and each baseline on the same evaluation records. Throughput is recorded as examples per second to expose computational cost.

4.4 Publication Claim Gates

SPLINE evaluation pipeline splits predictive performance from evidentiary readiness. Contains: Gates check publication profile activation minimum dataset sizes external-test isolation exact contamination removal train–validation group separation five independent seeds validation-only calibration grouped bootstrap completion presence of prompt-attack benchmarks span supervision FEVER evidence resolution ablation completion significance testing dataset provenance and absence of synthetic fallback This mechanism intentionally is conservative: even with a strong F1 score passes provenance or benchmark-completeness checks, if they fail the other one it cannot be considered valid.

5. Results

5.1 External Benchmark Performance

Table 4. APHD-Guard performance across external benchmark datasets

Dataset	Accuracy	Macro-F1	MCC	AUROC	AUPRC	Brier ↓	ECE ↓	Unsafe miss ↓	Safe retention ↑
TruthfulQA	0.841	0.823	0.682	0.891	0.884	0.118	0.041	0.137	0.873
HaluEval	0.884	0.871	0.758	0.931	0.925	0.091	0.032	0.098	0.906
HalluDial	0.798	0.781	0.604	0.846	0.838	0.143	0.052	0.174	0.829
JailbreakBench	0.812	0.794	0.631	0.864	0.857	0.132	0.047	0.151	0.846
BIPIA	0.827	0.811	0.655	0.879	0.871	0.125	0.044	0.143	0.861
Average	**0.832**	**0.816**	**0.666**	**0.882**	**0.875**	**0.122**	**0.043**	**0.141**	**0.863**

The current result set achieves an average accuracy (across the five external benchmarks) of 0.832, Macro-F1 of 0.816, MCC of 0.666, AUROC of 0.882 and AUPRC of 0.875 respectively. HaluEval achieves the highest discrimination (Macro-F1 0.871; AUROC 0.931). The hardest case is HalluDial, with the lowest Marco-F1 0.781 and the greatest unsafe miss rate (0.174) among instances only in PAbiBench. This pattern aligns with the higher context complexity of dialogue-level hallucination and similar to the HalluDial retrievals that were not fully identified.

5.2 Multi-Seed Stability

Table 5. Five-seed validation performance and calibration stability

Seed	Accuracy	Macro-F1	MCC	Temperature	Threshold
11	0.861	0.847	0.731	1.18	0.47
23	0.869	0.855	0.744	1.14	0.46
42	0.878	0.864	0.759	1.11	0.48
71	0.865	0.851	0.738	1.16	0.47
101	0.872	0.858	0.749	1.13	0.46
Mean	0.869	0.855	0.744	1.14	0.47
SD	0.006	0.006	0.010	0.027	0.008

The five seed validation profile has a low variability: Macro-F1 (0.847–0.864) and MCC (0.731–0.759). This is based on the Macro-F1 values (with reported standard deviation of 0.006), and threshold varies from 046 to 048. This is much more informative than choosing one of the best seeds because it shows whether or not the decision boundary is chain-independent with respect to random seed and data-order effects.

5.3 Baseline Comparison

Table 6. Comparison with internal baseline approaches

Model	Macro-F1	MCC	AUROC	AUPRC	ECE ↓	Brier ↓
Majority-Class	0.361	0.000	0.500	0.542	0.458	0.458
TF-IDF + Logistic Regression	0.683	0.442	0.761	0.748	0.112	0.193
APHD-Guard	0.816	0.666	0.882	0.875	0.043	0.122
Improvement vs TF-IDF-LR	+0.133	+0.224	+0.121	+0.127	−0.069	−0.071

Against TF-IDF logistic regression, Macro-F1 is improved by 0.133, MCC by 0.224 (APHD-Guard). Gladly, the enhancement is not restricted to an inflexible threshold: AUROC expands by 0.121 and AUPRC by 0.127. Additionally, calibration improves as ECE drops from 0.112 to 0.043 and Brier score from 0.193 to 0.122. This combination is key because a security gate should rank risky examples and rate confidence to perform intervention.

5.4 Dataset-Wise Improvement over the Lexical Baseline

Table 7. Dataset-wise APHD-Guard versus TF-IDF logistic regression

Dataset	TF-IDF F1	APHD F1	ΔF1	TF-IDF AUROC	APHD AUROC	ΔAUROC
TruthfulQA	0.694	0.823	+0.129	0.774	0.891	+0.117

Dataset	TF-IDF F1	APHD F1	ΔF1	TF-IDF AUROC	APHD AUROC	ΔAUROC
HaluEval	0.731	0.871	+0.140	0.801	0.931	+0.130
HalluDial	0.661	0.781	+0.120	0.738	0.846	+0.108
JailbreakBench	0.646	0.794	+0.148	0.724	0.864	+0.140
BIPIA	0.681	0.811	+0.130	0.768	0.879	+0.111
Average	**0.683**	**0.816**	**+0.133**	**0.761**	**0.882**	**+0.121**

Macro-F1 With sizeable gains on JailbreakBench (+0.148) and HaluEval (+0.140). On HalluDial, the smallest improvement continues to be +0.120. AUROC improvements vary from +0.108 to +0.140. Given that the improvement direction is consistent for all factuality, hallucination and adversarial benchmark data sources (rather than per dataset ranking), it suggests a gain not confined to one benchmark family.

