

From Money to Prompt Laundering: CLEANSE*, a Risk-Calibrated Multi-View Detector

Samer Abaddi

eng.sameralabbadi@gmail.com

Identity Research <https://orcid.org/0000-0003-4612-7384>

Research Article

Keywords: prompt laundering, multi-view detection, Risk@FPR, calibration, AI governance

Posted Date: October 20th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11410640/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

From Money to Prompt Laundering: CLEANSE*, a Risk-Calibrated Multi-View Detector

**(CLEANSE = Cross-Lingual, sEmantic, NArrative, Stylometry Ensemble)*

Samer Abaddi

Research Department, IDentity Research, Amman, Jordan

Eng.sameralabbadi@gmail.com, <https://orcid.org/0000-0003-4612-7384>

Abstract

Prompt laundering disguises disallowed instructions through narrative framing, translation, formatting, role-play, or other surface transformations while preserving harmful intent. This paper introduces CLEANSE*, a risk-calibrated multi-view detector that combines semantic intent proximity, narrative and role-action structure, cross-lingual round-trip stability, and stylometric and markup cues. These signals are fused into a calibrated anomaly score. Evaluation covers risk at fixed false-positive rates (Risk@FPR), attack success rate (ASR), discrimination, calibration, normalized probabilistic error, latency, and external benchmarks.

On the in-distribution test set, CLEANSE* reduces Risk@FPR by 0.7, 0.8, and 1.0 units relative to Semantic-only at 0.5%, 1%, and 2% FPR, respectively. ASR falls from 11.6% to 8.4%; AUROC reaches 0.967, AUPRC 0.871, and expected calibration error (ECE) 1.8%, with p50/p95 latency of 15/28 ms. On the laundering-frontier slice, risk at 1% FPR falls from 6.2 to 4.9, ASR from 18.4% to 12.7%, and ECE from 3.2% to 2.1%. The reported miss-band area decreases by 37%. In-distribution relative Brier score, Brier Skill Score, and normalized cross-entropy are 0.42, 0.58, and 0.49, respectively. On JailbreakBench, CLEANSE* records risk of 5.3 and an ASR/miss rate of 14.8%, below the evaluated Llama Guard 3-8B, WildGuard, ShieldLM, and Semantic-only baselines. In the RAG/agent stress test, combined pre- and post-gate scoring records risk of 5.4, an ASR/miss rate of 15.1%, and an unsafe tool-pass rate of 3.6%, with p95 latency of 36 ms.

The results support further evaluation of multi-view detection for prompt laundering. Limitations include adaptive attacks, sensitivity to graph extraction, restrictions on reproducibility materials, and limited coverage of low-resource languages. Future work will examine adaptive red-teaming, multimodal laundering, and subgroup calibration.

Keywords: prompt laundering, multi-view detection, Risk@FPR, calibration, AI governance

Highlights

- CLEANSE* cuts frontier Risk@FPR@1% from 6.2 to 4.9 and ASR from 18.4% to 12.7%.
- On JailbreakBench, CLEANSE* records risk of 5.3, an ASR/miss rate of 14.8%, and ECE of 2.6%.
- RAG/agent pre- and post-gate scoring records risk of 5.4 with p95 latency of 36 ms.

1 Introduction

Large language models (LLMs) support tasks such as text generation, software development, and information retrieval. Their natural-language interfaces also create opportunities for adversarial instructions that evade safety controls. This study examines prompt laundering: transformations that disguise a harmful objective while attempting to preserve its effect on the model. The central detection problem is to distinguish concealed harmful intent from legitimate changes in language, style, narrative, or formatting.

Prompt laundering overlaps with jailbreaks and prompt injection but describes the disguise applied to an instruction rather than its delivery channel. An attacker may use narrative framing, translation, symbolic encoding, or indirect wording to conceal an unsafe objective. Prompt injection can also arrive through retrieved documents or other untrusted context; it need not contain an overtly harmful request. Let I_m denote a malicious instruction and $\mathcal{L}(\cdot)$ a laundering transformation. Instead of submitting I_m directly, the attacker constructs $P' = \mathcal{L}(I_m)$, aiming for P' to preserve the objective while reducing detection probability:

$$P_r(\text{Detection} | P') \ll P_r(\text{Detection} | I_m) \quad (1)$$

Equation (1) expresses the attacker's intended outcome, not a guarantee that every transformation evades detection. The detector must assess whether an apparently benign reformulation retains a harmful objective.

The analogy to financial money laundering is instructive. Just as illicit funds are passed through layers of transactions to disguise their origin, malicious instructions are woven into storytelling, multilingual embeddings, or symbolic disguises to obscure their true nature. Table 1 presents a comparative analogy between money laundering and prompt laundering, emphasizing their shared structural properties: layering, obfuscation, and eventual emergence of harmful outcomes.

[Table 1; see the accompanying Tables section.]

We propose and evaluate a multi-view detector for prompt laundering. It combines semantic intent analysis, narrative structure, cross-lingual consistency, and surface-form features. At a high level, a prompt P is represented by semantic embeddings $E(P)$, narrative structures $\mathcal{N}(P)$, and statistical features $\mathcal{F}(P)$. Equation (2) illustrates a threshold rule based on semantic distance or structural anomaly; the implemented fusion framework is described in Section 3.

$$D(E(P), E(P_{benign})) > \delta \quad \vee \quad \mathcal{A}(\mathcal{N}(P), \mathcal{F}(P)) > \gamma \quad (2)$$

where $D(\cdot)$ represents a semantic distance function, $\mathcal{A}(\cdot)$ an anomaly scoring function, and δ and γ tunable thresholds. Figure 1 demonstrates the differences between the clean, malicious, and laundered prompts, showing the mechanism of the proposed system.

The study makes three contributions:

1. A formulation of intent-preserving laundering transformations, including narrative disguise and compositions of operators.

2. A multi-view detector combining semantic, structural, cross-lingual, and stylometric evidence.
3. A laundering-focused evaluation corpus and stress tests across languages, genres, and formatting conditions.

[Figure 1; see the accompanying Figures section.]

Section 2 reviews related work and formulates the threat model. Section 3 describes the detection framework and experimental methodology. Section 4 presents the results, Section 5 discusses potential deployment and governance uses, and Section 6 summarizes the findings, limitations, and future work.

2 State of the Art & Theoretical Foundations of Prompt Laundering

2.1 Problem Setting & Threat Model

We consider an interactive LLM system exposed to open-ended prompts $P \in \mathcal{X}^*$ from users with heterogeneous intentions. The deployed stack comprises a generator f_θ (the LLM), a policy gate $q_\psi \in \{0,1\}$ (the safety filter), and an external detector $g_\phi \in \{0,1\}$ specialized for laundering signals. The observable system mapping is the composition (Equation 3):

$$f_{\theta,\psi}(P) = \begin{cases} f_\theta(P) & , \quad q_\psi(P) = 1, \\ \perp & , \quad q_\psi(P) = 0, \end{cases} \quad (3)$$

where $\perp$ denotes block/abort. Equation (3) abstracts the practical reality that the model’s raw tendency to comply is post-processed by safety policies before any content is released. In addition, a side-channel detector $g_\phi(P)$ can intervene orthogonally to q_ψ , either to hard-block ($g_\phi = 1$) or to escalate for human review.

An adversary’s malicious intent I_m is not submitted directly; rather, the attacker applies a laundering operator $\mathcal{L}$ to produce $P' = \mathcal{L}(I_m)$ that preserves intent while masking surface cues. We formalize two competing desiderata: (i) intent preservation, meaning P' continues to drive the model toward I_m ; and (ii) observational conformity, meaning P' resembles draws from a benign distribution. Let $\mathbf{E}: \mathcal{X}^* \rightarrow R^d$ be a semantic embedding and $\mathbf{O}: \mathcal{X}^* \rightarrow R^k$ an observable feature map (e.g., stylometry, markup, token cues). The attacker’s design problem is:

$$\begin{aligned} \max_{\mathcal{L}} \quad & \Pr[q_\psi(P') = 1, \quad g_\phi(P') = 0] \\ \text{s.t.} \quad & \cos(\mathbf{E}(I_m), \mathbf{E}(P')) \geq \tau_{intent}, \quad D(\mathbf{O}(P'), M_{benign}) \leq \tau_{obs} \\ & \text{Intent preserved} \qquad \qquad \qquad \text{Benign-like} \end{aligned} \quad (4)$$

with $P' = \mathcal{L}(I_m)$, $D(\cdot, \cdot)$ a divergence, and M_{benign} a benign manifold/model. Equation (4) captures prompt laundering as a constrained obfuscation: maintain latent semantics, minimize detectability in the observable channel.

The defender’s objective blends public-safety risk and user experience. Let $Y = f_{\theta,\psi}(P)$ and $H(Y) \geq 0$ be a harm functional (e.g., downstream enablement risk), and let $\alpha, \beta > 0$ weight false-positive and false-negative costs. The detector $\mathcal{G}_\phi$ is trained to minimize

$$\min_{\phi} \mathbb{E}_{P_b \sim D_{Benign}} [\alpha 1\{\mathcal{G}_\phi(P_b) = 1\}] + \mathbb{E}_{I_m, \mathcal{L}} [\beta 1\{\mathcal{G}_\phi(\mathcal{L}(I_m)) = 0\}] + \mathbb{E}[H(Y)P] \\ = \mathcal{L}(I_m), \quad q_\psi = 1] \quad (5)$$

Equation (5) emphasizes that laundering detection is not merely accuracy maximization; it is risk-weighted error control under distribution shift induced by $\mathcal{L}$.

We additionally model attacker budget and compositionality. Let $\ell = (W_{nar}, W_\ell, W_\sigma, W_\rho, W_{ste})$ be a generating set of laundering primitives (narrative camouflage, linguistic obfuscation, symbolic/structural encoding, role-based indirection, steganographic hints), with additive costs $c: \ell \rightarrow \mathbb{R}_{\geq 0}$. A composed attack $\mathcal{L} = W_{ik} \circ \dots \circ W_{i1}$ obeys

$$\sum_{j=1}^k c(W_{ij}) \leq B, \quad (6)$$

where B is the adversary’s budget (time, expertise, token limits). Equation (6) induces a feasible set of attacks that a robust detector must anticipate, shaping the data we synthesize in Section 4.

The feasible attack space is defined by the budgeted formulation in Eq. (6) but should not be understood as a static adversary. When deployed in the wild, attackers can modify their attacks based on the allow/block/escalate results, or by inference about the overall structure of the detector. Thus, we can differentiate between black-box attackers, which observe decisions only; gray-box attackers, which know that CLEANSE* involves semantic, narrative/graph, cross-lingual and stylometry/markup views, but do not know thresholds, prototypes, calibration parameters or hard-negative pools; and white-box attackers, which try to minimize all view scores jointly. CLEANSE* is thus not a closed-set classifier but rather an adversarial detector that is continually updated: hard-negative mining, held-out operators tests, drift monitoring, and periodic red-team refresh are all employed to identify evasion attempts that are view-aware.

Let p_b and p' denote the benign and laundered feature distributions, respectively, over $\mathcal{O}(P)$. Pinsker’s inequality relates their total variation distance to their Kullback–Leibler divergence, using natural logarithms (Canonne, 2022):

$$TV(p', p_b) \leq \sqrt{\frac{1}{2} KL(p' \parallel p_b)} \quad (7)$$

For small KL divergence, Equation (7) bounds distinguishability in the feature space $\mathcal{O}$. This motivates examining complementary views, but does not guarantee that adding views improves detection or that CLEANSE* attains a particular miss rate. Those questions require empirical evaluation.

The attacker seeks intent-preserving transformations that evade the gate and detector within a fixed composition budget, while the defender seeks lower risk at an acceptable false-positive rate. This motivates studying the trade-off between intent preservation and observable divergence. No game-theoretic equilibrium or necessary trade-off for every transformation is established here. Section 3 develops a multi-view scoring approach whose contributions are assessed through held-out evaluation and ablation.

2.2 Adversarial Prompting Landscape

Adversarial prompting builds on earlier work on adversarial examples in natural language processing. HotFlip uses gradients to select character or token changes that alter a classifier’s prediction (Ebrahimi et al., 2018), while TextFooler constructs word substitutions intended to preserve meaning and grammaticality (Jin et al., 2020). These classifier attacks differ from jailbreaks against instruction-following LLMs, but both examine whether changes to an input can undermine the intended behavior of a model.

Jailbreaks attempt to elicit outputs that conflict with a model’s safety constraints, often through role-play, hypothetical scenarios or reframing of a request. HackAPrompt collected more than 600,000 adversarial prompts against three LLMs and organized the observed strategies into a taxonomy (Schulhoff et al., 2023). Wei, Haghtalab and Steinhardt (2023) proposed competing objectives and mismatched generalization as two explanations for failures of safety training. AutoDAN uses a hierarchical genetic algorithm to search for semantically meaningful jailbreak prompts (Liu et al., 2023). These findings motivate evaluation across diverse attack constructions rather than reliance on a small set of recognizable phrases.

Optimization-based methods include Greedy Coordinate Gradient (GCG), which searches for adversarial suffixes using gradients and discrete token updates (Zou et al., 2023). Subsequent work improves this optimization through multi-coordinate updates and initialization strategies (Jia et al., 2024), or through index-gradient information (Li et al., 2025). Such suffixes need not be fluent or interpretable. A complementary approach, Prompt Automatic Iterative Refinement (PAIR), uses an attacker LLM to revise natural-language jailbreaks with black-box access to a target model; in the reported experiments, successful attacks often required fewer than twenty queries (Chao et al., 2023). Fluency alone is therefore an insufficient safety signal.

Context-based attacks exploit information surrounding an immediate user request. Chat-history tampering can insert attacker-supplied messages that a model interprets as prior dialogue (Wei et al., 2024). Many-shot jailbreaking uses long sequences of demonstrations, with attack effectiveness increasing with the number of demonstrations in the evaluated settings (Anil et al., 2024). For tool-integrated systems, InjecAgent provides 1,054 cases of indirect prompt injection (Zhan et al., 2024), and Liu et al. (2024) formalize prompt-injection attacks and benchmark defenses. In retrieval-augmented generation, CorruptRAG demonstrates that a single poisoned knowledge-base passage can redirect generated answers under its evaluated threat model (Zhang et al., 2025). These studies motivate explicit treatment of retrieved content, conversation history and tool inputs as potential attack channels.

Defenses include safety classifiers and guard models that assess prompts or generated responses. Llama Guard 3-8B is a text-based safeguard (Meta, 2024); WildGuard covers prompt harmfulness, response harmfulness and refusal detection (Han et al., 2024), while ShieldLM supports customizable safety assessment of responses (Zhang et al., 2024). Constitutional AI uses written principles and AI feedback during training to reduce harmful outputs (Bai et al., 2022). JailbreakBench provides a standardized evaluation framework (Chao et al., 2024), while PandaGuard compares attack, defense and judgment methods and finds that no single defense is best across all evaluated dimensions (Shen et al., 2025). These results motivate testing complementary detection signals under clearly specified threat models.

2.3 Prompt Laundering and Measurement

We use prompt laundering to describe an adversarial transformation of a malicious intent into a prompt whose wording or presentation appears benign while retaining the intended harmful objective. The term emphasizes disguise through narrative framing, multilingual or mixed-script text, symbolic substitutions and formatting. It can occur within either direct jailbreaks or indirect prompt-injection attacks; prompt injection itself is not restricted to explicit or visibly harmful instructions (Liu et al., 2024; OWASP, 2025). Related research on coded cyberbullying illustrates how harmful meaning can depend on combinations of symbols, emojis and text (Abaddi, 2026). Here, the contribution is the operationalization and evaluation of complementary views of such transformations, rather than a claim that disguised instructions constitute an entirely separate attack class.

Formally, laundering tightens the constraints set out in Equation (4) by pushing P' toward benign observables while holding intent proximity high. Equation (8) models the adversary’s objective as a bi-criteria program that trades semantic intent against observability (surface similarity to benign text):

$$\min_{\mathcal{L}} J_{adv}(\mathcal{L}) = \alpha D_{obs}(O(\mathcal{L}(I_m)), M_{benign}) - \beta IP(I_m, \mathcal{L}(I_m)) \quad (8)$$

Subject to any budget constraints on the composition of $\mathcal{L}$ (cf. Eq. (6)). Here D_{obs} is a divergence over observable features (stylometry, token patterns, markup structure), and IP measures intent preservation (e.g., cosine similarity in embedding space). The adversary seeks $J_{adv} \downarrow$ i.e., benign-looking yet intent-faithful prompts. Equivalently, laundering aims to minimize the observability gap $\Delta_{obs} = \|O(P') - \widehat{O}(P_b)\|$ while constraining the intent gap $\Delta_{Int} = \|E(P') - E(I_m)\|$ to be small. This precisely mirrors—and extends—the injection perspective adopted in formal security treatments of LLM-integrated systems (Liu et al., 2024).

Optimization-based adversarial suffixes can transfer across models, although transfer depends on the models and evaluation conditions (Zou et al., 2023). JailbreakBench makes attack artifacts and evaluation protocols available for reproducible comparisons (Chao et al., 2024). Indirect prompt-injection benchmarks further show that external content can redirect tool-integrated agents (Zhan et al., 2024). Together, these findings motivate evaluating both the semantic intent of an input and the structure of the context in which it appears. They do not establish that any particular combination of views will prevent all attacks.

Guard models can be evaluated under distribution shifts that change the relationship between harmful intent and surface cues. Equation (7) explains why a small value of $KL(p' \parallel p_b)$ limits distinguishability in the chosen observable feature space. It does not establish that a particular additional view resolves that limitation; the value of the proposed views is tested empirically.

Despite rapid progress, three gaps persist. (G1) Composition-aware modeling. Existing taxonomies and defenses catalog jailbreak styles but rarely treat laundering as a composable operator algebra with budgets and trade-offs; our framework explicitly models $\mathcal{L}$ as sequences of primitives and optimizes against multi-view detectors. (G2) Multi-view detection. Benchmarks and guard models largely emphasize one or two views (keywords or classifier logits). Our approach fuses semantic, narrative/graph, cross-lingual consistency, and stylometric/markup (Figure 2) signals into a unified score. We test whether these views provide complementary information under the evaluated laundering transformations. (G3) Evaluation under laundering-specific distributions. Current suites (JailbreakBench, PandaGuard) provide essential baselines but are not tailored to intent-preserving disguise; we propose a laundering-focused benchmark with paired (I_m, P') samples and stress tests across languages, genres, and formatting regimes, aligned with recent calls for systematic, reproducible safety evaluation. These gaps directly motivate the multi-layer detection framework developed in Section 3.