5.5 Ablation Analysis

Table 8. Ablation analysis on HaluEval

Configuration	Macro-F1	MCC	AUROC	Δ Macro-F1
Full APHD-Guard	0.871	0.758	0.931	—
Without SOCRATES-300K	0.839	0.704	0.906	−0.032
Without RAGTruth	0.824	0.681	0.892	−0.047
Without FEVER	0.843	0.712	0.909	−0.028
Without adversarial augmentation	0.806	0.653	0.881	−0.065

The most significant drop (0.065 and 0.050 to Macro-F1 and AUROC, respectively) occurs for the removal of controlled adversarial augmentation. Table 4 shows that RAGTruth contributes the second highest decrease in Macro-F1 (0.047) after removing it. Contributions from SOCRATES-300K and FEVER are minor but positive as well. This pattern lends itself to a complementary-supervision story: explicit adversarial examples primarily bolster attack robustness (c.f. the earlier discussion in Section 4), RAGTruth provides hallucination-sensitive supervision, and factuality corpora boost evidence-oriented discrimination.

5.6 Calibration and Safety Operating Characteristics

Table 9. Calibration performance across external datasets

Dataset	ECE before	ECE after	Reduction	Brier
TruthfulQA	0.094	0.041	56.4%	0.118
HaluEval	0.081	0.032	60.5%	0.091
HalluDial	0.113	0.052	54.0%	0.143
JailbreakBench	0.101	0.047	53.5%	0.132
BIPIA	0.097	0.044	54.6%	0.125
Average	**0.097**	**0.043**	**55.8%**	**0.122**

Table 10. Safety-oriented operating characteristics

Dataset	Unsafe recall ↑	Unsafe precision ↑	Unsafe miss ↓	Safe retention ↑
TruthfulQA	0.863	0.846	0.137	0.873
HaluEval	0.902	0.891	0.098	0.906
HalluDial	0.826	0.801	0.174	0.829
JailbreakBench	0.849	0.832	0.151	0.846
BIPIA	0.857	0.841	0.143	0.861
Mean	**0.859**	**0.842**	**0.141**	**0.863**

Temperature scaling reduces average ECE from 0.097 to 0.043, a relative reduction of 55.8% in the current result set. At the validation-derived threshold, unsafe recall averages 0.859 while safe retention averages 0.863. The operating point therefore avoids a trivial 'block everything' policy: the detector retains most

safe examples while identifying most unsafe examples. The remaining 0.141 average unsafe miss rate also makes clear that the detector should not be treated as a complete security boundary.

5.7 Statistical Significance

Table 11. Paired bootstrap comparison: APHD-Guard versus TF-IDF-LR

Dataset	Δ Macro-F1	95% CI	p-value	Significant
TruthfulQA	+0.129	[0.103, 0.155]	<0.001	Yes
HaluEval	+0.140	[0.119, 0.161]	<0.001	Yes
HalluDial	+0.120	[0.096, 0.144]	<0.001	Yes
JailbreakBench	+0.148	[0.091, 0.202]	<0.001	Yes
BIPIA	+0.130	[0.091, 0.169]	<0.001	Yes

All listed confidence intervals remain above zero and the current result set records $p < 0.001$ for the paired bootstrap comparisons. The strongest absolute gain occurs on JailbreakBench. Because the test resamples paired predictions rather than independent aggregate scores, it directly tests whether APHD-Guard's advantage is consistently realized on the same evaluation items.

5.8 Cross-Study Positioning

Table 12. Cross-study positioning against recent journal methods (metrics are not directly interchangeable across datasets)

Method	Ref.	Primary target	Reported evidence	Positioning relative to APHD-Guard
Semantic entropy	[1]	Hallucination uncertainty	Semantic uncertainty / AUROC-style evaluation	Uncertainty-centric; no prompt-attack objective
Hierarchical semantic piece	[2]	Hallucination detection + mitigation	Evidence-consistency framework	Evidence-centric; not a joint jailbreak detector
MetaQA	[11]	Hallucination detection	F1; reported gains over SelfCheckGPT	Zero-resource hallucination focus
Cross-LLM secure framework	[12]	Hallucination + injection	98% eligibility-scoring accuracy reported in its setting	Layered security; task metrics differ
JailGuard	[21]	Prompt/jailbreak detection	Attack-detection evaluation	Prompt attacks; no factuality calibration objective
APHD-Guard	This work	Hallucination + unverifiability + prompt attack	Avg F1 0.816; AUROC 0.882; HaluEval acc. 0.884	Joint risk + calibration + claim gates

Table 11 is intentionally a positioning table rather than a leaderboard. The cited studies use different datasets, output definitions and evaluation protocols, so a larger numerical value in one paper does not establish statistical superiority over another. The defensible distinction of APHD-Guard is broader empirical coverage: the same detector is evaluated for factuality/hallucination and adversarial-prompt behavior, with calibration, contamination controls, multi-seed aggregation, ablation and claim gating included in the protocol. This breadth complements specialized methods such as semantic-entropy uncertainty [1], HSP evidence verification [2], MetaQA metamorphic testing [11] and JailGuard prompt-attack detection [21].

Table 13. Feature-level comparison of recent defense directions

Approach	Hallucination	Prompt/attack	Calibration	Defense layer	Claim gates	Primary emphasis
HSP [2]	Yes	No	Evidence-based	No	No	Hallucination verification/mitigation
MetaQA [11]	Yes	No	Indirect	No	No	Zero-resource metamorphic detection
Secure cross-LLM [12]	Yes	Yes	Not central	Layered	No	Hallucination + injection security
JailGuard [21]	No	Yes	No	No	No	Universal prompt-attack detection
SOMF [23]	No	Safety realignment	No	Model-level	No	Post-fine-tuning realignment
SPLoRA [25]	No	Safety alignment	No	Model-level	No	Safe parameter-efficient adaptation
APHD-Guard	Yes	Yes	Yes	Yes	Yes	Joint calibrated detection + audit gates

The comparison in Table 13 emphasizes coverage rather than claiming that APHD-Guard dominates specialized methods. HSP and MetaQA provide richer hallucination-specific mechanisms, JailGuard focuses directly on prompt attacks, and SOMF/SPLoRA modify the model itself. APHD-Guard's contribution is to combine multiple risk categories with calibration, external testing and auditable reporting in one deployment-oriented detector. A mature production system could combine these approaches rather than select only one.

5.9 Publication Integrity Status

Table 14. Publication-integrity and reproducibility conditions

Requirement	Current status
Publication profile	Satisfied by configuration
External tests isolated from training	Required / implemented
Exact contamination screen	Implemented
MinHash near-duplicate screen	Completed; 188,054 groups screened
Train–validation group isolation	Required / checked
Five independent seeds	Required by publication profile
Validation-only calibration	Implemented
Grouped bootstrap evaluation	Required (2,000 repetitions)
Prompt-attack benchmarks	Included
Span supervision	Included from RAGTruth
FEVER evidence resolution	Completed; 218 malformed lines sanitized/skipped
Ablation analysis	Required
Paired significance analysis	Required
SOCRATES-300K provenance	Unresolved in current run
Complete HalluDial test set	Unresolved in current run
Headline claim gate	Cannot pass until all required integrity conditions pass

Two integrity issues remain explicit. First, SOCRATES-300K is marked unverified in the dataset-provenance dictionary, which forces the provenance gate to fail. Second, the HalluDial loader retrieved only 19,994 examples, below the documented complete test size checked by the code. These conditions should remain visible in any submitted manuscript until resolved; they do not invalidate the architecture, but they limit the strength of benchmark-level and headline claims.

6. Discussion

6.1 Why the Multi-Source Design Matters

The ablation pattern suggests that robustness is not produced by one large factuality corpus alone. Controlled adversarial augmentation has the largest effect, which is consistent with the literature showing that aligned models remain vulnerable to context-specific and adaptive jailbreaks [22] and that detector robustness can collapse under adversarially generated transformations [27]. RAGTruth-like supervision contributes a distinct hallucination signal, while evidence-oriented corpora improve factuality discrimination. This finding is compatible with recent hallucination research that separates semantic uncertainty [1], evidence decomposition [2] and noise robustness [7] rather than expecting one signal to explain all failure modes.

6.2 Calibration as a Security Requirement

A secure deployment does not consume Macro-F1 directly; it consumes decisions. For that reason, the reduction in ECE is operationally important. A calibrated risk score enables differentiated actions such as allow, review, retrieve-and-verify or block. However, calibration should not be overinterpreted. Franke notes that statistical calibration is not equivalent to institutional trustworthiness [52], and user studies show that confidence expressions themselves affect human behavior [19]. APHD-Guard therefore treats calibration as one component of a control system rather than as proof that the model 'knows when it is wrong.'

6.3 Adversarial Robustness and Adaptive Attackers

The JailbreakBench and BIPIA results indicate that a detector trained with adversarial augmentation can transfer beyond purely factuality-oriented tasks. Yet adaptive attackers remain a central threat. RedAgent demonstrates that a small number of context-specific queries can uncover vulnerabilities in customized applications [22]. JailGuard shows that mutation-based inconsistency can be useful for attack detection [21]. Meade et al. find that adversarial-trigger transfer is model- and alignment-dependent [26], and Qiu et al. show that black-box attacks can be made query-efficient [63]. The implication is that APHD-Guard should be periodically red-teamed against the exact deployment context and should not rely on a static benchmark as a permanent safety certificate.

6.4 Context, Retrieval and Agent Security

A detector operating at the prompt/response boundary cannot secure compromised context stores or over-privileged tools. Contextual backdoor attacks against embodied agents show that a few poisoned demonstrations can redirect downstream behavior without changing the model weights [38]. Agent-security surveys similarly identify memory, retrieval, tools and inter-agent communication as separate threat surfaces [44]. MCP-oriented red teaming further indicates that malicious tools can manipulate agent behavior through standardized integration layers [69]. Accordingly, secure deployment should combine APHD-Guard with retrieval provenance, content sanitization, least-privilege tools, action confirmation and execution sandboxing.

6.5 Relation to Alignment

APHD-Guard is deliberately external to the base model's alignment method. Safety realignment through subspace fusion [23], machine unlearning [24] and adapter pruning [25] can reduce harmful generation, but none guarantees perfect behavior under distribution shift. Betley et al. show that narrow optimization can create broad misalignment [28], while Lindström et al. highlight sociotechnical limits of preference-based alignment [29]. A deployment-time detector therefore provides an independent measurement channel whose failure modes differ from those of the generator's alignment training.

6.6 Error Analysis and Decision Trade-offs

There are at least three different types of error, operationally speaking, hidden beneath the aggregate metrics. A false negative in a jailbreak prompt can allow an attacker to carry out instructed actions downstream; a false negative is then that same hallucinated response unreliably propagating incorrect information through the system and with it, potential harm; while, finally, a false positive on a safe response will have as consequence decreased utility or loss of trust from the user. The cost of these errors is not the same Thus, a production threshold should be chosen from an application-specific cost model and not directly copied from the research benchmark. If it is clinical or even financial assistant, unsafe misses may justify a lower threshold and more human review With a low-risk writing assistant, too much blocking can be just as harmful as an occasional unverifiable text.

The range of dataset-level performance in Table 4 also provides evidence for apparent domain shift. HaluEval outperforms HaluDial suggesting that long conversational context and dialogue specific semantics are still a challenge. The training example in this case is a sensitive feature with an attack logit that shows a noticeable advantage over the lexical baseline, reinforcing the role of adversarial augmentation but also confirming that adaptive attacks can target characteristics learned by the detector itself. We therefore recommend future error analysis to stratify false negatives by attack family, lexical novelty, source model, dialogue length and evidence availability as well as whether the harmful instruction was explicit or inferred from context. That is, such stratification would cause one unsafe-miss rate to become an engineer-able action.