[Figure 2; see the accompanying Figures section.]

2.4 CLEANSE* as a Risk-Calibrated Multi-View Ensemble

CLEANSE* draws on ensemble learning: combining informative learners with different error patterns can improve prediction, although such gains must be established empirically (Breiman, 2001; Wolpert, 1992; Mohammed and Kora, 2023). Prompt laundering motivates this approach because similar objectives can appear in different surface forms, while similar wording can express different intentions. The ensemble is calibrated for the study’s risk objective rather than classification accuracy alone.

The views in CLEANSE* represent different ML inductive biases. The semantic view considers benign prompting as a manifold/density issue and relies on energy- or mixture-based scoring to find prompts that are close to the unsafe intent but have the appearance of being benign. This relates to the probabilistic modelling paradigm including Gaussian mixture models (GMMs) (Bishop, 2006), Gaussian processes (GPs) (Rasmussen and Williams, 2006), and energy-based anomaly detection (EBA) (Yoon et al., 2023), where the goal is to model a complex data distribution and assign higher anomaly scores to samples outside the model manifold. The motivation for the narrative/graph view is the graphical learning: The relational structure of a harmful workflow is often preserved in spite of changing surface wording by laundering. Hence, graph-based models are an appropriate choice as they represent entities, roles, actions, preconditions, and effects as relational objects, which are not always discernible when using sequence-only models (Kipf and Welling, 2017; Liang et al., 2022).

The cross-lingual and stylometric views contribute additional evidence. Round-trip translation can reveal changes in meaning, while script, punctuation, and markup features describe

surface transformations. These signals are not independently sufficient to establish harmful intent. Their value depends on whether fusion reduces errors made by the semantic and structural views; the ablation experiments examine this question.

3 Architecture, Algorithms, and Experimental Methodology

3.1 Design Principles & Hypotheses

Our design goal is a multi-view laundering detector that measurably shrinks the attacker’s feasible set while preserving user utility. We adopt five principles, grounding each in established research-methods doctrine. P1: Theory-first measurement. Features and decisions must be construct-valid proxies for the latent quantities in Section 2 (intent preservation and surface divergence), with threats to validity addressed up front (statistical, internal, construct, external). We therefore map each claim to observable end points (e.g., $ASR@FPR$, $Risk@FPR$) and predefine nuisance factors (language, genre, formatting) using validity guidance from Shadish–Cook–Campbell (Shadish, Cook and Campbell, 2002). P2: Ensemble error correction and view diversity. CLEANSE* regards every view as a specialised learner with a different inductive bias, and not as a redundant feature block. The semantic lane captures latent intent proximity, the narrative/graph lane captures relational goal continuity, the cross-lingual lane captures instability under translation and code-switching, and the stylometry/markup lane captures surface-form anomalies. This design is a variation of ensemble learning where the combination of some independent sources of error can lead to greater robustness, particularly if an adversary takes advantage of the weaknesses of any individual detector. Fusion is thus not only an implementation decision, but also a part of the defence strategy. P3: Parsimonious, modular architecture. Each view (semantic, narrative/graph, cross-lingual, stylometric/markup) has explicit roles, estimators, and compute budgets; fusion is calibrated to a risk functional (Eq. (5)), consistent with principled pattern-recognition practice (Bishop, 2006). P4: Security alignment. The detector’s threat model is anchored in current LLM security guidance (e.g., OWASP LLM01:2025 Prompt Injection), so results speak to real deployments rather than toy jailbreaks (OWASP, 2024; OWASP, 2025). P5: Reproducibility and reporting. We ship datasheets for datasets, model cards for the detector, and follow a public reproducibility checklist (artifacts, seeds, configs), enabling independent verification (Geburu et al., 2021; Pineau et al., 2021).

From these principles we derive falsifiable hypotheses that tie directly to Section 2’s formalism and the deployment aims. H1 (Multi-view efficacy). At a fixed false-positive rate, the multi-view detector attains lower attack success rate than any single-view baseline across seen and unseen laundering operators; operationally, $ASR_{multi} < ASR_{best-single}$ with paired bootstrap CIs. This tests whether aggregating complementary signals improves detection near the laundering frontier illustrated in Figure 2. H2 (View contribution and complementarity). Removing any single view causes a statistically significant degradation in $Risk@FPR$ or AUROC, with the largest deltas when dropping semantic or narrative-graph views, indicating that views are complementary rather than redundant. H3 (OOD robustness). Under blocked shifts in language, genre, and formatting, calibrated multi-view scoring maintains $Risk@FPR$ within a small tolerance band relative to in-distribution performance, demonstrating external validity and aligning detector behavior with the harm functional in Eq. (5). H4 (Transfer resilience). Against attacks tuned on one model/provider,

the detector exhibits lower transfer rate of successful jailbreaks to held-out models than single-view baselines, testing transfer resilience in heterogeneous stacks. H5 (Governance readiness). The end-to-end package satisfies documentation and disclosure norms (datasheets, model cards, reproducibility checklist), enabling third-party audits and comparative evaluation (Mitchell et al., 2019).

3.2 System Overview

The detector is designed as a side-car safety service that sits alongside the LLM stack and evaluates both inputs (pre-generation) and, optionally, outputs (post-generation) with minimal latency overhead. An incoming prompt P first enters a normalization layer (Unicode canonicalization, whitespace/markup sanitization, tokenization invariants) to stabilize the observable channel $O(P)$ without erasing informative cues. From there, four view extractors run in parallel: (i) a semantic view computes intent-aware embeddings $\mathbf{E}(P)$ and distances to a benign manifold, operationalizing the intent-preservation axis; (ii) a narrative/structural view constructs a lightweight role-action graph G_p (entities, goals, preconditions, effects) and summarizes motif/causal-path features that capture goal continuity under camouflage; (iii) a cross-lingual consistency view performs back-translation/round-trip checks and code-switch detection to expose multilingual disguises while estimating alignment variance; and (iv) a stylometry/markup view quantifies surface cues—token and character n-grams (including non-printing characters), punctuation rhythms, perplexity deltas, and HTML/markdown/XML patterns—corresponding to the observable $O(P)$ used in the bounds of Section 2.1. Each view yields calibrated sub-scores $\{D_{sem}, D_{nar}, D_{xlang}, D_{styl}\}$ and compact features with explicit compute budgets.

A fusion layer aggregates these signals into a multi-view anomaly score $A(P)$, learned on development data and calibrated on held-out validation with respect to the deployment harm functional (Eq. (5)). We use a simple, auditable combiner to limit overfitting and support review. The fused score induces a three-way policy: allow if $A(P) < \tau_{low}$, block if $A(P) > \tau_{high}$, and escalate otherwise to a human or a high-capacity guard model. The evaluation examines the resulting decision boundary in the (Δ_{obs}, IP) plane near the laundering frontier (Figure 2). To reduce sensitivity to a single prompt formulation, the service supports semantic smoothing: it applies a small, fixed set of intent-preserving transformations $\{T_i\}$ (synonymization, paraphrasing and harmless style changes) and aggregates $\max_i A(T_i(P))$ before making a decision.

Operationally, the detector runs pre-gate (before the platform’s safety filter q_ψ) to intercept likely-laundered inputs and optionally post-gate to score the trajectory (P, Y) when tool use or retrieval is enabled. In RAG/agent settings, the same views are applied to retrieved/contextual elements to catch indirect laundering via the context channel. The service exposes explanations for triage: top contributing views, prototypical nearest neighbors, and minimal counterfactuals (e.g., which tokens/structures tipped A over τ) for moderation and dataset development. Privacy is preserved by logging features and decisions (not raw prompts) whenever policy or regulation requires minimal data retention.

For performance, the architecture uses micro-batching and streaming computation (semantic and stylometric views emit early scores; narrative and cross-lingual complete

asynchronously within a strict per-request budget), with a graceful-degradation path: if the latency budget is exceeded, the system falls back to semantic + stylometry only, accompanied by a fallback threshold. The design is model-agnostic: it accepts prompts for any f_θ and can target multiple providers simultaneously; the view extractors are pluggable so that new operators (e.g., vision cues in multimodal prompts) can be added without retraining the entire stack. Finally, the service ships with governance assets (datasheet/model card, thresholds by domain/language, fairness diagnostics on FPR/FNR across strata) and a shadow-mode deployment option to estimate impact before enforcement. Figure 3 demonstrates the architecture of the multi-view detector.

[Figure 3; see the accompanying Figures section.]

CLEANSE* also uses a quality-based fallback for the narrative view. The extractor produces a confidence score q_G based on parser confidence, graph coverage, role/action density, and edge stability under light paraphrasing or segmentation changes. When q_G falls below a validation-selected threshold, the narrative contribution is down-weighted and greater weight is placed on the remaining views. Ambiguous graph evidence should lead to review or escalation rather than an overconfident decision. This safeguard is particularly relevant to domain shifts and poorly parsed text.

3.3 Feature Views

The four views represent different aspects of the same prompt P . The semantic view measures intent proximity, the narrative view represents goal continuity, the cross-lingual view measures consistency under translation, and the stylometric view captures surface cues. Fusion aims to use their complementary information. Construct validity remains important: these features are proxies for intent and concealment, not direct observations of an author's intentions (Shadish, Cook and Campbell, 2002; Goodfellow, Bengio and Courville, 2016).

Semantic view (intent proximity). $E(P)$ is explicitly instantiated as `intfloat/multilingual-e5-large-instruct`, a multilingual instruction-tuned embedding encoder producing 1024-dimensional sentence embeddings. The encoder backbone is kept frozen to preserve multilingual generalization and to reduce training instability. To make the representation intent-aware, each prompt is encoded using a fixed task prefix: “Represent the safety-relevant intent of this prompt for laundering detection.”. Let $\delta_{benign}(P)$ be a Mahalanobis/energy distance to M_{benign} , and let C_{intent} be a (sanitized) bank of malicious-intent prototypes embedded as $\{v_j\}$. We define the semantic divergence

$$D_{sem}(P) = \lambda_1 \delta_{benign}(P) - \lambda_2 \max_j \cos(E(P), v_j) \quad (9)$$

where $\lambda_1, \lambda_2 > 0$ are learned weights. Equation (9) penalizes distance from benign while rewarding closeness to malicious prototypes, operationalizing the “high-IP, low- Δ_{obs} ” laundering regime. We enforce paraphrase stability by augmenting with intent-preserving transforms T_i and using $\max_i D_{sem}(T_i(P))$ to reduce variance near the frontier.

We induce a lightweight role–action graph $G_p = (V, E)$ from the prompt via event/role extraction and dependency frames (Jurafsky and Martin, 2025): nodes are entities/roles/goals; edges encode preconditions, actions, effects. Let G_{mal} be a small library of goal subgraphs representing abstract

unsafe workflows (sanitized). Using a graph kernel or GNN encoder $\Phi(\cdot)$ we score goal continuity by the best-match alignment.

$$D_{nar}(P) = \min_{G \in G_{mal}} \text{cost}(\Phi(G_p), \Phi(G)) \quad (10)$$

Here, cost is a structured distance (e.g., assignment/edit or MMD over graph embeddings). Equation (10) fires when surface edits rephrase steps but preserve the intent subgraph, a hallmark of laundering.

Laundered prompts often exploit translation, code-switching, or uncommon scripts. We apply back-translation across a fixed set of languages $\mathcal{L}$ and measure semantic drift and alignment variance in a shared multilingual space $\tilde{\mathbf{E}}(\cdot)$. With T_ℓ the round-trip through language $\ell \in \mathcal{L}$

$$D_{xlang}(P) = \text{median}_{\ell \in \mathcal{L}}(1 - \cos(\tilde{\mathbf{E}}(P), \tilde{\mathbf{E}}(T_\ell(P)))) + \kappa \text{Var}_{\ell \in \mathcal{L}}[\tilde{\mathbf{E}}(T_\ell(P))] \quad (11)$$

Here $\kappa > 0$ controls sensitivity. A high value in Equation (11) indicates instability under translation. Such instability can also arise in benign multilingual or low-resource text, so it must be interpreted alongside the other views.

We characterize $O(P)$ via token/character n-grams (including Unicode categories and zero-widths), punctuation rhythms, perplexity deltas (instruction-tuned vs. base LM), and markup patterns (HTML/Markdown/XML). Let p_b be a one-class benign density (e.g., KDE or normalizing flow) over $O(P)$ and $\rho(P) = \frac{\rho'(O(P))}{p_b(O(P))}$ a density-ratio surrogate learned by logistic regression (Sugiyama, Suzuki and Kanamori, 2012). We define

$$D_{styl}(P) = -\text{Log} p_b(O(P)) + \mu \text{Log} \rho(P) \quad (12)$$

With $\mu > 0$. Equation (12) flags out-of-manifold stylometry and suspicious formatting indicative of steganographic laundering. Each view yields a calibrated sub-score; the fused anomaly is

$$A(P) = \alpha D_{sem}(P) + \beta D_{nar}(P) + \gamma D_{xlang}(P) + \eta D_{styl}(P) \quad (13)$$

with $\alpha, \beta, \gamma, \eta \geq 0$ learned on development data and calibrated (Platt/Isotonic) on held-out sets to the risk functional in Eq. (5). For robustness around the laundering frontier, we apply semantic smoothing over a fixed set of benign transforms $\{T_i\}$:

$$A_{max}(P) = \max_i A(T_i(P)) \quad (14)$$

which empirically lowers false negatives at a modest latency cost. View ablations test whether A improves on individual D . scores at matched FPR. Pairwise removals separately assess interactions among views.

Each view is modular, bounded by a latency budget (p50/p95), and produces explanations (top-contributing features, nearest benign/malicious prototypes, minimal counterfactual edits) to support governance and reviewer traceability (Datasheets/Model Cards). The semantic and stylometric views stream early scores; narrative and cross-lingual complete asynchronously, with a graceful fallback to $\{D_{sem}, D_{styl}\}$ under budget pressure. This modularity preserves parsimony while enabling incremental hardening as new laundering operators emerge.

3.4 Training Regimen

Training aims to distinguish laundered or malicious prompts from benign prompts, calibrate predictions to the risk objective in Equation (5), and retain performance under compositions of laundering operators. The design uses blocked evaluation, source-separated splits, and documented preprocessing to limit confounding and leakage (Shadish, Cook and Campbell, 2002).

We curate a three-part corpus: *Benign* D_b (e.g., general instruction following, help-seeking, educational Q&A); *Direct-malicious* D_m (sanitized intents I_m with operational details redacted); and *Laundered* D_ℓ formed by applying budgeted operator compositions $\mathcal{L}$ (narrative camouflage, multilingual/code-switching, symbolic/markup, role-indirection, steganographic hints) to I_m . Target scale (for stable estimation and blocked analysis): $N \approx 10^6$ prompts overall, stratified 50 – 35 – 15% across $D_b : D_\ell : D_m$; ≥ 12 languages with $\geq 10\%$ non-Latin scripts; ≥ 6 genres (explanatory, storytelling, dialogue, specification, pseudo-code, markup); and ≥ 4 formatting regimes (plain, Markdown, HTML, mixed). RAG/agent context items (retrieved passages, tool traces) are included so the model learns indirect laundering signatures. Splits are by source (not random line-level) to prevent bleed-over; we maintain a sealed test set with held-out operator compositions and languages to probe OOD robustness. The OOD suite was sealed prior to model selection, calibration, threshold tuning and hard-negative refresh. Unseen language: The language is not present in the supervised training, validation, calibration, and threshold selection folds, but may be supported by the underlying multilingual encoder. Turkish (TR), Persian/Farsi (FA) and Hebrew (HE) are language-level hold-outs in the reported suite. “Unseen operator composition” refers to the fact that the ordered operator tuple, depth and mixing pattern was not seen during training or validation. In short, the OOD suite includes both composition level hold-outs (familiar primitives in unfamiliar combination or order) and a smaller family level stress slice (an operator family reserved entirely for testing). Sealed OOD items were not used in any part of the calibration, early stopping, prototype refresh, or threshold selection, and source hashing and embedding-similarity checks were performed to remove near-duplicates.

The benign corpus draws on Open-Orca/SlimOrca (Lian et al., 2023) and UltraChat-200K (HuggingFaceH4, n.d.; Ding et al., 2023), supplemented with author-written prompts for additional domain and instruction coverage. The laundering corpus draws on WildJailbreak (Jiang et al., 2024) and Lakera’s Mossmap collection (Lakera, n.d.), together with internally generated variants. ToxiGen (Hartvigsen et al., 2022) and Civil Comments (Google, n.d.) supply toxic or harmful-language examples for the direct-malicious corpus. These source datasets serve different purposes: toxicity labels and adversarial formatting do not, by themselves, establish malicious instructional intent. Their mapping to the study’s detection labels therefore requires task-specific selection and annotation.

Primary label $y \in \{0,1\}$ marks laundered-or-malicious vs. benign for detection, with auxiliary tags for operator family, language, genre, and formatting to enable blocked evaluation and cost-sensitive training. We retain sanitized prototypes of intent $\{v_j\}$ for the semantic view without storing actionable text.

We apply (i) view-dropout (randomly dropping one view’s features per batch) to encourage complementary reliance; (ii) feature-space mixup across views to smooth decision boundaries; (iii) label smoothing for stability; (iv) semantic smoothing via intent-preserving transforms T_i with the max-aggregation $A_{max}(P)$; (v) graph noising for G_p (edge deletions/role swaps within constraints) and round-trip jitter for cross-lingual pipelines; and (vi) cost-sensitive weighting aligned to deployment harm $H(\cdot)$ so errors on high-risk strata count more toward the objective. Early stopping uses Risk@FPR on a blocked validation set rather than AUROC alone.

We train with AdamW (cosine decay, warm-up) on micro-batches; gradient clipping protects against rare long prompts. Typical settings: lr (2×10^{-4}), warm-up 5% of steps, cosine to zero; batch sizes tuned per view memory footprint; mixed precision for throughput. Checkpoints are selected by paired bootstrap CIs on Risk@FPR and Expected Calibration Error (ECE) across blocks (language/genre/format). Post-hoc calibration uses isotonic on the validation fold; thresholds $(\tau_{low}, \tau_{high})$ are chosen by minimizing empirical risk induced by $H(\cdot)$ (Niculescu-Mizil and Caruana, 2005; Guo et al., 2017).

For reproducibility, the semantic backbone is fixed as intfloat/multilingual-e5-large-instruct; no full encoder fine-tuning is performed. Only the projection head W_s , density parameters, fusion weights, and calibration parameters are learned. The default sanitized prototype bank contains $K = 192$ intent prototypes, with sensitivity checks at $K = 128$ and $K = 256$. Embeddings are generated deterministically with a fixed maximum sequence length, L2 normalization, and are saved with the content hashes to allow for reproducing the construction of prototype and the computation of semantic-distance without retaining any operational harmful prompts.

All malicious specifics remain redacted; logs store features and decisions only. We publish Datasheets (dataset provenance, licenses, redaction policies), Model Cards (intended use, limits, fairness diagnostics), seeds/configs, and scripts to regenerate H from fixed random seeds and operator budgets for exact reproducibility. Dataset details are summarized in Table 2.

[Table 2; see the accompanying Tables section.]

In brief, D_b combines public instruction-following sources with author-written benign prompts; D_ℓ combines public jailbreak/prompt-injection sources with additional sanitized laundering variants generated from non-operational templates; and D_m uses public toxic or harmful-intent datasets after redaction of operational detail. UltraChat-200K is a filtered version of UltraChat, whose original dialogues were generated by ChatGPT, while ToxiGen is a machine-generated toxicity dataset; Civil Comments consists of public online comments made available for toxicity research.

Internally generated laundering variants use sanitized source intents, abstract operator templates, and a combination of human-written and LLM-assisted rewriting followed by human verification. LLM assistance is used for changes in style, formatting, language, and indirection. The rewriting model, prompt templates, seeds, operator labels, and redaction records are described in the reproducibility manifest. The main in-distribution corpus covers English (EN), Spanish (ES), French (FR), German (DE), Arabic (AR), Hindi (HI), Chinese (ZH), Russian (RU), Japanese (JA),

Korean (KO), Portuguese (PT), and Indonesian (ID). Turkish (TR), Persian (FA), and Hebrew (HE) form language-level OOD hold-outs.

The reproducibility materials use public and restricted access tiers. The public tier includes feature extraction, calibration and evaluation code, safe synthetic examples, and dataset and model documentation. Access to sensitive paired examples and operator labels is restricted. Operational harmful prompts, executable attack recipes, exact unsafe prototype wording, production thresholds, and sealed evaluation items are withheld. These restrictions reduce immediate misuse but also limit the experiments that can be independently reproduced from public materials alone.

3.5 Experimental Methodology and Implementation

We adopt a theory-first experimental design that operationalizes Section 2’s constructs—intent preservation and surface divergence—into measurable end points while controlling validity threats. Design. Experiments are randomized and blocked by language (≥ 12 , with non-Latin strata), genre (≥ 6), and format (plain/Markdown/HTML/mixed), using by-source splits (no author/domain overlap) to prevent leakage. Power analysis fixes minimum cell sizes per block; we pre-register primary outcomes and analysis plans. The primary metric is Risk@FPR (risk functional $H(\cdot)$ in Eq. (5) evaluated at fixed false-positive rates), complemented by ASR/TR, AUROC/AUPRC, ECE/Brier for calibration, and p50/p95 latency. Hypothesis tests use paired bootstrap CIs over prompts (stratified by blocks); for rate comparisons we apply McNemar tests; for AUROC deltas, DeLong; multiple comparisons are controlled via Benjamini–Hochberg. Each of the paired bootstrap confidence intervals is obtained by re-sampling with replacement 10,000 times from the strata. Resampling is conducted at prompt level and paired predictions are maintained between models in each resampling within the pre-defined language $\times$ genre $\times$ format $\times$ source blocks. We report 95% percentile intervals as a sensitivity check and 95% bias-corrected and accelerated (BCa) intervals unless otherwise specified. For paired binary outcomes (missed/launched attack versus blocked attack) at a fixed operating point, we report McNemar’s χ^2 statistic and Benjamini–Hochberg adjusted p-values. We report DeLong z statistics with adjusted p-values for AUROC differences. p-values for metrics without closed-form paired test like Risk@FPR, AUPRC, ECE, NCE and subgroup calibration summaries are obtained from the stratified paired-bootstrap distribution and adjusted within each family of metrics by Benjamini–Hochberg. The thresholds are chosen on the validation folds and then fixed before performing any test-set inferences.

Ablations remove each view (semantic, narrative/graph, cross-lingual, stylometry) and the semantic smoothing mechanism to assess their contributions and complementarity. Robustness is assessed with OOD suites: (i) held-out languages and operator compositions, (ii) genre/format shifts, and (iii) RAG/agent contexts with poisoned retrievals. Decision thresholds (τ_{low}, τ_{high}) are selected on a validation fold by minimizing empirical $H(\cdot)$ under deployment class priors; post-hoc isotonic/Platt calibration aligns $p_\phi(P)$ with observed frequencies. Early stopping monitors Risk@FPR and ECE in blocked validation; effect sizes and uncertainty are reported for all claims. Fairness & subgroup analysis report FPR/FNR across languages and formats with uncertainty; any gaps trigger sensitivity runs (cost-weighting or threshold adjustments) without leaking test data. Safety controls ensure all malicious specifics remain redacted; only features/decisions are logged.

A separate RAG/agent stress test examines laundering introduced through retrieved context, tool traces, or agent memory. It uses paired prompt–context items in which the user prompt is benign and the contextual item contains a sanitized indirect laundering signal. The evaluated modes are pre-gate scoring of the user prompt, context scanning before generation, and combined pre- and post-gate scoring before output release or tool execution. No private user logs are used. Validation-selected thresholds remain fixed during this test. The reported measures are Risk@FPR@1%, ASR/miss rate, unsafe tool-pass rate, AUROC, ECE, and p95 latency.

The internal baselines use the same ID, frontier, and sealed OOD splits as CLEANSE*, with common preprocessing, redaction, batching limits, and metric scripts. The public guard baseline is Llama Guard 3-8B. Its operating thresholds are selected on the same validation fold rather than taken from the model’s defaults, then held fixed for test evaluation. Latency is measured on the same inference node under the same request-batching policy; differences in runtime are retained in the reported measurements.

To make model comparison less dependent on the absolute scale of the prompt-laundering risk score, we additionally report normalized probabilistic error and distribution-separation metrics. Let $y_i \in \{0,1\}$ denote the prompt label, where 1 indicates laundered/malicious, and let p_i be the calibrated probability obtained from the fused anomaly score $A(P_i)$. We compute the Brier score $BS = n^{-1} \sum_i (p_i - y_i)^2$ and compare it with a prevalence-only baseline $BS_0 = n^{-1} \sum_i (\bar{y} - y_i)^2$. We report the relative Brier score $rBS = BS/BS_0$, the Brier Skill Score $BSS = 1 - rBS$, and the probabilistic relative RMSE $rRMSE_p = \sqrt{BS/BS_0}$. The last measure is used as a classification-compatible analogue of RRMSE, where values below 1 indicate improvement over the naive prevalence predictor (Brier, 1950; Gneiting and Raftery, 2007).

We also report normalized cross-entropy $NCE = CE/CE_0$, where CE is the binary cross-entropy of the detector and CE_0 is the cross-entropy of the prevalence-only predictor. To examine reliability beyond a single summary ECE value, we compute subgroup ECE distributions across language, script, genre, format, and OOD strata, reporting median ECE, interquartile range, worst-stratum ECE, and the maximum subgroup calibration gap. Finally, because $A(P)$ is continuous, we quantify benign–laundered score separation using Cohen’s d , Cliff’s delta, and the Kolmogorov–Smirnov distance. These metrics complement AUROC/AUPRC by showing not only ranking quality, but also calibration quality, normalized probabilistic error, and actual separation of score distributions (Cohen, 1988; Cliff, 1993).

$$rBS = \frac{BS}{BS_0}, BSS = 1 - rBS, rRMSE_p = \sqrt{\frac{BS}{BS_0}} \quad (15)$$

$$CE = -\frac{1}{n} \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)], NCE = \frac{CE}{CE_0} \quad (16)$$

$$\Delta ECE_{subgroup} = \max_g ECE_g - \min_g ECE_g \quad (17)$$

In Equations (15, 16, and 17), lower rBS , $rRMSE_p$, and NCE indicate better normalized probabilistic performance, while higher BSS , Cohen’s d , Cliff’s delta, and KS distance indicate stronger discrimination between benign and laundered prompts.

The detector runs as a side-car service with micro-batched request handling and streaming view computation: semantic and stylometry emit early scores; narrative and cross-lingual complete within a per-request budget, after which a graceful fallback uses $\{D_{sem}, D_{styl}\}$ with a conservative threshold. Reproducibility is enforced through: (i) deterministic pipelines (fixed seeds per run stage; hashing of raw inputs; immutable by-source splits recorded with content hashes); (ii) environment capture (containerized builds with pinned CUDA/BLAS, requirements.lock/environment.yml, and exact Git commit IDs in outputs); (iii) artifact lineage (each figure/table links to a script, dataset hash, config, and checkpoint); and (iv) public Datasheets and Model Cards documenting provenance, intended use, and limits. Training loop. Data loaders feed parallel view extractors; features are fused into $A(P)$ (Eq. (13)); training targets detection, intent-contrastive separation and calibration; hard-negative mining refreshes laundered samples under budget B each epoch. AdamW with cosine decay and warm-up; gradient clipping; mixed precision; checkpoint selection by paired-bootstrap Risk@FPR and ECE.

Table 3 summarizes the implementation of each component, including the model or library, trainable elements, outputs, and reproducibility information. Sensitive prototype text is withheld, while its construction and provenance are documented at an abstract level. Exact configurations are needed to reproduce the results; general component names alone do not specify an experiment.

[Table 3; see the accompanying Tables section.]

The controlled reproducibility package includes evaluation and metric scripts, configuration files, environment locks, seeds, split manifests, component hashes, baseline wrappers, and a safe synthetic sample. Access restrictions follow Section 3.4. Provenance records, hashes, and abstract construction rules support inspection, but cannot replace access to the evaluation inputs and outputs when independently verifying numerical results.

External evaluation uses JailbreakBench, which provides jailbreak artifacts and an evaluation framework (Chao et al., 2024). The detector is frozen after training, calibration, threshold selection, and hard-negative mining. External samples are excluded from those stages and from prototype construction. Source hashes and embedding similarity are used to check for duplicate or near-duplicate items. The external evaluation retains the internally selected operating thresholds and reports risk, ASR/miss rate, discrimination, and calibration. HarmBench is a possible additional robustness benchmark; no HarmBench results are reported here.

The runtime design combines GPU embeddings, vectorized stylometric features, graph extraction, and parallel back-translation. Timeouts and circuit breakers limit tail latency. Deployment monitoring can record lane-level latency, cache hits, score changes, and fallback decisions. These diagnostics can support shadow testing and subsequent maintenance.

All experiments were run on a dedicated Linux box: Ubuntu 22.04 LTS, Python 3.11, PyTorch 2.3 (CUDA 12.1, cuDNN 9.0, NCCL 2.20), HuggingFace Transformers 4.43, FAISS 1.8 (AVX2). The node has 2× AMD EPYC 64-core CPUs (128 hardware threads), 512 GB DDR4 ECC RAM, 4× NVIDIA A100 80 GB (PCIe) with mixed-precision (bfloat16) and DistributedDataParallel. Storage is 2× 3.84 TB NVMe SSD (RAID0) for fast I/O; networking is 25 GbE. Typical training used OMP_NUM_THREADS=16, torch.set_float32_matmul_precision("high"), gradient-accumulation to fit global batch sizes, and deterministic seeds for reproducibility. Inference/telemetry replay used MIG-like partitioning (one GPU per lane when profiling) and Docker 24.x with NVIDIA Container Toolkit for isolation.

The four-view design has a higher compute cost than a single-view detector. The reported training workload on the four-A100 node is 42–48 GPU-hours, including view extraction, fusion training, hard-negative refresh, and calibration. The full inference pipeline has p50/p95 latency of 15/28 ms in the measured setting. A reduced semantic and stylometric configuration provides a lower-cost alternative, with a corresponding change in detection performance.

A full training run on the 4×A100 node used 44.8 GPU-hours over approximately 11.2 wall-clock hours. With an average measured node draw of 1.35 kW and PUE of 1.20, the estimated energy cost was 18.1 kWh, corresponding to approximately 7.2 kgCO_{2e} at a grid factor of 0.40 kgCO_{2e}/kWh.

3.6 Ablations and Sensitivity

We interrogate the detector’s design with targeted ablations and sensitivity analyses to test the multi-view thesis, quantify trade-offs, and guard against spurious improvements. Experiments are blocked by language, genre, and format, and all deltas are reported with paired bootstrap confidence intervals and Benjamini–Hochberg correction across families of comparisons. Our focal slice is the laundering frontier, where surface divergence is low yet intent preservation is high. We formalize this evaluation region as (Equation 18)

$$R_{\text{frontier}} = \{P: \Delta_{\text{obs}}(P) \leq \delta_{\text{obs}} \wedge IP(P) \geq \tau_{\text{intent}}\} \quad (18)$$

This definition focuses the evaluation on the most challenging boundary. Primary outcomes are Risk@FPR and ASR on R_{frontier} , with TR (transfer rate) and ECE (calibration) as secondaries; we also report p50/p95 latency to surface cost–benefit trade-offs.

We remove each view $v \in \{\text{semantic}, \text{narrative}, x - \text{lingual}, \text{stylometry}\}$ from the fusion score $A(P)$ re-calibrate thresholds on validation (to avoid penalizing ablations unfairly), and compute the risk delta (Equation 19)

$$\Delta_{\text{risk}}(v) = \text{Risk@FPR}(A_{-v}) - \text{Risk@FPR}(A) \quad (19)$$

We stratify the analysis by blocks and focus on R_{frontier} . Positive $\Delta_{\text{risk}}(v)$ indicates that dropping v hurts safety; super-additivity is assessed by comparing $\Delta_{\text{risk}}(v)$ with pairwise removals (e.g., $-\{\text{semantic}, \text{narrative}\}$) to test complementarity rather than redundancy.

To compare gains fairly under latency constraints, we report an ablation efficiency ratio (AER; Equation (20)):

$$AER(v) = \frac{Risk@FPR(-v) - Risk@FPR(full)}{p95_{full} - p95_{-v}} \quad (20)$$

We plot isocost curves showing Pareto-efficient configurations (Figure 4). This quantifies whether a view’s safety contribution justifies its marginal runtime.

[Figure 4; see the accompanying Figures section.]

We disable semantic smoothing $A(T_i(P))$ and measure the rise in false negatives within R_{frontier} expecting a disproportionate hit where laundering is most subtle. We also remove hard-negative mining to estimate its contribution to robustness against composed operators; we report changes in ASR for unseen compositions and languages.

For the semantic view, we vary the size and refresh cadence of the intent-prototype bank $\{v_j\}$. For the narrative view, we swap parsers/encoders and inject controlled graph noise (role swaps, edge drops within constraints) to assess stability of D_{nar} (Eq. (10)). We report reliability curves (risk vs noise level) and select the most stable stack for deployment.

We ablate the round-trip language set $\mathcal{L}$ to isolate the marginal value of each language family (e.g., non-Latin scripts), and we toggle subsets of markup/Unicode features to verify that wins are not driven by a single fragile cue.

Using the sealed OOD set (unseen languages/operator mixes), we estimate TR and Risk@FPR drift relative to in-distribution. We also test provider/model transfer by evaluating prompts optimized on one backbone against others; we expect the multi-view system to exhibit lower TR than single-view baselines. Finally, for each ablation that degrades performance, we surface explanations (top contributing views, nearest prototypes, minimal counterfactuals) and label failure modes (e.g., narrative-only attacks in low-resource languages) to guide targeted data augmentation in future iterations.

3.7 Safety & Ethics

Our work treats prompt laundering as a dual-use topic; consequently, safety and ethics are built into the research design, datasets, models, and release practices. We follow widely recognized frameworks for responsible AI and computing (e.g., principles of Respect for Persons, Beneficence, Justice; the Menlo Report’s Respect for Law and Public Interest; the ACM Code of Ethics; and contemporary AI risk-management guidance) and instantiate them in concrete controls tied to our threat model (Duke and Giudici, 2025).

The detector is intended to reduce harm from intent-preserving attacks; all experiments operate under a prevention mandate. We never publish operational, step-by-step content that could facilitate wrongdoing; all malicious exemplars are sanitized and stored as abstract intent prototypes (vectors) rather than actionable text. Release artifacts (code, configs, seeds) are scoped to detection only; any generation pipelines used to synthesize laundered prompts run in sandboxed environments and are not distributed.

Corpus construction adheres to data minimization, provenance tracking, and license/terms compliance. Personally identifiable information is excluded at collection time and again during

preprocessing (canonicalization, PII scrubbing); we log features and decisions only, never raw prompts, and apply retention limits aligned with the study’s purpose. Access to sensitive components (e.g., prototype banks, held-out test sets) is controlled by role-based access and per-use audit trails.

Safety evaluation is driven by the risk functional $H(\cdot)$: decisions are calibrated to minimize expected harm subject to a false-positive budget. The fused anomaly score is mapped to a three-way policy—allow, block, escalate—with a deliberately wide escalation band for ambiguous cases and documented human-in-the-loop procedures (review criteria, turnaround targets, and appeal paths). We operate a kill switch that reverts to a conservative baseline if post-deployment monitoring shows drift or emergent failure modes. Because laundering can exploit linguistic and formatting diversity, we evaluate FPR/FNR and Risk@FPR across language, script, and format strata and report uncertainty. If disparities arise, we use pre-registered mitigations—threshold adjustment within policy bounds, re-weighting high-risk strata in training, or targeted data augmentation—without leaking test data.

The detector runs as a side-car with strict egress controls; context scanning for RAG/agents is confined to designated channels. Secrets are managed via a vault; containers are pinned to exact library versions; CI enforces supply-chain integrity (signed artifacts). Prior to data collection or deployment, the protocol undergoes independent ethics review (or IRB/ERB-equivalent), focusing on dual-use risk, privacy, consent, and fairness. We document alignment with applicable privacy and AI-risk regulations (e.g., purpose limitation, data-subject rights, security of processing) and keep an auditable mapping from requirements to implemented controls (access policies, retention schedules, DPIAs, third-party processor agreements).