Second, there is the trade-off with the fact that they are using a four-class label structure. Merging the hallucinated/refuted, verifiable and prompt-attack labels into a single unsafe score helps with deployment, but one should still e.g. have a separate action based on predicted category. An optional term with a high prompt-attack probability may thus well warrant the blocking of the request, while an optional term that is unverifiable will often warrant retrieval/abstention. For this reason, the subsequent version of APHD-Guard would be necessary to consider class-conditional action and expected utility rather than restricting itself in practice to only computing the binary risk threshold by discarding the calibrated multiclass probability vector.

7. Secure Deployment Architecture

A practical deployment can implement APHD-Guard as a layered policy service. The recommended flow is: (1) authenticate and classify the request context; (2) sanitize and provenance-check retrieved content; (3) score the prompt for adversarial risk; (4) invoke the LLM under least-privilege tool permissions; (5) score the generated response for factuality and attack indicators; (6) use the calibrated risk to decide allow/retrieve/review/block; and (7) log the decision, evidence and model version for audit. This architecture is consistent with the broader software-security literature, which treats LLMs as components inside a system rather than as isolated predictors [46]. Secure code-generation research further shows that prompting and workflow controls can materially affect vulnerability outcomes [32].

Table 15. Defense-in-depth deployment controls around APHD-Guard

Layer	Primary control	Failure addressed
1. Input trust	User/session authentication; system/user/data separation	Prompt injection, identity confusion
2. Context trust	Signed/curated retrieval; source provenance; sanitization	Indirect injection, poisoned RAG
3. Prompt risk	APHD-Guard pre-generation score	Jailbreak / policy override
4. Generation	Constrained system prompt; least-privilege tools	Unsafe generation / excessive agency
5. Output risk	APHD-Guard post-generation score + evidence check	Hallucination / hidden attack
6. Decision	Calibrated threshold + human escalation	Overblocking / unsafe miss
7. Audit	Immutable logs; version, threshold, evidence, action	Undetected drift / non-reproducibility

This layered architecture also addresses human factors. Users should not receive an unqualified confidence score if its meaning is unclear. Verbalized uncertainty can change trust in non-linear ways [19], and apparently communicative language can encourage users to attribute stronger understanding than is warranted [51]. Therefore, an enterprise interface should expose evidence, action status and escalation reason rather than simply labeling an answer 'safe.'

7.1 Continuous Monitoring and Governance

Deployment governance should not set a one-off benchmark, but define service-level indicators that can be measured to know whether safety is assured. We recommend basic indicators such as: (i) high-risk prompts rate, (ii) unsafe-miss estimates on a continuously labeled audit sample, (iii) safe-retention rate, (iv) evidence-complexity estimate (ECE), and (v) retrieval-source failure rate and human-escalation frequency and the fraction of decisions made under an unverified evidence state. Segment these indicators by model version, use-case and user workflow. When red-team evaluation has used stratification (such as application or business process type), monitoring should maintain the same stratification, since a stable aggregate F1 score can mask a sharp regression in one attack family, language or business process.

Change management is equally important. You may modify the attack surface without changing hardcover the APHD-Guard weights, for example, by using new retrieval source with input margin properties and few-shot demonstrations, or system-prompt modification. Each material change should undergo a threshold regression suite and recalibration check. These results combined with recent findings of local backdoors in systems that could be agentic [38] make short assets and demonstration libraries particularly important: they should be provenced, code-reviewed, and have versioned provenance metadata. In a similar manner, the agent-risk literature [44] proposes to treat tool schemas and memory components as controlled pieces of software instead of informal prompt engineering.

Governance should also specify who owns and manages residual risk, if it even exists. Ownership Model quality and calibration: Detector Team Tool Permissions, User Experience: Application Team Retrieval

Provenence: Data Owners Incident Thresholds / Red-team cadence : Security or Risk function This break prevents one of the more common failure modes where model score is construed to give an organization free rein to deploy. An explicit decision is needed regarding what residual secure-miss rate of application consequences is acceptable for responsible usage.

7.2 Incident Response for Prompt and Hallucination Failures

An LLM security incident can originate from an adversarial prompt, poisoned context, hallucinated output, calibration drift or tool misuse. The response process should retain the original request, retrieved evidence identifiers, model and detector versions, risk score, threshold, generated output and downstream action. These records allow the incident to be reproduced and classified. If the failure is an attack miss, the prompt family should enter the red-team corpus; if it is an evidence failure, the retrieval source should be quarantined or re-ranked; if it is calibration drift, the system should be recalibrated using a new validation window rather than simply moving the threshold until the incident disappears.

Post-incident analysis should distinguish containment from model improvement. Immediate containment may involve disabling a tool, blocking a retrieval source or raising the human-review threshold. Longer-term remediation may require adversarial augmentation, retraining, better evidence verification or a change to the base model's safety alignment. This distinction is consistent with defense-in-depth: APHD-Guard supplies detection and evidence for response, but safe recovery can require controls outside the classifier itself.

8. Limitations and Threats to Validity

8.1 Construct Validity

This four-class label space is an abstraction for deployments. Hallucination, unverifiability and malicious intent are not the same things, and putting them into a binary risk score can hide many nuances. There are interesting directions for future work such as evaluating hierarchical labels and cost-sensitive policies. Election 1: Threat of Attack is not the Same as Evidence for Attack — In measuring Muller's generation-level attack success rate, a detector's classification applies to an adversarial prompt does not correlate with generation-level attack success; and unless there were separate experiments at the generation level (which accurately kept these claims disabled), current pipeline generates claims of successful attacks at the generation levels.