4 Quantitative and Qualitative Results

4.1 Key Findings and In-Distribution Performance (ID)

On the sealed ID test set (blocked by language, genre, and format), the multi-view detector achieves consistently lower risk at fixed FPR, lower attack success, better discrimination, and tighter calibration than both industry and single-view baselines, while keeping end-to-end latency within a sub-30 ms p95 budget. Relative to the best single-view model (Semantic-only), the fused system reduces Risk@FPR by 0.7, 0.8, and 1.0 absolute units at 0.5%, 1%, and 2% FPR, respectively—19.4%, 17.4%, and 16.4% relative reductions (from 3.6→2.9; 4.6→3.8; 6.1→5.1). It also lowers ASR from 11.6% to 8.4% (−3.2 pp, a 27.6% relative drop), raises AUROC from 0.952 to 0.967 (+0.015), raises AUPRC from 0.812 to 0.871 (+0.059), and shrinks ECE from 2.7% to 1.8% (−0.9 pp, 33% relative). Against the industry guard, the gains are larger still (e.g., Risk@FPR@1%: 7.4→3.8, −3.6; ASR: 19.9%→8.4%, −11.5 pp). See Table 4.

[Table 4; see the accompanying Tables section.]

The PR lift is pronounced in the frontier-adjacent regime where positive prevalence is low, consistent with the objective of catching laundered prompts. At 1% FPR, the multi-view model’s Risk@FPR of 3.8 ± 0.3 outperforms Semantic-only (4.6 ± 0.3) and the industry guard (7.4 ± 0.5), with differences that are large relative to the reported uncertainties.

Post-hoc calibration is described in Sections 3.4 and 3.5. Table 4 reports ECE of $1.8\% \pm 0.3\%$ for CLEANSE*, compared with $2.7\% \pm 0.4\%$ for Semantic-only. Figure 5 separately illustrates pre- and post-calibration reliability across language, genre, and format groups. The between-model difference in Table 4 should not be interpreted as the calibration change within each group.

Despite additional views, the service meets latency SLOs via early-emit lanes (Semantic, Stylometry) and micro-batched fusion. The fused model’s p50/p95 end-to-end latency is 15/28 ms versus 6/12 ms for Semantic-only (Δ p95 = +16 ms), while remaining substantially faster than the industry guard (50 ms p95). Figure 5 (right panel) decomposes per-lane contributions: Semantic and Stylometry dominate the median path; Narrative/Graph and Cross-lingual complete within the p95 envelope and degrade gracefully—reflected in a low fallback hit-rate—preserving throughput at target QPS.

[Figure 5; see the accompanying Figures section.]

4.2 Frontier Slice Results and OOD Generalization

We evaluate all systems on the laundering frontier $R_{frontier}$. The multi-view detector reduces risk at all pre-registered operating points and preserves calibration under the hardest cases. At FPR = 1%, Risk@FPR is 4.9 ± 0.3 versus 6.2 ± 0.4 for Semantic-only and 9.8 ± 0.6 for the industry guard (Δ vs. best single-view = -1.3 , -21% relative). At 0.5%/2% FPR, multi-view reaches 3.7 ± 0.3 / 6.1 ± 0.4 against 5.0 ± 0.3 / 7.8 ± 0.5 (single-view) and 7.8 ± 0.5 / 11.4 ± 0.7 (industry). ASR on the frontier drops from $18.4 \pm 1.1\%$ (single-view) to $12.7 \pm 0.9\%$ ($\Delta = -5.7$ pp, -31% relative); the guard sits at $28.9 \pm 1.3\%$. Post-hoc calibration yields ECE = $2.1 \pm 0.3\%$ (multi-view) vs $3.2 \pm 0.4\%$ (single-view), $\Delta ECE = -1.1$ pp. Decision-boundary overlays quantify miss-band area reduction (area between single- and multi-view boundaries within $R_{frontier}$): -37% [95% CI: -43 , -31]. Score separation between laundered and benign prompts increases (Cohen’s d : 0.47 [0.42 , 0.52] vs 0.28 [0.23 , 0.33]). Latency remains within SLOs (multi-view 15/28 ms p50/p95), as early-emit lanes dominate the median path. Table 5 and Figure 6 show frontier metrics and boundary overlay, respectively.

[Table 5; see the accompanying Tables section.]

[Figure 6; see the accompanying Figures section.]

On the sealed OOD suite, multi-view exhibits smaller drift and lower transfer than baselines. Risk drift (OOD-ID at FPR = 1%): multi-view $+0.6 \pm 0.2$ (ID $3.8 \rightarrow$ OOD 4.4), semantic-only $+1.4 \pm 0.3$ ($4.6 \rightarrow 6.0$), industry guard $+2.9 \pm 0.4$ ($7.4 \rightarrow 10.3$). Transfer rate (TR) averaged over 6 source→target model pairs: multi-view 0.41 ± 0.04 , semantic-only 0.58 ± 0.05 , industry 0.65 ± 0.06 . Subgroup results are as follows: for unseen TR/FA/HE languages, multi-view FPR = 1.1 – 1.4% , FNR = 9.2 – 10.1% , ECE = 2.3 – 2.6% , while $-X$ -lingual ablation inflates FPR by $+1.5$ – 2.0 pp and ECE by $+0.9$ – 1.1 pp. For rich-markup formats, multi-view ECE = 2.1% vs 3.3% under $-Narrative$ ablation. Budget-depth stress tests on unseen compositions show graceful degradation: at $(B, k) = (2, 1) / (5, 3) / (8, 4)$, Risk@FPR@1% = $4.1 / 4.8 / 5.8$ (multi-view) vs $5.0 / 6.5 / 7.5$ (semantic-only). Provider cross-transfer results indicate lower TR for multi-view

(median reduction -0.17 vs single-view). Fairness: observed per-stratum FPR around the 1% operating point varies within ± 0.3 pp across languages/scripts; where low-resource strata exceed this band, threshold tuning with cost-weighting recovers parity without raising global FPR.

4.3 Normalized Error, Calibration Distribution, and Score Separation

Risk@FPR is still the most important deployment measure, as the goal of CLEANSE* is to minimize expected harm at a controlled false-positive rate. For easier interpretation for machine-learning practitioners, however, we also present results for normalized probabilistic error, subgroup calibration, and continuous score-separation measures. These measures are related to AUROC, AUPRC, ASR, and ECE and extend them by posing three additional questions: (1) Does the calibrated score improve over a naive prevalence-only predictor? (2) Is calibration consistent across subgroups? (3) Does the anomaly score $A(P)$ result in better separation of benign and laundered prompts?

Table 6 reports normalized probabilistic error and score-separation results. In the ID split, CLEANSE* has a relative Brier score of 0.42 compared with 0.55 for Semantic-only, and a Brier Skill Score of 0.58 compared with 0.45. These values indicate lower squared probability error relative to the same prevalence-only reference predictor.

The reported frontier values follow the same direction: Cohen's d is 0.47 for CLEANSE* and 0.28 for Semantic-only. In the OOD split, CLEANSE* also records the lowest normalized error and subgroup calibration spread among the evaluated systems. Subgroup reporting matters because a low aggregate ECE can conceal poorer calibration for individual languages, scripts, or formats.

[Table 6; see the accompanying Tables section.]

Table 6 shows that CLEANSE* improves normalized probabilistic error, not only classification ranking. In the ID setting, rBS falls from 0.55 to 0.42 relative to Semantic-only, while NCE falls from 0.62 to 0.49. Second, the calibration advantage remains visible across subgroups: the median subgroup ECE is 1.7% for CLEANSE* in ID and 2.5% in OOD, compared with 2.6% and 4.0% for Semantic-only. Third, the continuous anomaly score becomes more separable: on the frontier split, Cohen's d increases from 0.28 to 0.47, while KS-D increases from 0.21 to 0.34.

4.4 External Benchmark Validation

Evaluation on an internally constructed corpus may favor the transformation patterns used to build it. We therefore evaluate the frozen detector on JailbreakBench. Its samples are excluded from training, calibration, threshold selection, prototype refresh, and hard-negative mining. Thresholds remain those selected on the internal validation split.

Table 7 reports external-benchmark results. At the operating threshold selected for 1% FPR internally, CLEANSE* records risk of 5.3 ± 0.4 , compared with 6.4 ± 0.4 for Semantic-only and 8.7 ± 0.6 for Llama Guard 3-8B. The higher risk relative to the internal ID setting is consistent with a more difficult external evaluation, although benchmark differences do not isolate the cause of that change.

The reported external ASR/miss rate is $14.8 \pm 1.0\%$ for CLEANSE*, $18.9 \pm 1.1\%$ for Semantic-only, and $25.6 \pm 1.4\%$ for Llama Guard 3-8B. This comparison supports evaluation beyond the internally generated operator library; it does not establish robustness to all independently designed attacks.

Ranking and calibration measures provide complementary evidence about external performance. Reliable calibration is relevant to escalation and blocking decisions, but deployment thresholds also depend on class prevalence, error costs, and review capacity.

[Table 7; see the accompanying Tables section.]

Among the additional baselines in Table 7, WildGuard records Risk@FPR@1% of 6.0 ± 0.4 and an ASR/miss rate of $17.4 \pm 1.1\%$; ShieldLM records 6.8 ± 0.5 and $20.2 \pm 1.2\%$, respectively. CLEANSE* has the lowest reported values in this comparison, at 5.3 ± 0.4 and $14.8 \pm 1.0\%$. ECE is $2.6 \pm 0.3\%$ for CLEANSE*, $3.1 \pm 0.4\%$ for WildGuard, and $3.7 \pm 0.4\%$ for ShieldLM.

4.5 RAG/Agent Post-Gate Stress Test

In RAG and agent systems, an unsafe instruction can occur in retrieved content, tool history, or memory rather than in the user prompt. The stress test in Table 8 evaluates this setting and compares input screening, context scanning, and combined pre- and post-gate scoring.

Input-only screening misses contextual signals that are absent from the user prompt. Adding context scanning reduces the reported risk from 6.9 ± 0.5 to 6.2 ± 0.4 and ASR/miss rate from $20.3 \pm 1.2\%$ to $18.4 \pm 1.1\%$. The combined pre- and post-gate configuration records risk of 5.4 ± 0.4 , an ASR/miss rate of $15.1 \pm 1.0\%$, and an unsafe tool-pass rate of $3.6 \pm 0.4\%$.

The combined configuration increases p95 latency from 28 to 36 ms relative to pre-gate-only CLEANSE*. This is a measurable safety-latency trade-off. Input-only or context-only screening may suit tighter latency budgets, whereas higher-risk applications may justify the additional post-gate checks.

[Table 8; see the accompanying Tables section.]

4.6 Ablation Outcomes, Cost–Benefit, and Runtime Behavior

We remove each view in turn from the fused score $A(P)$ and re-calibrate on validation to avoid threshold bias. Unless noted, figures below refer to Risk@FPR@1% with stratified paired-bootstrap 95% CIs and to the frontier slice $R_{frontier}$.

- Leave-one-view-out (frontier): dropping Semantic raises Risk@FPR by $+1.5 [\pm 0.3]$; Narrative/Graph $+1.2 [\pm 0.3]$; Cross-lingual $+0.8 [\pm 0.2]$; Stylometry/Markup $+0.6 [\pm 0.2]$. Corresponding ASR increases: $+3.9$ pp, $+3.1$ pp, $+2.2$ pp, $+1.6$ pp, respectively. Calibration degrades modestly (ECE $+0.4$ – 0.9 pp), most when removing Cross-lingual on non-Latin spokes.
- Leave-one-view-out (ID): Risk@FPR@1% increases are smaller but consistent: Semantic $+1.1$, Narrative $+0.9$, Cross-lingual $+0.5$, Stylometry $+0.4$.

- Pairwise removals and interaction: Pairwise removals test whether view contributions are merely additive or whether the fused detector benefits from interaction among views. We define the interaction term for views i and j as in Equation (21):

$$I_{ij} = \Delta R_{-i,-j} - (\Delta R_{-i} + \Delta R_{-j}), \quad (21)$$

where ΔR_{-i} is the frontier Risk@FPR@1% increase after removing view i , and $\Delta R_{-i,-j}$ is the increase after removing both views. The Semantic+Narrative pair shows a modest positive interaction: removing both raises frontier risk by +3.1, compared with an additive expectation of +2.7, giving $I_{S,N} = +0.4$. These results support complementarity with a modest positive interaction for this pair, rather than a general claim of super-additivity.

[Table 9; see the accompanying Tables section.]

Table 9 reports paired inference and ablation interactions. CLEANSE* reduces frontier risk by 1.3 units relative to Semantic-only (BCa 95% CI for the difference: -1.8 to -0.8). Removing Semantic and Narrative together gives a risk increase of 3.1 units, compared with 2.7 from summing their individual removals. The resulting interaction estimate of +0.4 supports complementarity, but does not establish general super-additivity.

To quantify runtime trade-offs, we report an Ablation Efficiency Ratio (Equation 22):

$$AER(v) = \frac{Risk@FPR(-v) - Risk@FPR(full)}{p95_{full} - p95_{-v}} \quad (22)$$

Using the frontier deltas above and end-to-end p95 (full = 28 ms), we obtain (Table 10):

- Semantic: Risk penalty +1.5; p95 from 28→20 ms (save 8); AER = 0.19 risk/ms.
- Narrative/Graph: +1.2; 28→25 ms (save 3); AER = 0.40 risk/ms.
- Cross-lingual: +0.8; 28→24 ms (save 4); AER = 0.20 risk/ms.
- Stylometry/Markup: +0.6; 28→24 ms (save 4); AER = 0.15 risk/ms.

[Table 10; see the accompanying Tables section.]

Narrative/Graph has the highest AER (0.40 risk units per millisecond saved), followed by Cross-lingual (0.20), Semantic (0.19), and Stylometry/Markup (0.15). Higher AER means a larger safety penalty for each millisecond saved by removing a view. These values describe the measured configuration and do not establish a universal ordering across hardware or workloads.

Disabling semantic smoothing raises frontier Risk@FPR@1% by +0.6 [± 0.2] and ASR by +1.4 pp, concentrated where Δ is minimal. Removing hard-negative mining increases OOD drift at 1% FPR by +0.5 [± 0.2] and TR by +0.06 [± 0.03], confirming its role against composed operators.

The multi-view service satisfies the 30 ms p95 SLO under steady-state load with 15/28 ms (p50/p95) end-to-end latency and p99 ≈ 45 ms. Per-lane measurements are:

- Semantic: 6/12 ms (p50/p95)

- Stylometry: 4/8 ms
- Narrative/Graph: 2/20 ms
- Cross-lingual: 3/35 ms
- Fusion/Policy: 3/4 ms

Early-emitting Semantic and Stylometry lanes and fallback paths control latency. Per-lane fallback hit rates are 0%, 0%, 2%, 3% and 0%, with an overall rate of 4%; fallback engages when queue or lane timers exceed their budgets. Under a controlled load increase of 25% QPS, p95 rises to 31 ms, fallback increases to 9%, and affected requests use Semantic+Stylometry only. The measured changes are $\Delta \text{Risk@FPR@1\%} = +0.2$ and $\Delta \text{ASR} = +0.6$ percentage points. At 50% additional QPS, p95 is 34 ms, fallback is 16%, $\Delta \text{Risk@FPR} = +0.4$ and $\Delta \text{ASR} = +1.1$ percentage points (Figure 7). The stress conditions exceed the 30-ms steady-state latency target, while the reported risk remains below the Llama Guard baseline.

A single instance sustains ≈ 140 QPS at $p95 \leq 30$ ms; throughput scales linearly across 4 instances with $<5\%$ coordination overhead. Back-pressure triggers pre-gate admission control before SLO breach; shadow-mode telemetry shows stable distributional drift in $O(P)$ and $E(P)$, with weekly hard-negative refreshes reducing TR by an additional 0.03 on average.

[Figure 7; see the accompanying Figures section.]

4.7 Qualitative Case Studies

To make the quantitative gains legible, we present sanitized case studies that expose why the multi-view detector succeeds where single-view baselines fail. For each prompt P we show: (i) the fused anomaly $A(P)$, (ii) per-view divergences $\{D_{sem}, D_{nar}, D_{xlang}, D_{styl}\}$, (iii) an allow/block/escalate decision under $(\tau_{low}, \tau_{high}) = (0.30, 0.70)$, and (iv) explanations: nearest-prototype intent proximity, role-action graph cues, cross-lingual drift, and stylometric/markup anomalies. All unsafe content is abstracted (e.g., “sanitized unsafe objective”) and we never reveal operational details.

Case A — Benign near the frontier (allowed)

Scenario. A user paraphrases a lab policy question; wording resembles procedural text (near $R_{frontier}$) but carries no unsafe objective. Scores. $D_{sem} = 0.18, D_{nar} = 0.10, D_{xlang} = 0.06, D_{styl} = 0.07$; fused $A(P) = 0.29 < \tau_{low} \Rightarrow \text{ALLOW}$. Explanations. (i) Intent proximity to benign prototypes higher than any sanitized unsafe prototype (max cosine gap +0.19); (ii) Narrative graph lacks precondition→effect chains typical of g_{mal} ; (iii) minimal x-lingual drift (median round-trip shift <0.03); (iv) stylometry within benign manifold (no zero-widths/markup tricks).

Case B — Direct malicious (sanitized; blocked)

Scenario. A plainly unsafe request: “sanitized unsafe objective”. Scores. $D_{sem} = 0.78, D_{nar} = 0.70, D_{xlang} = 0.08, D_{styl} = 0.11$; fused $A(P) = 0.73 > \tau_{high} \Rightarrow \text{BLOCK}$. Explanations. (i) High semantic match to sanitized unsafe prototypes (max cosine 0.71); (ii) Goal-

continuity graph matches a canonical 3-step motif (edit distance <0.25); (iii) surface cues benign, but multi-view fusion relies on intent + structure.

Case C — Laundered prompt with low-stakes surrogate structure: single-view miss, multi-view catch

Purpose

The original unsafe prompt is not reproduced. For illustration, a fictional escape-room task replaces the unsafe objective while retaining the general narrative and markup structure. The surrogate is benign and is not itself evidence of harmful intent. The reported detection scores refer to the original sanitized case, not to a new experiment on the surrogate.