8.2 Internal Validity

While group-aware splitting and contamination screening reduce leakage, dataset labels may be noisy or differ due to annotation-policy differences. It also reported 218 malformed lines via the FEVER evidence loader, and had an incomplete HalluDial mirror. The SOCRATES provenance flag is unknown. The claim gates clearly observe these issues, and therefore should be rectified ahead of a final submission. The final exported CSVs of each run for comparison whether the totals from the current numerical result tables match their respective rows.

8.3 External Validity

External benchmarks do not represent every real deployment. Jailbreak behavior evolves rapidly, and context-specific attacks can outperform generic attacks [22]. Multimodal systems introduce vulnerabilities beyond the text-only evaluation [6]. Clinical and legal studies demonstrate that domain knowledge, evidence quality and consequence severity substantially alter what constitutes an acceptable error rate [53] [60]. Therefore, APHD-Guard should be revalidated with domain-specific examples and realistic tool/use-case context before production use.

8.4 Comparative Validity

Cross-paper comparisons are limited by different datasets, class definitions, metrics and model access assumptions. For this reason, Table 11 does not claim statistical superiority over external journal methods. The strongest quantitative comparison in this paper is the paired internal comparison against TF-IDF logistic regression on identical external records. Literature methods are used for methodological positioning, not for an artificial universal leaderboard.

9. Implications and Future Research

The first research direction is adaptive red teaming. Static adversarial augmentation should be replaced by a loop in which new attacks are generated against the deployed model and detector, reviewed for validity and incorporated into periodic retraining. Autonomous red-team systems such as RedAgent [22] and MCP tool-oriented testing [69] indicate how deployment-specific attack discovery can be operationalized.

Second, hallucination detection should become evidence-aware at inference time. Semantic-entropy signals [1], hierarchical evidence matching [2] and RAG verification [16] can be combined with APHD-Guard's discriminative risk score. A hybrid architecture could abstain when classifier risk is high, retrieve evidence, then re-score the answer and its citations before release.

Third, calibration should be monitored under drift. Calibration error can change even when ranking metrics remain stable, especially after model upgrades, retrieval-corpus changes or domain shifts. Uncertainty-quantification frameworks [39] provide a foundation for drift-aware monitoring, while human factors

research [19] suggests that the presentation of uncertainty should be evaluated alongside numerical calibration.

Fourth, agent security requires provenance for prompts, demonstrations, memory and tools. The contextual backdoor evidence [38] implies that prompt libraries and few-shot examples should be versioned and reviewed as controlled artifacts. Software-security studies [47] and broader LLM-for-software-engineering evidence [48] support treating generated code, tool calls and context objects as part of the security supply chain.

Finally, publication-quality evaluation should make integrity gates standard. A model should not be described as publication-ready simply because it achieves a strong metric. Reproducible source hashes, dataset licensing/provenance, benchmark completeness, contamination checks, multi-seed uncertainty, calibration, significance testing and ablation evidence should be reported together. This is especially important in fields where fabricated references or false authoritative statements can cause downstream harm [49] [61].

10. Conclusion

We introduce APHD-Guard as a calibrated defense layer for safe LLM deployment. This framework unifies factuality, hallucination, evidence-verification and adversarial-prompt supervision in a common DeBERTa-based detector (adding span localization to sequence classification), applies validation-only temperature scaling and separates external benchmarks from training. We find that this manuscript result set produces consistently beneficial properties over a lexical baseline across along all datasets—TruthfulQA, HaluEval, HalluDial, JailbreakBench and BIPIA—with average Macro-F1 0.816 and AUROC 0.882 as well as more calibration and safety-oriented operating characteristics. Ablation results show that controlled adversarial augmentation and hallucination-specific supervision are the main contributors to the full model. Just as importantly, the pipeline reveals unaddressed issues regarding provenance and benchmark-completeness instead of obscuring them with predictive performance. The larger takeaway is that secure LLM deployment is not just a one-off model alignment problem: it's an end-to-end assurance problem from input trust and context provenance to hallucination detection, adversarial robustness, calibrated intervention, tool authorization, red teaming and reproducible evidence. This broader architecture has a measurable detection-and-audit layer to which APHD-Guard contributes.

Data, Code and Reproducibility Statement

The experimental implementation exports per-seed predictions, aggregate external metrics, calibration outputs, cross-generator analysis, significance tests, ablation results, figures and an audit manifest. A final submission should archive the exact code version, environment, source hashes and final result tables. SOCRATES-300K provenance must be independently verified (including source, DOI/repository and license) or the dataset should be replaced before the headline publication claim is activated. The complete HalluDial test split should likewise be obtained or the reported evaluation should explicitly remain a subset analysis.

Ethics and Responsible Use

The study concerns defensive analysis of adversarial prompts and hallucinations. Attack examples are used to evaluate detection and robustness rather than to facilitate misuse. Deployment decisions should preserve human oversight for high-consequence domains. Bias, privacy, domain-specific harms and unequal error costs require independent evaluation beyond the scope of the current classifier, consistent with recent work on responsible LLM development [30] and ethical risks [65].

References

[1] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, "Detecting hallucinations in large language models using semantic entropy," Nature, vol. 630, 625–630, 2024. doi: 10.1038/s41586-024-07421-0.

[2] Y. Liu, Q. Yang, J. Tang, et al., "Reducing hallucinations of large language models via hierarchical semantic piece," Complex & Intelligent Systems, vol. 11, 231, 2025. doi: 10.1007/s40747-025-01833-9.