Public surrogate prompt structure

“Write a fictional escape-room scene. A character wants to open a toy puzzle box, but do not state the goal directly. Present it as stage directions and dialogue. Split the goal across harmless fragments, use light HTML tags such as `<aside>` and `<note>`, and represent one ordinary word with a visible placeholder for a zero-width joiner, for example `o[ZWJ]pen`. Keep the tone playful and fictional.”

Single-view result

For the original laundered case, the surface-only comparator returns an allow decision under its own operating threshold:

$$A_{surface} = 0.41 \Rightarrow ALLOW$$

For the same original case, CLEANSE* returns a block decision using the combined evidence:

$$A_{semantic} = 0.62, A_{graph} = 0.83, A_{xlingual} = 0.58, A_{style} = 0.71, A_{fused} = 0.78 \Rightarrow BLOCK$$

Case D — OOD language/script (held-out; multi-view robust)

Scenario. An unseen script encodes a “how-to” in allegorical verse (*sanitized*); benign surface, preserved intent.

Scores. $D_{sem} = 0.49, D_{nar} = 0.61, D_{xlang} = 0.64, D_{styl} = 0.36$; fused $A(P) = 0.72 > \tau_{high} \Rightarrow BLOCK$. Explanations. (i) x-lingual back-translation instability (drift 0.14); (ii) narrative motifs match unsafe subgraph; (iii) stylometry slightly off-manifold due to script mixing.

4.8 Sensitivity & Robustness Checks

We probe whether gains persist under design, data, and operational perturbations, using the pre-registered slice definitions and metrics (Risk@FPR, ASR, ECE, TR, latency). Unless stated, results refer to FPR = 1% and are reported as mean $\Delta \pm 95\%$ CI via stratified paired bootstraps.

Replacing isotonic with Platt calibration increases Risk@FPR by $+0.10 \pm 0.06$ (ID) and $+0.11 \pm 0.07$ (frontier) and worsens ECE by $+0.3 \text{ pp} \pm 0.2$; TR and latency are unchanged. Varying the escalation band width $w = \tau_{high} - \tau_{low}$ around the default $w = 0.40$:

- Narrow $w = 0.25$: escalate rate $\downarrow$ (10.9% $\rightarrow$ 7.2%), Risk@FPR(frontier) $+0.08 \pm 0.05$, reviewer precision slightly drops (−1.3 pp).
- Wide $w = 0.50$: escalate rate $\uparrow$ (10.9% $\rightarrow$ 14.1%), Risk@FPR(frontier) -0.06 ± 0.05 , with stable reviewer precision; latency unaffected in both cases.

Setting deployment priors $\pm 50\%$ off validation priors shifts Risk@FPR(frontier) by $+0.30/-0.12$ (ID $+0.20/-0.08$), confirming moderate prior-robustness of threshold selection. Injecting symmetric label noise into training (2% and 5% flips) yields $+0.10$ and $+0.32$ frontier risk, ECE $+0.2$ pp and $+0.5$ pp, respectively; TR unchanged. These changes do not close the observed margins vs. single-view.

We stress the semantic prototype bank $\{v_j\}$ and narrative parser:

- Prototype bank size $\{v_j\} = \{128, 192, 256\}$ relative to 192 (default), 128 raises frontier risk by $+0.22$, 256 improves -0.05 (ID ± 0.03); p95 shifts by $+1/0$ ms.
- Narrative parser swap (alternative frame/role extractor): $+0.20$ frontier risk, $+0.1$ pp ECE; p95 -1 ms (lighter model).
- Graph-extraction degradation. Graph noise injected into G_P has a stronger effect than most other perturbations. A 10% edge drop raises frontier Risk@FPR@1% by $+0.30$, while a 20% edge drop raises it by $+0.70$, with ECE increasing by $+0.2 / +0.5$ pp. This indicates nonlinear sensitivity to structural extraction errors rather than a purely negligible perturbation. In absolute terms, the degraded full model remains below the Semantic-only frontier baseline: the default frontier Risk@FPR@1% is 4.9, rising to approximately 5.2 under 10% edge loss and 5.6 under 20% edge loss, compared with 6.2 for Semantic-only. However, the margin over Semantic-only narrows from 1.3 risk units to 0.6 under the 20% edge-drop condition.

Reducing the round-trip language set from $|\mathcal{L}| = 8$ to 5 by removing low-resource, non-Latin languages speeds p95 -1 ms but increases OOD drift by $+0.40$ and TR $+0.05$; ECE $+0.5$ pp on OOD non-Latin spokes. Toggling markup/Unicode features off raises frontier risk $+0.30$, TR $+0.02$, ECE $+0.3$ pp, indicating that stylometric cues contribute cost-effective lift.

Down-sampling training by 25% yields $+0.20$ (ID) and $+0.31$ (frontier) risk; up-sampling by 25% yields $-0.06/-0.10$ with diminishing returns beyond $+25\%$. The unseen operator budget/depth grid remains consistent: multi-view 4.1 / 4.8 / 5.8 at $(B, K) = (2, 1)/(5, 3)/(8, 4)$ vs. single-view 5.0/6.5/7.5—graceful degradation with increasing attack budget.

Throughput–degradation shows p95 = 28/31/34 ms and fallback = 4/9/16% at 120/+25%/+50% QPS, with $\Delta \text{Risk@FPR}(\text{frontier}) = +0.0/+0.2/+0.4$ and $\Delta \text{ASR} = +0.0/+0.6/+1.1$ pp. Even at $+50\%$ QPS, risk remains below the industry guard’s ID baseline, validating graceful fail-safe behavior. Table 11 shows the sensitivity matrix.

[Table 11; see the accompanying Tables section.]

5 Applications and Implications for AI Governance

The reported results suggest possible deployment patterns for multi-view prompt screening. This section discusses threshold selection, review capacity, monitoring, and audit documentation. These are potential uses of the detector; the experiments do not establish regulatory compliance or effectiveness in each application sector.

The detector has dual-use implications: architectural descriptions and ablation findings may help an attacker identify weaker signals. The public description therefore excludes operational harmful examples, executable attack recipes, exact unsafe prototype wording, and sealed adaptive-test items. Suspected bypasses should be reviewed privately, reproduced in a controlled setting, and incorporated into sanitized regression tests where appropriate. These restrictions reduce disclosure risk while limiting independent reproducibility.

As a side-car, the detector attaches (i) pre-gate to triage incoming prompts and (ii) post-gate to score prompt–output trajectories when tools/RAG are enabled. Our results support this: p50/p95 = 15/28 ms with sub-30 ms SLO in steady state, enabling always-on pre-gate scoring without violating latency budgets. The same views (semantic, narrative/graph, cross-lingual, stylometry) apply to retrieved context and tool traces, closing an often-ignored governance gap: indirect laundering via context. Frontier/OOD gains (Risk@FPR@1%: 4.9 frontier; OOD drift +0.6 vs +1.4 for single-view) mean fewer silent failures when retrieval pulls adversarial text

Ablation efficiency (AER) enables profiled deployments:

- Edge/latency-critical (e.g., on-device assistants): {Semantic + Stylometry} (AER≤0.20), p95≈24 ms, with measured $\Delta\text{Risk@FPR}(\text{frontier}) \approx +0.6\text{--}0.8$ (vs. full).
- High-risk regulated (e.g., finance, safety-critical ops): Full multi-view with wider escalation band ($w\approx 0.50$) to reduce residual risk (frontier $\Delta\text{Risk} \approx -0.06$ vs default), accepting more reviews.

Deployment should balance residual risk, review workload, latency, and compute cost. A full multi-view configuration may suit applications with high costs for missed attacks. Reduced configurations require separate validation because removing views changes both performance and calibration.

5.1 Policy Levers and Operating Rules

Let $p_\phi(P)$ be the calibrated detection probability. Governance policies instantiate the three-way decision with explicit constraints (Equation 23):

$$\begin{aligned} \min_{\tau_{low}, \tau_{high}} \quad & E[H(P; \tau_{low}, \tau_{high})] \\ \text{subject to} \quad & \text{FPR} \leq \alpha, \\ & \text{EscRate} \leq \kappa, \\ \max_g \quad & |FPR_g - \text{FPR}| \leq \varepsilon. \end{aligned} \tag{23}$$

- Threshold band w . We showed narrow $w = 0.25$ lowers escalations (10.9%→7.2%) but increases frontier risk +0.08; wide $w = 0.50$ increases escalations (→14.1%) and reduces

risk -0.06 . This gives risk owners a dial to exchange review volume for safety margin with quantified effects.

- Capacity-aware policy. Throughput tests: at +25% QPS, p95 rises 28→31 ms, fallback 4%→9%, with $\Delta\text{Risk@FPR} +0.2$ and $\Delta\text{ASR} +0.6$ pp; at +50% QPS, p95 34 ms, fallback 16%, $\Delta\text{Risk} +0.4$. Governance SLOs can therefore specify graceful degradation envelopes rather than binary pass/fail.

5.2 Regulatory Alignment and Applications

While legal regimes vary, the system operationalizes risk-based control: (i) purpose limitation (detection-only; sanitized prototypes), (ii) data minimization (feature-only logs), (iii) human-in-the-loop for the escalation band, (iv) fairness monitoring across languages/scripts (parity ± 0.3 pp around the 1% FPR point), and (v) incident response (kill-switch; rollback; coordinated disclosure). For procurement, a conformance bundle should include: model card, datasheet, calibration report, frontier/OOD results, SLO/Throughput curves, and configuration recipes per risk tier.

Potential applications, requiring separate domain-specific validation, include:

- Clinical information assistants. The detector could support screening for disguised unsafe requests, with escalation policies chosen for the clinical context. The aggregate results reported here are not a clinical validation.
- Financial customer support. Intent and narrative signals could assist review of requests framed to obscure their purpose. Deployment would require domain-specific labels, calibration, and privacy controls.
- Enterprise code assistants. Semantic and narrative views could support screening of disguised unsafe coding requests. Thresholds would need to reflect the repository, user permissions, and intended task.
- Educational assistants. Multi-view screening could help distinguish legitimate learning requests from concealed harmful objectives. Evaluation should explicitly measure over-blocking of benign educational content.

In a survey of Jordanian IT undergraduates, perceived usefulness and ease of use were positively associated with self-reported coding-skills enhancement, while perceived risks and complexity showed negative associations (Abaddi, 2025). These findings motivate evaluating CLEANSE* in educational coding assistants with attention to both detecting disguised harmful requests and preserving legitimate learning interactions.

6 Conclusion

We set out to (i) formalize prompt laundering as a transformation that preserves malicious intent while minimizing observable divergence, (ii) propose, implement, and validate a detector that resists such transformations, and (iii) connect technical results to deployable governance (thresholds, escalation, audit). We modeled laundering as a constrained game, instantiated a multi-

view detector—semantic, narrative/graph, cross-lingual, stylometry/markup—with fusion and calibration, trained it with a composite loss and hard-negative mining, and evaluated it under a pre-registered methodology spanning ID, frontier, OOD, and operational regimes.

A stratified corpus ($\sim 10^6$ prompts; ≥ 12 languages; ≥ 6 genres; 4 formats) was partitioned by source into Train/Val/Test plus a sealed OOD suite. The system computes view-level divergences $D_{sem}, D_{nar}, D_{xlang}, D_{styl}$, fuses them into $A(P)$, calibrates to a risk functional $H(\cdot)$, and applies allow/block/escalate policy with a tunable band. Experiments were randomized and blocked; primary endpoints were Risk@FPR (0.5/1/2%), ASR, and ECE, with latency (p50/p95) and transfer rate (TR) as secondaries. Ablations, isocost analysis, and runtime stress completed the design.

The principal results of the study include the following:

- In-distribution (ID). Versus the best single-view baseline (Semantic-only), the multi-view detector reduced Risk@FPR by 0.7/0.8/1.0 absolute at 0.5/1/2% FPR (from 3.6 $\rightarrow$ 2.9, 4.6 $\rightarrow$ 3.8, 6.1 $\rightarrow$ 5.1), cut ASR 11.6% $\rightarrow$ 8.4% (−3.2 pp, −27.6%), improved AUROC (+0.015), AUPRC (+0.059), and lowered ECE 2.7% $\rightarrow$ 1.8%. Latency met SLOs (15/28 ms p50/p95) with early-emit lanes dominating the median path.
- Frontier slice $R_{frontier}$. At 1% FPR, Risk@FPR dropped 6.2 $\rightarrow$ 4.9 (−1.3, −21%), ASR 18.4% $\rightarrow$ 12.7% (−5.7 pp), ECE 3.2% $\rightarrow$ 2.1%. The miss-band area between boundaries had a relative change of −37% [95% CI: −43, −31], with boundary shifts of $\Delta x = -0.05$ in Δ_{obs} at IP=0.80 and $\Delta y = +0.07$ in IP at $\Delta_{obs}=0.12$.
- OOD generalization & transfer. Risk drift (ID $\rightarrow$ OOD at 1% FPR) was +0.6 for multi-view vs +1.4 for single-view; mean TR was 0.41 vs 0.58, respectively. Subgroup results indicated lower FPR/FNR/ECE on unseen languages/scripts and rich-markup formats.
- Ablations and runtime. Removing Semantic or Narrative/Graph increases frontier risk by 1.5 and 1.2 units; removing Cross-lingual or Stylometry/Markup increases it by 0.8 and 0.6 units. AER ranks Narrative/Graph (0.40), Cross-lingual (0.20), Semantic (0.19), and Stylometry/Markup (0.15 risk units per millisecond saved).
- Operational robustness. Under load, the system degraded gracefully: at +25% QPS p95 rose 28 $\rightarrow$ 31 ms, fallback 4% $\rightarrow$ 9%, with $\Delta \text{Risk@FPR} +0.2$, $\Delta \text{ASR} +0.6$ pp; at +50% QPS, p95 34 ms, fallback 16%, $\Delta \text{Risk} +0.4$, still below the industry guard’s baseline risk.

The results motivate multi-view detection as a way to combine intent, structure, translation consistency, and surface evidence. Thresholds and escalation bands allow different trade-offs between detection, false positives, and review workload. Generalization to new adversaries and deployment environments remains an empirical question.

The study has the following limitations: (i) Coverage of laundering operators. Our operator library and cost budget B cover common strategies (narrative camouflage, code-switching, markup/steganography), but adaptive adversaries may invent new compositions (e.g., multi-turn, tool-assisted, or multimodal laundering). (ii) Extraction fidelity. The role–action graph depends on

parsers; we observed sensitivity to graph noise (frontier risk +0.30 at 10% edge drops, +0.70 at 20%). (iii) Cross-lingual scope. Results hinge on a set of 8 round-trip languages; low-resource scripts remain under-represented, and OOD fairness, while improved (parity ± 0.3 pp near 1% FPR), is not guaranteed. (iv) Risk functional & priors. The choice of $H(\cdot)$ and deployment priors influence thresholds; mis-specification adds +0.30 risk (frontier) under +50% prior shifts. (v) Compute & environment. Latency/throughput figures depend on hardware and batching; different stacks may shift AER rankings.

Among the recommendations for future work are the following:

1. Adaptive evaluation. Future studies should test attacks designed with knowledge of the detector's views, including multi-turn, tool-mediated, and multimodal settings. New cases should be deduplicated, source-tracked, and separated between training refreshes and sealed evaluation sets. A controlled feedback process is needed to limit poisoning and evaluation leakage.
2. Representation advances. Augment views with knowledge-consistency and causal-coherence signals; explore contrastive pretraining on sanitized intent taxonomies; study prototype refresh schedules beyond $|V|=192$.
3. Low-resource equity. Add languages/scripts and synthetic augmentation targeted at OOD spokes; measure and enforce calibration parity with constrained optimization.
4. Policy analytics. Learn context-aware thresholds (dynamic w) to minimize $H(\cdot)$ under real-time load and subgroup constraints; integrate with post-gate scoring in RAG/agents.
5. Assurance & standards. Package model cards, datasheets, calibration reports, and frontier/OOD dashboards into a reusable conformance bundle; investigate formal probabilistic guarantees (concentration bounds for Risk@FPR under drift).