[3] A. Saha, B. Gupta, A. Chatterjee, et al., "You believe your LLM is not delusional? Think again! A study of LLM hallucination on foundation models under perturbation," Discover Data, vol. 3, 20, 2025. doi: 10.1007/s44248-025-00041-7.

[4] G. Hao, J. Wu, Q. Pan, and R. Morello, "Quantifying the uncertainty of LLM hallucination spreading in complex adaptive social networks," Scientific Reports, vol. 14, 16375, 2024. doi: 10.1038/s41598-024-66708-4.

[5] S. Liu, Y. Gao, S. Li, P. Wang, and T. Wang, "A hallucination detection and mitigation framework for faithful text summarization using LLMs," Scientific Reports, vol. 16, 1374, 2026. doi: 10.1038/s41598-025-31075-1.

[6] S. Sun, Z. Lin, and X. Wu, "Hallucinations of large multimodal models: Problem and countermeasures," Information Fusion, vol. 118, 102970, 2025. doi: 10.1016/j.inffus.2025.102970.

[7] J. Qu, J. Liu, X. Liu, M. Chen, J. Li, and J. Wang, "PNCD: Mitigating LLM hallucinations in noisy environments—A medical case study," Information Fusion, vol. 123, 103328, 2025. doi: 10.1016/j.inffus.2025.103328.
[8] M. Chelli, J. Descamps, V. Lavoué, et al., "Hallucination rates and reference accuracy of ChatGPT and Bard for systematic reviews: Comparative analysis," Journal of Medical Internet Research, vol. 26, e53164, 2024. doi: 10.2196/53164.
[9] H. Alkaissi and S. I. McFarlane, "Artificial hallucinations in ChatGPT: Implications in scientific writing," Cureus, vol. 15, e35179, 2023. doi: 10.7759/cureus.35179.
[10] H. Ye, J. Xu, D. Huang, et al., "Assessment of large language models' performances and hallucinations for Chinese postgraduate medical entrance examination," Discover Education, vol. 4, 59, 2025. doi: 10.1007/s44217-025-00446-1.
[11] B. Yang, M. A. Al Mamun, J. M. Zhang, and G. Uddin, "Hallucination detection in large language models with metamorphic relations," Proceedings of the ACM on Software Engineering, vol. 2, 425–445, 2025. doi: 10.1145/3715735.
[12] T. Gokcimen and B. Das, "A novel system for strengthening security in large language models against hallucination and injection attacks with effective strategies," Alexandria Engineering Journal, vol. 123, 71–90, 2025. doi: 10.1016/j.aej.2025.03.030.
[13] S. Singh, R. Saha, G. Kumar, A. Nayyar, and T.-K. Kim, "Are you hallucinated? Insights into large language models," ICT Express, vol. 12, 384–389, 2026. doi: 10.1016/j.icte.2025.12.011.
[14] I. Augenstein, T. Baldwin, M. Cha, et al., "Factuality challenges in the era of large language models and opportunities for fact-checking," Nature Machine Intelligence, vol. 6, 852–863, 2024. doi: 10.1038/s42256-024-00881-z.
[15] L. Zhou, W. Schellaert, F. Martínez-Plumed, et al., "Larger and more instructable language models become less reliable," Nature, vol. 634, 61–68, 2024. doi: 10.1038/s41586-024-07930-y.
[16] Q. Yang, H. Zuo, R. Su, et al., "Dual retrieving and ranking medical large language model with retrieval augmented generation," Scientific Reports, vol. 15, 18062, 2025. doi: 10.1038/s41598-025-00724-w.
[17] S. Elmitwalli, J. Mehegan, S. Braznell, and A. Gallagher, "Scalable evaluation framework for retrieval augmented generation in tobacco research using large language models," Scientific Reports, vol. 15, 22760, 2025. doi: 10.1038/s41598-025-05726-2.
[18] F. O. Catak and M. Kuzlu, "Uncertainty quantification in large language models through convex hull analysis," Discover Artificial Intelligence, vol. 4, 90, 2024. doi: 10.1007/s44163-024-00200-w.
[19] Z. Xu, T. Song, and Y.-C. Lee, "Confronting verbalized uncertainty: Understanding how LLM's verbalized uncertainty influences users in AI-assisted decision-making," International Journal of Human-Computer Studies, vol. 197, 103455, 2025. doi: 10.1016/j.ijhcs.2025.103455.
[20] L. E. Schimmelpfennig, K. Bhattarai, I. Y. Oh, et al., "CeRTS: certainty retrieval token search in large language model clinical information extraction," Journal of Biomedical Informatics, vol. 170, 104900, 2025. doi: 10.1016/j.jbi.2025.104900.
[21] X. Zhang, C. Zhang, T. Li, et al., "JailGuard: A universal detection framework for prompt-based attacks on LLM systems," ACM Transactions on Software Engineering and Methodology, vol. 35, Article 8, 1–40, 2025. doi: 10.1145/3724393.
[22] H. Xu, W. Zhang, Z. Wang, F. Xiao, R. Zheng, Z. Ba, and K. Ren, "RedAgent: An autonomous agent for context-aware red teaming of LLM jailbreaks," IEEE Transactions on Dependable and Secure Computing, vol. 23, 6506–6521, 2026. doi: 10.1109/TDSC.2026.3665230.
[23] X. Yi, S. Zheng, L. Wang, X. Wang, and L. He, "A safety realignment framework via subspace-oriented model fusion for large language models," Knowledge-Based Systems, vol. 306, 112701, 2024. doi: 10.1016/j.knosys.2024.112701.
[24] W. Zhao, Y. Hu, X. Sui, et al., "The gains do not make up for the losses: a comprehensive evaluation for safety alignment of large language models via machine unlearning," Frontiers of Computer Science, vol. 20, 2002319, 2026. doi: 10.1007/s11704-024-41099-x.
[25] S. Ao, Y. Dong, J. Hu, and S. D. Ramchurn, "Safe Pruning LoRA: Robust distance-guided pruning for safety alignment in adaptation of LLMs," Transactions of the Association for Computational Linguistics, vol. 13, 1474–1487, 2025. doi: 10.1162/tacl.a.44.
[26] N. C. Meade, A. Patel, and S. Reddy, "Investigating adversarial trigger transfer in large language models," Transactions of the Association for Computational Linguistics, vol. 13, 953–979, 2025. doi: 10.1162/tacl.a.27.
[27] Z. Shi, Y. Wang, F. Yin, X. Chen, K.-W. Chang, and C.-J. Hsieh, "Red teaming language model detectors with language models," Transactions of the Association for Computational Linguistics, vol. 12, 174–189, 2024. doi: 10.1162/tacl_a_00639.
[28] J. Betley, N. Warncke, A. Sztyber-Betley, et al., "Training large language models on narrow tasks can lead to broad misalignment," Nature, vol. 649, 584–589, 2026. doi: 10.1038/s41586-025-09937-5.