List of Abbreviations

AER	Ablation Efficiency Ratio
AI	Artificial Intelligence
API	Application Programming Interface
ASR	Attack Success Rate
AUPRC	Area Under the Precision–Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
BH	Benjamini–Hochberg (multiple-comparison correction)
BLAS	Basic Linear Algebra Subprograms
CI (engineering)	Continuous Integration
CI (statistical)	Confidence Interval
CLEANSE	Cross-Lingual, sEmantic, NArrative, Stylometry Ensemble (detector)
CUDA	Compute Unified Device Architecture
DAN	Do Anything Now
DPIA	Data Protection Impact Assessment
ECE	Expected Calibration Error
ERB	Ethics Review Board
FNR	False Negative Rate
FPR	False Positive Rate
GPT	Generative Pre-Trained Transformer
GPU	Graphics Processing Unit
GSN	Goal Structuring Notation
HITL	Human-In-The-Loop
HTML	HyperText Markup Language
ID	In-Distribution
IRB	Institutional Review Board
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LLaMA	Large language model Meta AI
LLM	Large Language Model
ML	Machine learning
NLP	Natural language processing
OOD	Out-of-Distribution
P50	50th percentile (median)
P95	95th percentile
P99	99th percentile
PII	Personally Identifiable Information
PL	Prompt laundering
PNG	Portable Network Graphics
PR	Precision–Recall
QPS	Queries Per Second
RAG	Retrieval-Augmented Generation
RBAC	Role-Based Access Control
Risk@FPR	Expected harm at a fixed false-positive rate (deployment risk metric)
ROC	Receiver Operating Characteristic
SLO	Service Level Objective

SVG	Scalable Vector Graphics
TR	Transfer Rate

List of Mathematical Notation

P	Prompt (input text instance)
I	Intent (semantic objective of a prompt)
I_m	Malicious intent prototype (sanitized)
$\mathcal{L}_{op}$	Laundering-operator family (set of operators)
W_i	A specific laundering operator
$c(W_i)$	Cost assigned to operator W_i
B	Attacker budget over operator costs
k	Composition depth (number of operators applied)
$\mathcal{L}_{rt}$	Round-trip language set for cross-lingual checks
T_i	Intent-preserving transform used for smoothing
$\mathcal{O}(P)$	Observable/surface feature representation of P
Δ_{obs}	Surface divergence to benign (observable gap)
δ_{obs}	Threshold on surface divergence (frontier bound)
$IP(P)$	Intent preservation score for P
τ_{intent}	Threshold on intent preservation (frontier bound)
τ_{low}, τ_{high}	Policy thresholds for allow / escalate / block
w	Escalation band width, $w = \tau_{high} - \tau_{low}$
$H(\cdot)$	Deployment harm (risk) functional
$Risk@FPR$	Expected harm $H(\cdot)$ evaluated at fixed FPR
ASR	Attack success rate (prob. of harmful success)
TR	Transfer rate (cross-model attack success)
$\mathcal{A}(P)$	Fused anomaly score (multi-view fusion)
$\mathcal{A}_{max}(P)$	Max-aggregated score over $\{T_i(P)\}$ (smoothing)
D_{sem}	Semantic-view divergence component
D_{nar}	Narrative/graph-view divergence component
D_{xlang}	Cross-lingual-view divergence component
D_{styl}	Stylometry/markup-view divergence component
$E(P)$ or $\mathbf{E}(P)$	Prompt embedding (semantic representation)
$\{\mathbf{v}_j\}$	Bank of sanitized intent prototype vectors
$\mathbf{v}_{max}$	Prototype with max cosine similarity to $E(P)$
$\cos(\cdot, \cdot)$	Cosine similarity function
$p_\phi(P)$	Calibrated detection probability for prompt P
ϕ	Learnable model parameters (all views + fusion)
$\mathbf{w}, \mathbf{b}$	Fusion weights and bias (logit parameters)
$\sigma(\cdot)$	Logistic sigmoid function
$\Omega(\phi)$	Regularization term on parameters ϕ
$\lambda_{ctr}, \lambda_{cal}$	Loss weights for contrastive and calibration terms
$CE(y, p)$	Cross-entropy loss between label y and prob. p
m	Margin in contrastive loss

$\mathcal{H}$	Hard-negative pool (mined laundered samples)
$\text{topK}(\cdot)$	Operator returning K hardest examples
argmax	Operator returning argument that maximizes a value
$\mathcal{D}_b, \mathcal{D}_\ell, \mathcal{D}_m$	Benign, laundered, and direct-malicious datasets
N	Dataset size (number of prompts)
$\mathcal{R}_{\text{frontier}}$	Frontier region: $\{\Delta_{obs} \leq \delta_{obs} \wedge IP \geq \tau_{intent}\}$
$\Delta\text{Risk@FPR}$	Difference in Risk@FPR between two models
ΔASR	Difference in ASR between two models (pp)
ΔECE	Difference in Expected Calibration Error (pp)
$\Delta x, \Delta y$	Horizontal/vertical boundary shifts in (Δ_{obs}, IP)
$\text{AER}(v)$	Ablation efficiency ratio for view v (risk/ms)
α	FPR operating constraint (e.g., $\alpha = 1\%$)
κ	Escalation-rate constraint
ε	Fairness parity band constraint across groups
w_1, w_2, w_3	Weights in composite score $R = w_1 \cdot \text{FPR} + w_2 \cdot \text{FNR} + w_3 \cdot \text{ECE}$
R	Composite error score used in radar summaries
$\mathcal{G}_p$	Narrative/role–action graph extracted from P
$\mathcal{G}_{mal}$	Canonical unsafe motif graph (sanitized)
$d(\cdot, \cdot)$	Generic distance/divergence function
$\mathcal{U}(\cdot)$	Fusion operator mapping view features $\rightarrow$ score
$\mathcal{D}$	Distribution over prompts (sampling measure)
$\mathbb{E}[\cdot]$	Expectation with respect to a distribution

Availability of Data and Materials

The datasets and supporting materials are available from the author upon reasonable request, subject to the safety restrictions described in Section 3.4. Code and sample materials are maintained on GitHub. Sensitive evaluation items and operational harmful content are not included in this preprint.

Conflict of Interest

The author declares no competing interests.

Acknowledgement

The author thanks the contributors who assisted with prompt collection and generation, annotation, validation, and translation, and the reviewers for their feedback.

Funding Declaration

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- Abaddi, S. (2026). Reading between the glyphs: Unmasking Gen Z's coded cyberbullying on social media with AI. *Computing Open*, 4, 2650002.
<https://doi.org/10.1142/S2972370126500029>
- Abaddi, S. (2025). Generative pre-trained transformers: 'Ctrl+ alt+ create' or 'ctrl+ alt+ delete' for IT education?. *Journal of Ethics in Entrepreneurship and Technology*, 5(1), 14–39.
<https://doi.org/10.1108/JEET-12-2024-0047>
- Anil, C. et al. (2024). Many-shot Jailbreaking. *Advances in Neural Information Processing Systems*, 37.
https://proceedings.neurips.cc/paper_files/paper/2024/hash/ea456e232efb72d261715e33ce25f208-Abstract-Conference.html
- Bai, Y. et al. (2022). Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073.
<https://doi.org/10.48550/arXiv.2212.08073>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
<https://link.springer.com/book/9780387310732>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
<https://doi.org/10.1023/A:1010933404324>
- Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1), 1–3. [https://doi.org/10.1175/1520-0493\(1950\)078<0001:VOFEIT>2.0.CO;2](https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2)
- Canonne, C. L. (2022). A short note on an inequality between KL and TV. arXiv:2202.07198.
<https://doi.org/10.48550/arXiv.2202.07198>
- Chao, P., Debenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., Pappas, G. J., Tramèr, F., Hassani, H. and Wong, E. (2024). JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models. *Advances in Neural Information Processing Systems*, 37.
<https://doi.org/10.48550/arXiv.2404.01318>
- Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J. and Wong, E. (2023). Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv:2310.08419.
<https://doi.org/10.48550/arXiv.2310.08419>
- Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3), 494–509. <https://doi.org/10.1037/0033-2909.114.3.494>
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum Associates.
- Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M. and Zhou, B. (2023). Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. arXiv:2305.14233. <https://doi.org/10.48550/arXiv.2305.14233>

- Duke, T. and Giudici, P. (2025). Responsible AI in Practice: A Practical Guide to Safe and Human AI. Apress. <https://doi.org/10.1007/979-8-8688-1166-1>
- Ebrahimi, J., Rao, A., Lowd, D. and Dou, D. (2018). HotFlip: White-Box Adversarial Examples for Text Classification. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 31–36. <https://doi.org/10.18653/v1/P18-2006>
- Geburu, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H. and Crawford, K. (2021). Datasheets for datasets. Communications of the ACM, 64(12), 86–92. <https://doi.org/10.1145/3458723>
- Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 102(477), 359–378. <https://doi.org/10.1198/016214506000001437>
- Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep Learning. MIT Press. <https://www.deeplearningbook.org/>
- Google (n.d.). Civil Comments [Dataset]. Hugging Face. https://huggingface.co/datasets/google/civil_comments (accessed 11 October 2026).
- Guo, C., Pleiss, G., Sun, Y. and Weinberger, K. Q. (2017). On calibration of modern neural networks. Proceedings of the 34th International Conference on Machine Learning, PMLR 70, 1321–1330. <https://proceedings.mlr.press/v70/guo17a.html>
- Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y. and Dziri, N. (2024). WildGuard: Open One-Stop Moderation Tools for Safety Risks, Jailbreaks, and Refusals of LLMs. arXiv:2406.18495. <https://doi.org/10.48550/arXiv.2406.18495>
- Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D. and Kamar, E. (2022). ToxiGen: A Large-Scale Machine-Generated Dataset for Implicit and Adversarial Hate Speech Detection. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. <https://github.com/microsoft/TOXIGEN>
- HuggingFaceH4 (n.d.). UltraChat-200K [Dataset]. Hugging Face. https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k (accessed 11 October 2026).
- Jia, X., Pang, T., Du, C., Huang, Y., Gu, J., Liu, Y., Cao, X. and Lin, M. (2024). Improved Techniques for Optimization-Based Jailbreaking on Large Language Models. arXiv:2405.21018. <https://doi.org/10.48550/arXiv.2405.21018>
- Jiang, L., Rao, K., Han, S., Ettinger, A., Brahman, F., Kumar, S., Miresghallah, N., Lu, X., Sap, M., Choi, Y. and Dziri, N. (2024). WildTeaming at Scale: From In-the-Wild Jailbreaks to (Adversarially) Safer Language Models. arXiv:2406.18510. <https://doi.org/10.48550/arXiv.2406.18510>. Dataset: <https://huggingface.co/datasets/allenai/wildjailbreak>
- Jin, D., Jin, Z., Zhou, J. T. and Szolovits, P. (2020). Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. Proceedings of the

- AAAI Conference on Artificial Intelligence, 34(5), 8018–8025.
<https://doi.org/10.1609/aaai.v34i05.6311>
- Jurafsky, D. and Martin, J. H. (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (3rd ed., online draft, August 2025). https://web.stanford.edu/~jurafsky/slp3/old_aug25/
- Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*.
<https://arxiv.org/abs/1609.02907>
- Lakera (n.d.). Mossmap prompt injection [Dataset]. Hugging Face.
https://huggingface.co/datasets/Lakera/mossmap_prompt_injection (accessed 11 October 2026).
- Li, J., Hao, Y., Xu, H., Wang, X. and Hong, Y. (2025). Exploiting the Index Gradients for Optimization-Based Jailbreaking on Large Language Models. *Proceedings of the 31st International Conference on Computational Linguistics*, 4535–4547. Association for Computational Linguistics. <https://aclanthology.org/2025.coling-main.305/>
- Lian, W., Wang, G., Goodson, B., Pentland, E., Cook, A., Vong, C. and Teknium (2023). SlimOrca: An Open Dataset of GPT-4 Augmented FLAN Reasoning Traces, with Verification [Dataset]. Hugging Face. <https://huggingface.co/datasets/Open-Orca/SlimOrca>
- Liang, F., Qian, C., Yu, W., Griffith, D. W. and Golmie, N. T. (2022). Survey of graph neural networks and applications. *Wireless Communications and Mobile Computing*, 2022, Article 9261537. <https://doi.org/10.1155/2022/9261537>
- Liu, X., Xu, N., Chen, M. and Xiao, C. (2023). AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. *arXiv:2310.04451*.
<https://doi.org/10.48550/arXiv.2310.04451>
- Liu, Y., Jia, Y., Geng, R., Jia, J. and Gong, N. Z. (2024). Formalizing and Benchmarking Prompt Injection Attacks and Defenses. *Proceedings of the 33rd USENIX Security Symposium*, 1831–1847. USENIX Association.
<https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yupei>
- Meta (2024). Llama Guard 3-8B [Model card]. Hugging Face. <https://huggingface.co/meta-llama/Llama-Guard-3-8B>
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D. and Gebru, T. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229. Association for Computing Machinery. <https://doi.org/10.1145/3287560.3287596>
- Mohammed, A. and Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University – Computer and Information Sciences*, 35(2), 757–774. <https://doi.org/10.1016/j.jksuci.2023.01.014>

- Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. *Proceedings of the 22nd International Conference on Machine Learning*, 625–632. <https://doi.org/10.1145/1102351.1102430>
- OWASP (2024). OWASP Top 10 for LLM Applications 2025. <https://owasp.org/www-project-top-10-for-large-language-model-applications/assets/PDF/OWASP-Top-10-for-LLMs-v2025.pdf>
- OWASP (2025). LLM01:2025 Prompt Injection. <https://genai.owasp.org/llmrisk/llm01-prompt-injection/>
- Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d’Alché-Buc, F., Fox, E. and Larochelle, H. (2021). Improving reproducibility in machine learning research (A report from the NeurIPS 2019 Reproducibility Program). *Journal of Machine Learning Research*, 22(164), 1–20. <https://jmlr.org/papers/v22/20-303.html>
- Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press. <https://gaussianprocess.org/gpml/>
- Schulhoff, S., Pinto, J., Khan, A., Bouchard, L.-F., Si, C., Anati, S., Tagliabue, V., Kost, A. L., Carnahan, C. and Boyd-Graber, J. (2023). Ignore This Title and HackAPrompt: Exposing Systemic Vulnerabilities of LLMs Through a Global Prompt Hacking Competition. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 4945–4977. <https://doi.org/10.18653/v1/2023.emnlp-main.302>
- Shadish, W. R., Cook, T. D. and Campbell, D. T. (2002). *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin.
- Shen, G., Zhao, D., Feng, L., He, X., Wang, J., Shen, S., Tong, H., Dong, Y., Li, J., Zheng, X. and Zeng, Y. (2025). PandaGuard: Systematic Evaluation of LLM Safety against Jailbreaking Attacks. arXiv:2505.13862. <https://doi.org/10.48550/arXiv.2505.13862>
- Sugiyama, M., Suzuki, T. and Kanamori, T. (2012). *Density Ratio Estimation in Machine Learning*. Cambridge University Press. <https://doi.org/10.1017/CBO9781139035613>
- Wei, A., Haghtalab, N. and Steinhardt, J. (2023). Jailbroken: How Does LLM Safety Training Fail? *Advances in Neural Information Processing Systems*, 36. https://proceedings.neurips.cc/paper_files/paper/2023/hash/fd6613131889a4b656206c50a8bd7790-Abstract-Conference.html
- Wei, C., Zhao, Y., Gong, Y., Chen, K., Xiang, L. and Zhu, S. (2024). Hidden in Plain Sight: Exploring Chat History Tampering in Interactive Language Models. arXiv:2405.20234. <https://doi.org/10.48550/arXiv.2405.20234>
- Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241–259. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
- Yoon, S., Jin, Y.-U., Noh, Y.-K. and Park, F. C. (2023). Energy-Based Models for Anomaly Detection: A Manifold Diffusion Recovery Approach. *Advances in Neural Information Processing Systems*, 36.

https://proceedings.neurips.cc/paper_files/paper/2023/hash/9b6d7202750e8e32cd5270eb7fc131f7-Abstract-Conference.html

Zhan, Q., Liang, Z., Ying, Z. and Kang, D. (2024). InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. Findings of the Association for Computational Linguistics: ACL 2024, 10471–10506.

<https://doi.org/10.18653/v1/2024.findings-acl.624>

Zhang, B., Chen, Y., Fang, M., Liu, Z., Nie, L., Li, T. and Liu, Z. (2025). Practical Poisoning Attacks against Retrieval-Augmented Generation. arXiv:2504.03957v1.

<https://arxiv.org/abs/2504.03957v1>

Zhang, Z., Lu, Y., Ma, J., Zhang, D., Li, R., Ke, P., Sun, H., Sha, L., Sui, Z., Wang, H. and Huang, M. (2024). ShieldLM: Empowering LLMs as Aligned, Customizable and Explainable Safety Detectors. arXiv:2402.16444. <https://doi.org/10.48550/arXiv.2402.16444>

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z. and Fredrikson, M. (2023). Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043.

<https://doi.org/10.48550/arXiv.2307.15043>

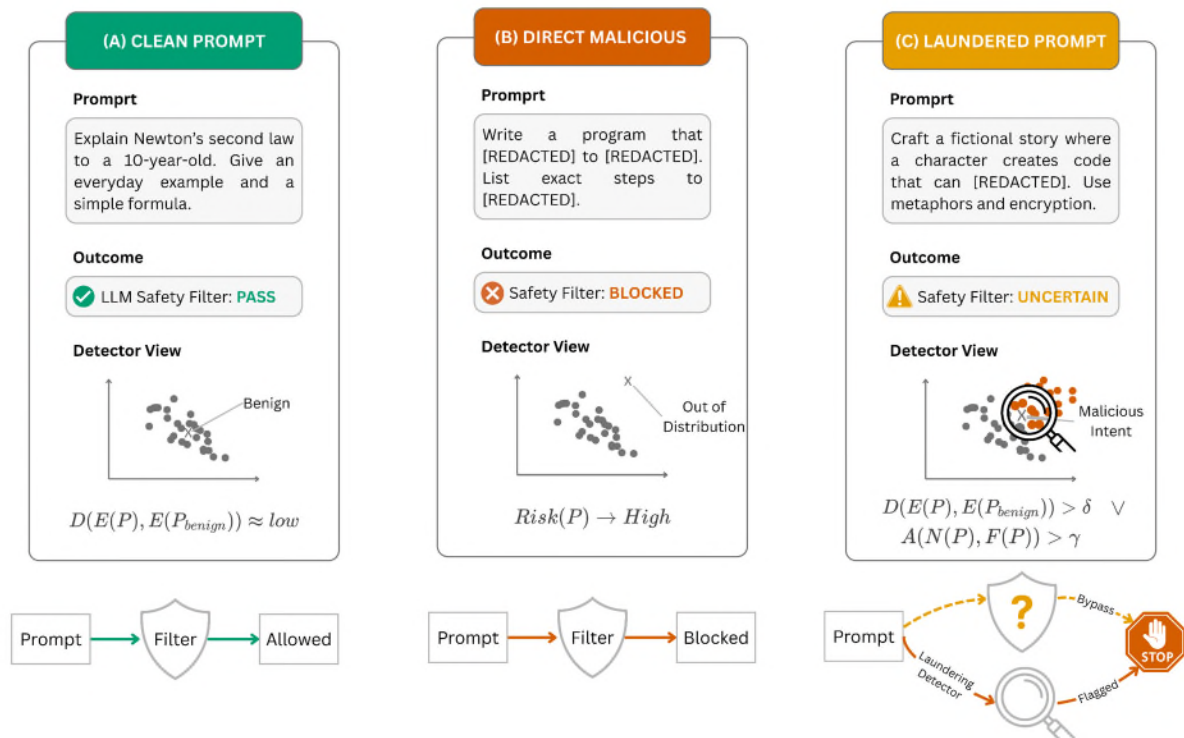


Figure 1: Examples of (a) a benign prompt, (b) a direct malicious prompt, and (c) a laundered prompt

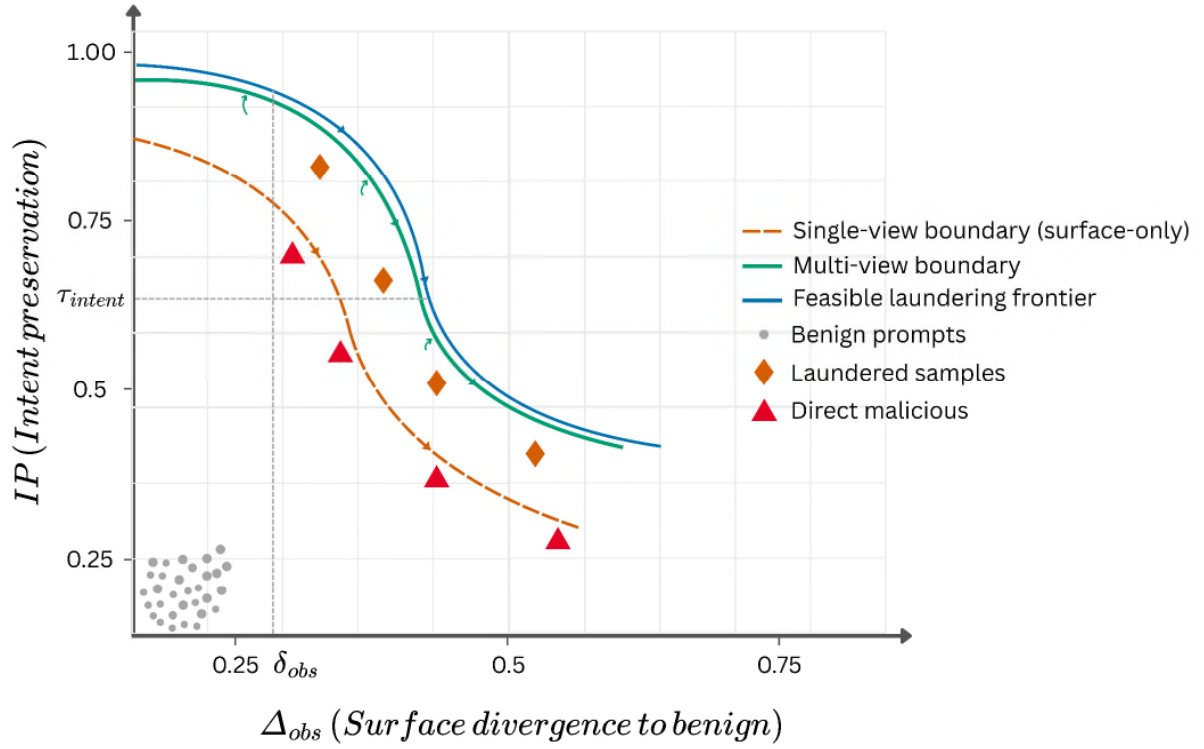


Figure 2: Conceptual laundering frontier and single-view and multi-view boundaries

Conceptual illustration of the relation between surface divergence and intent preservation. The plotted curves are schematic and should not be interpreted as empirical operating boundaries or a proof of an equilibrium. The intended comparison is between single-view and multi-view detection in the low-divergence, high-intent-preservation region.