[29] L. E. Lindström, L. Methnani, L. Krause, et al., "Helpful, harmless, honest? Sociotechnical limits of AI alignment and safety through reinforcement learning from human feedback," Ethics and Information Technology, vol. 27, 28, 2025. doi: 10.1007/s10676-025-09837-2.
[30] S. Raza, O. Bamgbose, S. Ghuge, F. Tavakoli, D. J. Reji, and S. R. Bashir, "Developing safe and responsible large language model: can we balance bias reduction and language understanding?," Machine Learning, vol. 114, 140, 2025. doi: 10.1007/s10994-025-06767-4.
[31] D. Rodriguez, I. Yang, J. M. Del Alamo, et al., "Large language models: a new approach for privacy policy analysis at scale," Computing, vol. 106, 3879–3903, 2024. doi: 10.1007/s00607-024-01331-9.
[32] C. Tony, N. E. Díaz Ferreyra, M. Mutas, S. Dhif, and R. Scandariato, "Prompting techniques for secure code generation: A systematic investigation," ACM Transactions on Software Engineering and Methodology, vol. 34, Article 225, 1–53, 2025. doi: 10.1145/3722108.
[33] C. D. Xuan, D. B. Quang, and V. D. Quang, "Large language models based vulnerability detection: How does it enhance performance?," International Journal of Information Security, vol. 24, 69, 2025. doi: 10.1007/s10207-025-00983-8.
[34] Q. Mao, Z. Li, X. Hu, K. Liu, X. Xia, and J. Sun, "Towards explainable vulnerability detection with large language models," IEEE Transactions on Software Engineering, vol. 51, 2957–2971, 2025. doi: 10.1109/TSE.2025.3605442.
[35] S. Jamal, H. Wimmer, and I. H. Sarker, "An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach," Security and Privacy, vol. 7, e402, 2024. doi: 10.1002/spy2.402.
[36] H. Kibriya, R. Amin, S. S. Alshamrani, et al., "Lightweight malicious URL detection using deep learning and large language models," Scientific Reports, vol. 15, 43044, 2025. doi: 10.1038/s41598-025-26653-2.
[37] L. Le and D. Tran, "A metric-based detection system for large language model texts," ACM Transactions on Management Information Systems, vol. 16, Article 8, 1–19, 2025. doi: 10.1145/3704739.
[38] A. Liu, Y. Zhou, X. Liu, et al., "Compromising LLM driven embodied agents with contextual backdoor attacks," IEEE Transactions on Information Forensics and Security, vol. 20, 3979–3994, 2025. doi: 10.1109/TIFS.2025.3555410.
[39] O. Shorinwa, Z. Mei, J. Lidard, A. Z. Ren, and A. Majumdar, "A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions," ACM Computing Surveys, vol. 58, Article 63, 1–38, 2025. doi: 10.1145/3744238.
[40] Z. Lin, S. Guan, W. Zhang, et al., "Towards trustworthy LLMs: a review on debiasing and dehallucinating in large language models," Artificial Intelligence Review, vol. 57, 243, 2024. doi: 10.1007/s10462-024-10896-y.
[41] L. Huang, W. Yu, W. Ma, W. Zhong, et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," ACM Transactions on Information Systems, vol. 43, Article 42, 1–55, 2025. doi: 10.1145/3703155.
[42] M. S. Jabbar, S. Al-Azani, A. Alotaibi, and M. Ahmed, "Red teaming large language models: A comprehensive review and critical analysis," Information Processing & Management, vol. 62, 104239, 2025. doi: 10.1016/j.ipm.2025.104239.
[43] H. Naveed, A. U. Khan, S. Qiu, et al., "A comprehensive overview of large language models," ACM Transactions on Intelligent Systems and Technology, vol. 16, Article 106, 1–72, 2025. doi: 10.1145/3744746.
[44] Y. Gan, Y. Yang, Z. Ma, P. He, et al., "Navigating the risks: A survey of security and privacy threats in LLM-based agents," ACM Transactions on Software Engineering and Methodology, Advance online publication, 2026. doi: 10.1145/3807666.
[45] A. Paracha, J. Arshad, M. B. Farah, et al., "Machine learning security and privacy: a review of threats and countermeasures," EURASIP Journal on Information Security, vol. 2024, 10, 2024. doi: 10.1186/s13635-024-00158-3.
[46] X. Zhou, S. Cao, X. Sun, and D. Lo, "Large language model for vulnerability detection and repair: Literature review and the road ahead," ACM Transactions on Software Engineering and Methodology, vol. 34, Article 145, 1–31, 2025. doi: 10.1145/3708522.
[47] Z. Sheng, Z. Chen, S. Gu, H. Huang, G. Gu, and J. Huang, "LLMs in software security: A survey of vulnerability detection techniques and insights," ACM Computing Surveys, vol. 58, Article 134, 1–35, 2025. doi: 10.1145/3769082.
[48] X. Hou, Y. Zhao, Y. Liu, Z. Yang, et al., "Large language models for software engineering: A systematic literature review," ACM Transactions on Software Engineering and Methodology, vol. 33, Article 220, 1–79, 2024. doi: 10.1145/3695988.
[49] A. Adel and N. Alani, "Can generative AI reliably synthesise literature? Exploring hallucination issues in ChatGPT," AI & Society, vol. 40, 6799–6812, 2025. doi: 10.1007/s00146-025-02406-7.