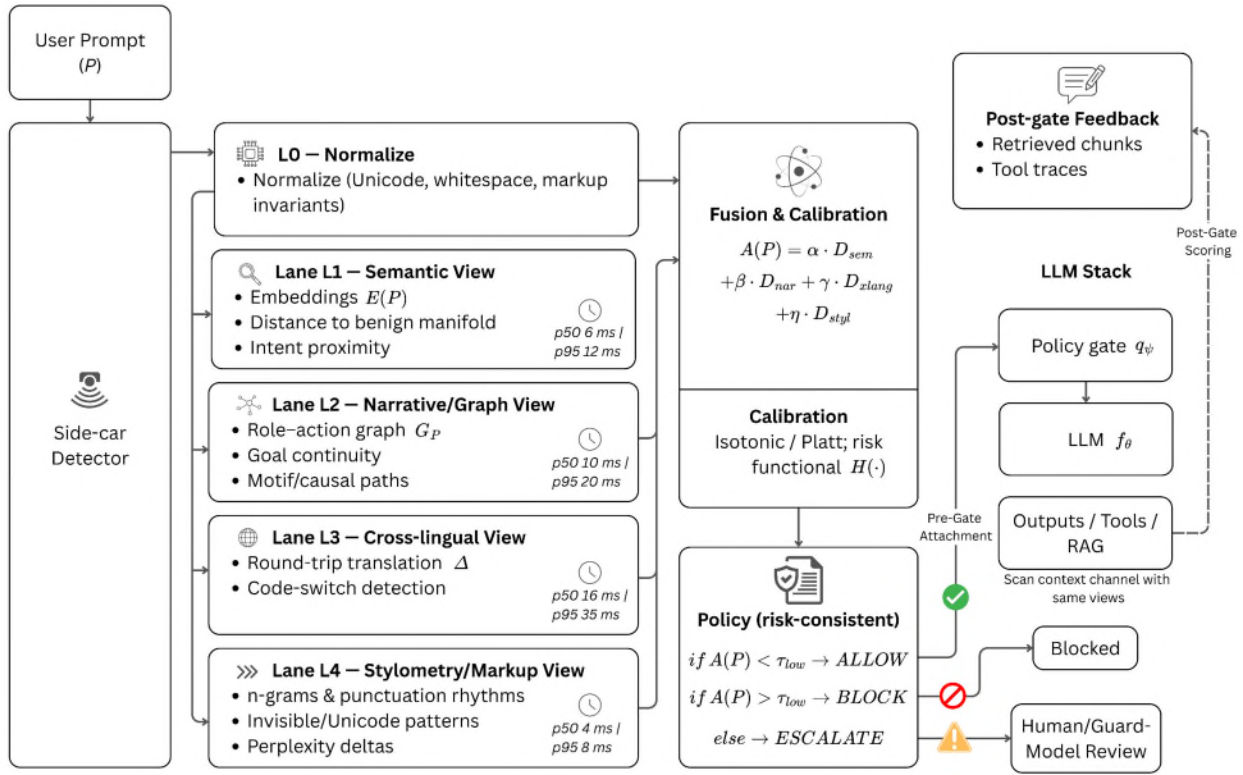


Figure 3: Architecture of the multi-view laundering detector

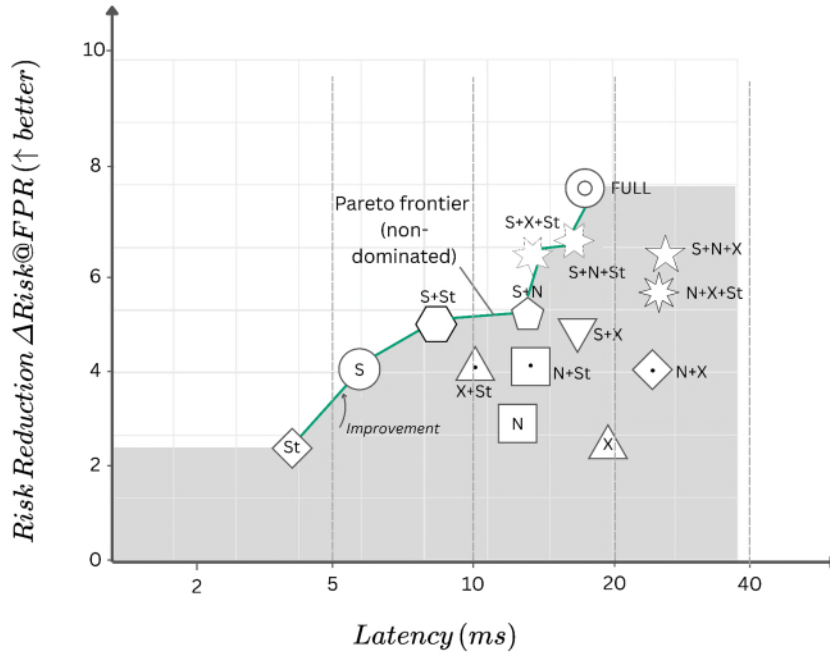


Figure 4: Risk reduction and latency for feature-view subsets

S, N, X, and St denote Semantic, Narrative/Graph, Cross-lingual, and Stylometry/Markup views, respectively. The plot compares view subsets in terms of latency and risk reduction. Its interpretation depends on a common risk baseline and evaluation slice.

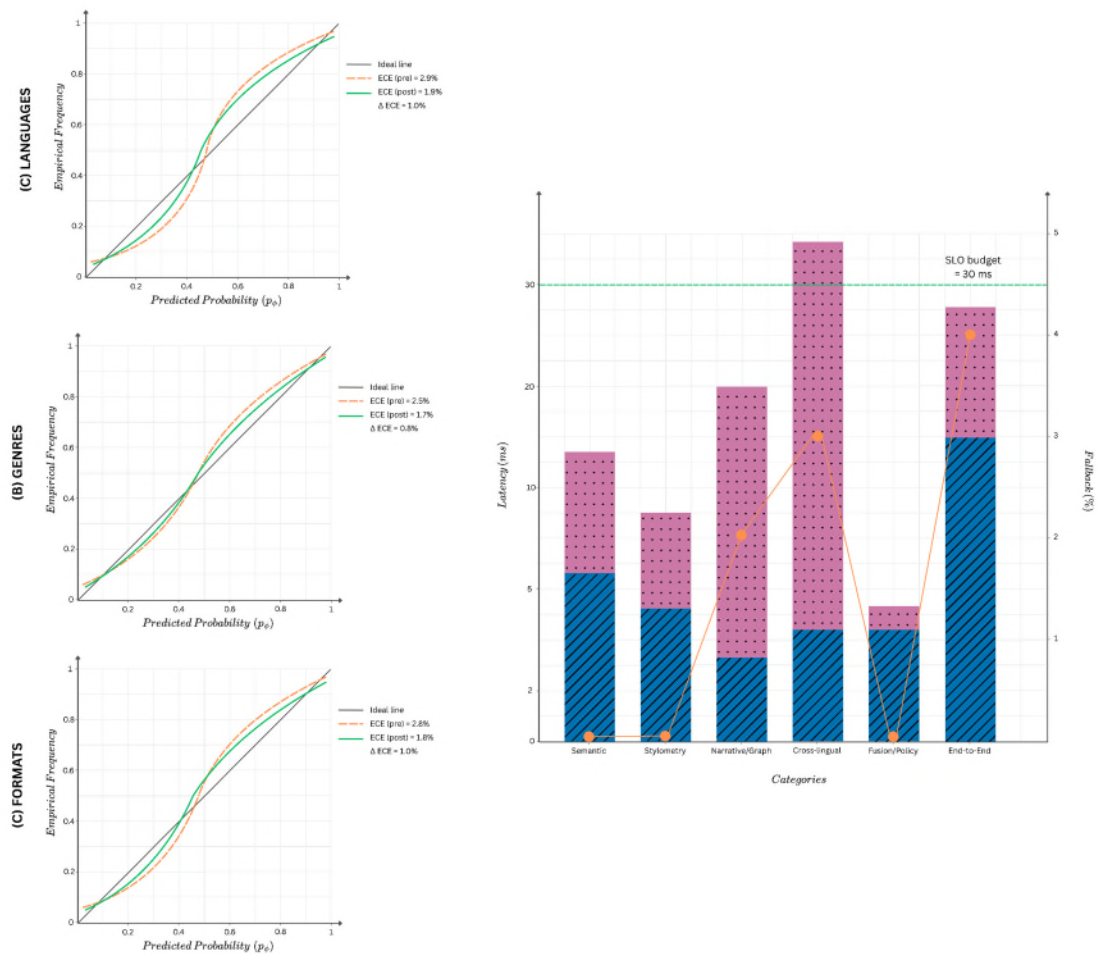


Figure 5: In-distribution calibration reliability (left) and latency by view (right)

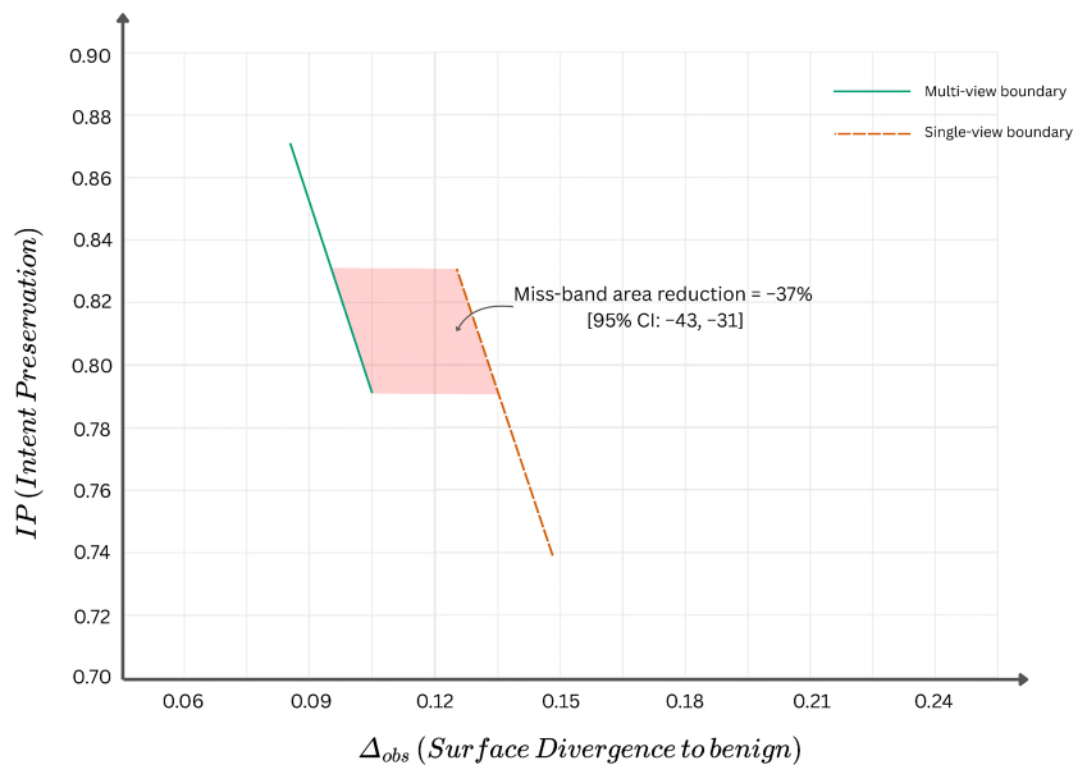


Figure 6: Single-view and multi-view boundaries in the laundering-frontier region

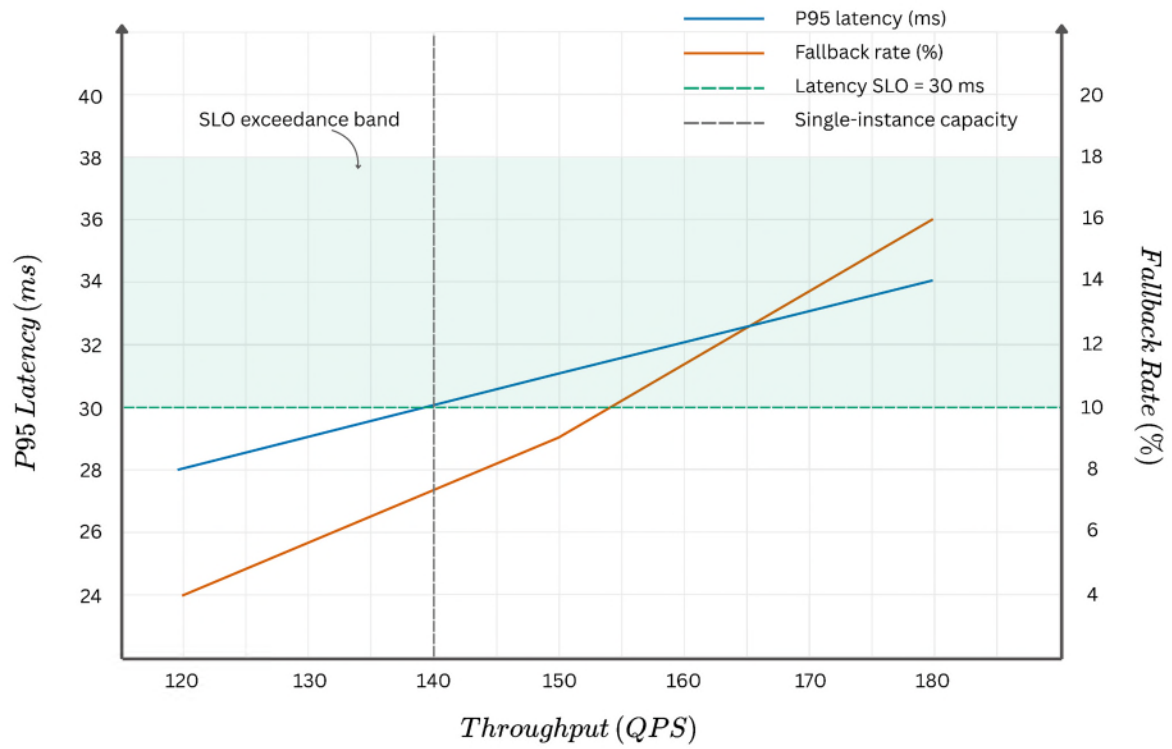


Figure 7: Throughput, p95 latency, and fallback rate

Table 1: Analogy between money laundering and prompt laundering

Characteristic	Money Laundering	Prompt Laundering
Input	Illicit funds	Malicious instructions
Transformation	Layering through banks, shell companies, and cryptocurrencies	Storytelling, encoding, translation, steganography
Output	Apparently legitimate funds usable in the economy	Harmless-seeming prompt usable in LLM interface
Detection challenge	Hidden transaction flows	Hidden semantic intent

Table 2: Dataset composition and factors

Component	Description	Approximate number of instances	Languages (≥ 12 ; $\geq 10\%$ non-Latin)	Genres (≥ 6)	Formats (4)
Benign $\mathcal{D}_b$	General instruction-following, help-seeking, educational Q&A, creative non-harmful	$\approx 500k$ (50%)	Stratified: EN $\sim 40\%$, ES/FR/DE/AR/HI/ZH/RU/JA/KO/PT + others; ensure long-tail strata $\geq 2\%$	Explanatory, storytelling, dialogue, specification, pseudo-code, markup	Plain, Markdown, HTML, Mixed
Laundered $\mathcal{D}_\ell$	Intent-preserving disguises generated from sanitized intents via operator compositions	$\approx 350k$ (35%)	Same strata as $\mathcal{D}_b$; add code-switch subsets ($\geq 15\%$)	Story-camouflage, fictional role-play, procedural/spec hybrid	Heavy Markdown/H TML, mixed punctuation, Unicode variants
Direct-malicious $\mathcal{D}_m$	Redacted malicious intents (operational details removed)	$\approx 150k$ (15%)	Match $\mathcal{D}_b$, language mix; include non-Latin	Spec-like, imperative, stepwise	Plain/Markdown
Context items	Retrieved passages, tool traces, agent memory (for RAG/agents)	$\approx 200,000$ auxiliary items linked to prompts	Mirrors prompt strata	Varied	Plain/HTML/JSON
Held-out OOD test	Unseen languages & operator compositions	$\approx 50k$	Languages not in train (e.g., TR, FA, HE)	Mix	Mix
TOTAL	Prompts only; auxiliary context excluded	$\approx 1,000,000$	—	—	—

* The held-out OOD test is sealed: it is excluded from training, validation, hard-negative mining, calibration, prototype refresh, and threshold selection. Language-level hold-outs include TR, FA, and HE. Operator-level hold-outs include unseen ordered compositions of known primitives, plus a smaller family-level stress slice in which an operator family is reserved entirely for testing. “Unseen” therefore refers to split membership and operator construction, not to the general linguistic capability of the underlying multilingual encoder.