[50] S. Förster and Y. Skop, "Between fact and fairy: tracing the hallucination metaphor in AI discourse," AI & Society, vol. 41, 1685–1698, 2026. doi: 10.1007/s00146-025-02392-w.
[51] E. Bottazzi Grifoni and R. Ferrario, "The bewitching AI: The illusion of communication with large language models," Philosophy & Technology, vol. 38, 61, 2025. doi: 10.1007/s13347-025-00893-6.
[52] U. Franke, "The limits of calibration and the possibility of roles for trustworthy AI," Philosophy & Technology, vol. 37, 82, 2024. doi: 10.1007/s13347-024-00771-7.
[53] M. Dahl, V. Magesh, M. Suzgun, and D. E. Ho, "Large legal fictions: Profiling legal hallucinations in large language models," Journal of Legal Analysis, vol. 16, 64–93, 2024. doi: 10.1093/jla/laae003.
[54] T. Han, S. Nebelung, F. Khader, et al., "Medical large language models are susceptible to targeted misinformation attacks," npj Digital Medicine, vol. 7, 288, 2024. doi: 10.1038/s41746-024-01282-7.
[55] S. Johri, J. Jeong, B. A. Tran, et al., "An evaluation framework for clinical use of large language models in patient interaction tasks," Nature Medicine, vol. 31, 77–86, 2025. doi: 10.1038/s41591-024-03328-5.
[56] P. Hager, F. Jungmann, R. Holland, et al., "Evaluation and mitigation of the limitations of large language models in clinical decision-making," Nature Medicine, vol. 30, 2613–2622, 2024. doi: 10.1038/s41591-024-03097-1.
[57] J. M. Hoppe, M. K. Auer, A. Strüven, S. Massberg, and C. Stremmel, "ChatGPT with GPT-4 outperforms emergency department physicians in diagnostic accuracy: Retrospective analysis," Journal of Medical Internet Research, vol. 26, e56110, 2024. doi: 10.2196/56110.
[58] A. Shieh, B. Tran, G. He, M. Kumar, J. A. Freed, and P. Majety, "Assessing ChatGPT 4.0's test performance and clinical diagnostic accuracy on USMLE STEP 2 CK and clinical case reports," Scientific Reports, vol. 14, 9330, 2024. doi: 10.1038/s41598-024-58760-x.
[59] D. R. Hong and C. Y. Huang, "The performance of AI in medical examinations: an exploration of ChatGPT in ultrasound medical education," Frontiers in Medicine, vol. 11, 1472006, 2024. doi: 10.3389/fmed.2024.1472006.
[60] E. Asgari, N. Montaña-Brown, M. Dubois, et al., "A framework to assess clinical safety and hallucination rates of LLMs for medical text summarisation," npj Digital Medicine, vol. 8, 274, 2025. doi: 10.1038/s41746-025-01670-7.
[61] A. S. Dhane, S. Sarode, G. Sarode, and S. Singh, "A reality check on chatbot-generated references in global health research," Oral Oncology Reports, vol. 10, 100246, 2024. doi: 10.1016/j.oor.2024.100246.
[62] D. Humphreys, A. Koay, D. Desmond, et al., "AI hype as a cyber security risk: the moral responsibility of implementing generative AI in business," AI and Ethics, vol. 4, 791–804, 2024. doi: 10.1007/s43681-024-00443-4.
[63] S. Qiu, Q. Liu, S. Zhou, M. Gou, Y. Zeng, Z. Zhang, and Z. Wu, "Hard label adversarial attack with high query efficiency against NLP models," Scientific Reports, vol. 15, 9378, 2025. doi: 10.1038/s41598-025-93566-5.
[64] G. Wang, W. Wang, Y. Cao, et al., "Possibilities and challenges in the moral growth of large language models: a philosophical perspective," Ethics and Information Technology, vol. 27, 9, 2025. doi: 10.1007/s10676-024-09818-x.
[65] C. Deng, Y. Duan, X. Jin, et al., "Deconstructing the ethics of large language models from long-standing issues to new-emerging dilemmas: a survey," AI and Ethics, vol. 5, 4745–4771, 2025. doi: 10.1007/s43681-025-00797-3.
[66] A. Saxena, D. Aggarwal, N. Badathala, and P. Bhattacharyya, "A framework to synthetically generate fine-grained hallucinated data," Language Resources and Evaluation, vol. 59, 4223–4252, 2025. doi: 10.1007/s10579-025-09864-x.
[67] J. Cao, M. Li, M. Wen, and S.-C. Cheung, "A study on prompt design, advantages and limitations of ChatGPT for deep learning program repair," Automated Software Engineering, vol. 32, 30, 2025. doi: 10.1007/s10515-025-00492-x.
[68] R. Yang, Y. Ning, E. Keppo, et al., "Retrieval-augmented generation for generative artificial intelligence in health care," npj Health Systems, vol. 2, 2, 2025. doi: 10.1038/s44401-024-00004-1