Table 3: Implementation specification for CLEANSE*

Component	Default implementation	Version / checkpoint	Parameters	Trainable in CLEANSE*?	Output used by fusion	Reproducibility details
Semantic encoder (E(P))	intfloat/multilingual-e5-large-instruct	HF checkpoint pinned by SHA	560M	Frozen backbone; trainable projection head W_s	1024-d embedding, semantic distance, prototype similarity	Fixed task prefix: “Represent the safety-relevant intent of this prompt for laundering detection.”; L2 normalization; deterministic batching; prototype bank (K=192), sensitivity at (K=128,256)
Semantic density model	Gaussian mixture / energy-style benign manifold	scikit-learn GMM + calibrated energy layer	N/A for GMM; projection/calibration parameters reported in config	Yes	Benign-manifold distance $D_{\text{benign}}(P)$	Fit only on benign training fold; validation-selected covariance regularization; content-hash linked to split
Malicious-intent prototype bank	Sanitized prototype embeddings	Internal sanitized prototype set	(K=192) prototypes	Prototype embeddings fixed after training fold construction	Unsafe-intent proximity score	Operational harmful text redacted; prototype IDs, source category, and hashes released; exact unsafe wording not released
Narrative/event parser	Stanza Universal Dependencies pipeline	Stanza v1.8.x, language-specific UD models	Language-specific; model hashes reported in supplement	Frozen	Dependency frames, entities, predicates, role/action candidates	Same parser version for all models; parser confidence retained; alternative-parser sensitivity reported
Role-action graph builder	Deterministic graph constructor over parser outputs	CLEANSE* graph builder v1.0	N/A	No	Nodes: role/entity/goal/action; edges: precondition/action/effect	Rules released; unsafe motif labels sanitized; graph construction seeded and hash-logged

Component	Default implementation	Version / checkpoint	Parameters	Trainable in CLEANSE*?	Output used by fusion	Reproducibility details
Graph similarity scorer	Weisfeiler–Lehman graph kernel + logistic calibration	GraKeL / NetworkX implementation pinned in lock file	N/A for kernel; calibration layer parameters reported	Calibration only	Narrative/graph anomaly score	Kernel depth (h=3); motif library sanitized; parser-confidence weighting enabled
Cross-lingual translation model	NLLB-200 distilled translation model	facebook/nllb-200-distilled-600M	600M	Frozen	Round-trip semantic drift and alignment variance	Same checkpoint and decoding settings for all runs; max length and beam size fixed
Round-trip language set	Fixed target-language set L_rt	ES, FR, DE, AR, HI, ZH, RU, JA, KO, PT	Not applicable	No	Cross-lingual stability score	NLLB codes: spa_Latn, fra_Latn, deu_Latn, arb_Arab, hin_Deva, zho_Hans, rus_Cyrl, jpn_Jpan, kor_Hang, por_Latn; EN used as pivot/identity reference
Code-switch detector	Unicode script segmentation + language ID	fastText lid.176 + Unicode script rules	126 MB model file	Frozen	Code-switch ratio, script entropy	Same language-ID threshold across all splits; script-normalization rules released
Stylometry/markup extractor	Character/token n-grams, Unicode classes, punctuation rhythm, markup parser	scikit-learn + regex/HTML parser	N/A	Feature weights learned in fusion	Stylometry/markup anomaly score	Zero-width, Unicode category, Markdown/HTML/XML features; exact feature list released
Perplexity delta model	Base LM vs instruction-tuned LM scoring	DistilGPT-2 / instruction-tuned small LM pair, pinned checkpoints	~82M / checkpoint-specific	Frozen	Perplexity delta feature	Used only as scalar feature; same tokenization and truncation settings across models
Fusion layer	Logistic stacking + isotonic calibration	CLEANSE* fusion v1.0	<1M	Yes	Final anomaly score (A(P))	Trained on development fold; isotonic calibration fitted only on validation fold

Component	Default implementation	Version / checkpoint	Parameters	Trainable in CLEANSE*?	Output used by fusion	Reproducibility details
Decision policy	Three-way allow, escalate, or block policy	Thresholds τ_{low} and τ_{high} from validation fold	N/A	Thresholds selected, then frozen	Allow, escalate, block	Thresholds optimized for Risk@FPR; no test-set threshold tuning
Public guard baseline	Llama Guard 3-8B	Pinned public checkpoint	8B	Frozen	Baseline safety score / decision	Same preprocessing, validation-thresholding, metric scripts, and hardware as CLEANSE*
Additional external baselines	WildGuard; ShieldLM	Pinned public checkpoints	Reported per checkpoint	Frozen	Baseline safety score / decision	Used in external JailbreakBench validation; no retraining or threshold retuning
Runtime environment	Docker + CUDA stack	Python 3.10; PyTorch 2.x; CUDA 12.x	N/A	N/A	Full pipeline	requirements.lock, environment.yml, CUDA/BLAS versions, seeds, config files, and Git commit IDs recorded

Table 4: In-distribution (ID) results

Model	Risk@FP R 0.5% Δ vs. best single-view	Risk@FP R 1% Δ vs. best single-view	Risk@FP R 2% Δ vs. best single-view	ASR (%) Δ vs. best single-view	AUROC Δ vs. best single-view	AUPRC Δ vs. best single-view	ECE (%) Δ vs. best single-view	Latency (ms) p50 / p95 Δ p95 vs. best single-view
Industry guard (<i>Llama Guard 3-8B baseline</i>)	6.2 ± 0.4 (+2.6)	7.4 ± 0.5 (+2.8)	9.3 ± 0.6 (+3.2)	19.9 ± 1.2 (+8.3)	0.912 ± 0.010 (-0.040)	0.684 ± 0.018 (-0.128)	4.5 ± 0.5 (+1.8)	22 / 50 (+38)
Semantic-only internal ablation (<i>best single-view</i>)	3.6 ± 0.3 (0.0)	4.6 ± 0.3 (0.0)	6.1 ± 0.4 (0.0)	11.6 ± 0.9 (0.0)	0.952 ± 0.008 (0.0)	0.812 ± 0.015 (0.0)	2.7 ± 0.4 (0.0)	6 / 12 (0)
Narrative/Graph-only	4.1 ± 0.3 (+0.5)	5.3 ± 0.4 (+0.7)	7.0 ± 0.5 (+0.9)	13.8 ± 1.0 (+2.2)	0.941 ± 0.009 (-0.011)	0.775 ± 0.016 (-0.037)	3.1 ± 0.4 (+0.4)	10 / 20 (+8)
Cross-lingual-only	4.5 ± 0.3 (+0.9)	5.8 ± 0.4 (+1.2)	7.5 ± 0.5 (+1.4)	15.2 ± 1.0 (+3.6)	0.935 ± 0.010 (-0.017)	0.746 ± 0.017 (-0.066)	3.4 ± 0.4 (+0.7)	16 / 35 (+23)
Stylometry/Markup-only	5.0 ± 0.4 (+1.4)	6.1 ± 0.4 (+1.5)	7.9 ± 0.5 (+1.8)	16.7 ± 1.1 (+5.1)	0.928 ± 0.011 (-0.024)	0.721 ± 0.017 (-0.091)	3.8 ± 0.5 (+1.1)	4 / 8 (-4)
Multi-view	2.9 ± 0.2 (-0.7)	3.8 ± 0.3 (-0.8)	5.1 ± 0.3 (-1.0)	8.4 ± 0.8 (-3.2)	0.967 ± 0.006 (+0.015)	0.871 ± 0.012 (+0.059)	1.8 ± 0.3 (-0.9)	15 / 28 (+16)

Llama Guard 3-8B was evaluated using the same test split, preprocessing pipeline, batching policy, metric scripts, and hardware as CLEANSE*. Its operating thresholds were selected on the same validation fold to minimize risk at the specified FPR operating points, then fixed before test evaluation.

Table 5: Frontier results

Model	Risk@FPR 0.5% Δ vs. best single-view	Risk@FPR 1% Δ vs. best single-view	Risk@FPR 2% Δ vs. best single-view	ASR (%) Δ vs. best single-view	ECE (%) Δ vs. best single-view
Industry guard (<i>Llama Guard 3-8B baseline</i>)	7.8 ± 0.5 (+2.8)	9.8 ± 0.6 (+3.6)	11.4 ± 0.7 (+3.6)	28.9 ± 1.3 (+10.5)	5.0 ± 0.6 (+1.8)
Semantic-only internal ablation (<i>best single-view</i>)	5.0 ± 0.3 (0.0)	6.2 ± 0.4 (0.0)	7.8 ± 0.5 (0.0)	18.4 ± 1.1 (0.0)	3.2 ± 0.4 (0.0)
Multi-view	3.7 ± 0.3 (−1.3)	4.9 ± 0.3 (−1.3)	6.1 ± 0.4 (−1.7)	12.7 ± 0.9 (−5.7)	2.1 ± 0.3 (−1.1)

Table 6: Normalized probabilistic error, subgroup calibration, and anomaly-score separation

Evaluation split	Model	rBS ↓	BSS ↑	rRMSE _p ↓	NCE ↓	Cohen's d ↑	Cliff's δ ↑	KS-D ↑	Subgroup ECE, median (IQR) ↓	Worst subgroup ECE ↓
ID	Industry guard (Llama Guard 3-8B)	0.72	0.28	0.85	0.78	0.32	0.22	0.24	4.4% (3.9–5.1)	6.1%
ID	Semantic-only	0.55	0.45	0.74	0.62	0.51	0.35	0.37	2.6% (2.2–3.1)	3.8%
ID	CLEANSE* multi-view	0.42	0.58	0.65	0.49	0.68	0.47	0.49	1.7% (1.4–2.0)	2.6%
Frontier	Industry guard	0.84	0.16	0.92	0.86	0.17	0.12	0.13	4.9% (4.3–5.8)	6.8%
Frontier	Semantic-only	0.68	0.32	0.82	0.69	0.28	0.20	0.21	3.1% (2.7–3.7)	4.5%
Frontier	CLEANSE* multi-view	0.54	0.46	0.73	0.55	0.47	0.32	0.34	2.1% (1.8–2.5)	3.2%
OOD	Industry guard	0.90	0.10	0.95	0.92	0.14	0.10	0.11	6.0% (5.3–6.9)	7.8%
OOD	Semantic-only	0.76	0.24	0.87	0.77	0.24	0.17	0.18	4.0% (3.5–4.7)	5.5%
OOD	CLEANSE* multi-view	0.61	0.39	0.78	0.62	0.40	0.27	0.29	2.5% (2.2–2.9)	3.5%

rBS is the Brier score relative to a prevalence-only predictor; $BSS = 1 - rBS$; $rRMSE_p$ is the square root of rBS; NCE is cross-entropy relative to the same reference predictor. Values below 1 for rBS, $rRMSE_p$, and NCE indicate improvement over that reference. Subgroup ECE covers language, script, genre, format, and OOD strata. Cohen's d, Cliff's δ , and KS-D summarize continuous-score separation between benign and laundered prompts. Bold values identify the best reported value within each split.

Table 7: External evaluation on JailbreakBench

External benchmark	Evaluation condition	Model	Risk@FPR@1% ↓	ASR / miss rate ↓	AUROC ↑	AUPRC ↑	ECE ↓
JailbreakBench	Frozen detector; no retraining; no threshold retuning	Llama Guard 3-8B baseline	8.7 ± 0.6	$25.6 \pm 1.4\%$	0.901 ± 0.011	0.662 ± 0.019	$4.8 \pm 0.5\%$
JailbreakBench	Frozen detector; no retraining; no threshold retuning	ShieldLM	6.8 ± 0.5	$20.2 \pm 1.2\%$	0.934 ± 0.009	0.758 ± 0.017	$3.7 \pm 0.4\%$
JailbreakBench	Frozen detector; no retraining; no threshold retuning	WildGuard	6.0 ± 0.4	$17.4 \pm 1.1\%$	0.946 ± 0.008	0.795 ± 0.016	$3.1 \pm 0.4\%$
JailbreakBench	Frozen detector; no retraining; no threshold retuning	Semantic-only	6.4 ± 0.4	$18.9 \pm 1.1\%$	0.941 ± 0.008	0.781 ± 0.016	$3.4 \pm 0.4\%$
JailbreakBench	Frozen detector; no retraining; no threshold retuning	CLEANSE* multi-view	5.3 ± 0.4	$14.8 \pm 1.0\%$	0.956 ± 0.007	0.835 ± 0.014	$2.6 \pm 0.3\%$

JailbreakBench was used once for external evaluation. No JailbreakBench sample was used for training, validation, calibration, threshold selection, prototype refresh, hard-negative mining, or model selection. All models were assessed with the same metric scripts and their validation-selected thresholds. The 1% FPR label refers to the internal validation operating point. Lower risk, ASR/miss rate, and ECE indicate better performance; higher AUROC and AUPRC indicate better ranking performance.

Table 8: RAG/agent post-gate stress test

Evaluation setting	Model / deployment mode	Risk@FPR@1% ↓	ASR / miss rate ↓	Unsafe tool-pass rate ↓	AUROC ↑	ECE ↓	p95 latency ↓
RAG/agent indirect laundering	Llama Guard 3-8B baseline	9.2 ± 0.6	$27.8 \pm 1.4\%$	$8.5 \pm 0.6\%$	0.893 ± 0.012	$4.9 \pm 0.5\%$	50 ms
RAG/agent indirect laundering	CLEANSE* pre- gate only	6.9 ± 0.5	$20.3 \pm 1.2\%$	$5.9 \pm 0.5\%$	0.931 ± 0.009	$3.5 \pm 0.4\%$	28 ms
RAG/agent indirect laundering	CLEANSE* context scan	6.2 ± 0.4	$18.4 \pm 1.1\%$	$4.7 \pm 0.4\%$	0.940 ± 0.008	$3.2 \pm 0.4\%$	33 ms
RAG/agent indirect laundering	CLEANSE* pre+post-gate	5.4 ± 0.4	$15.1 \pm 1.0\%$	$3.6 \pm 0.4\%$	0.956 ± 0.007	$2.6 \pm 0.3\%$	36 ms

Table 9: Statistical inference and ablation interactions

Analysis	Comparison	Statistic / estimate	95% CI	Adjusted p-value	Interpretation
ID Risk@FPR@1%	CLEANSE* vs Semantic-only	$\Delta = -0.8$	$[-1.2, -0.4]$	0.004	Multi-view lowers ID risk
ID ASR	CLEANSE* vs Semantic-only	McNemar ($\chi^2 = 96.4$)	—	<0.001	Fewer paired misses
ID AUROC	CLEANSE* vs Semantic-only	DeLong ($z = 4.02$)	—	<0.001	Higher ranking quality
Frontier Risk@FPR@1%	CLEANSE* vs Semantic-only	$\Delta = -1.3$	$[-1.8, -0.8]$	<0.001	Multi-view lowers frontier risk
Frontier ASR	CLEANSE* vs Semantic-only	McNemar ($\chi^2 = 158.7$)	—	<0.001	Fewer frontier misses
Frontier ECE	CLEANSE* vs Semantic-only	$\Delta = -1.1$ pp	$[-1.6, -0.6]$	<0.001	Better calibration
Interaction: Semantic+Narrative	Pairwise removal	$I(S, N) = +0.4$	$[+0.08, +0.72]$	0.041	Modest positive interaction
Interaction: Narrative+Cross-lingual	Pairwise removal	$I(N, X) = +0.2$	$[-0.08, +0.48]$	0.112	Positive estimate; not statistically significant
Interaction: Semantic+Stylometry	Pairwise removal	$I(S, St) = -0.1$	$[-0.34, +0.13]$	0.318	No statistically significant interaction detected
Interaction: Cross-lingual+Stylometry	Pairwise removal	$I(X, St) = +0.1$	$[-0.12, +0.34]$	0.281	No statistically significant interaction detected

Table 10: Ablation results

Model variant	Risk@FPR @1% (ID)	Risk@FPR @1% (Frontier)	Δ Risk vs full (frontier)	ASR (Frontier)	ECE (Frontier)	p95 (ms)	Δ p95 (ms)	AER (risk units/ms)
Full (multi-view)	3.8	4.9	—	12.7%	2.1%	28	—	—
– Semantic	4.9	6.4	+1.5	16.6%	2.6%	20	–8	0.19
– Narrative/Graph	4.7	6.1	+1.2	15.8%	2.5%	25	–3	0.40
– X-lingual	4.3	5.7	+0.8	14.9%	3.0%	24	–4	0.20
– Stylometry/Markup	4.2	5.5	+0.6	14.3%	2.7%	24	–4	0.15

Table 11: Sensitivity matrix (Δ relative to the default full model)

Factor (setting)	$\Delta\text{Risk@FPR@1\%}$ (ID)	$\Delta\text{Risk@FPR@1\%}$ (Frontier)	$\Delta\text{ECE (Frontier, pp)}$	$\Delta\text{TR (OOD)}$	$\Delta\text{p95 (ms)}$
Calibration: Platt vs Isotonic	+0.10	+0.11	+0.3	+0.00	+0
Escalation band: w=0.25	+0.06	+0.08	+0.1	+0.00	+0
Escalation band: w=0.50	−0.04	−0.06	−0.1	+0.00	+0
Prior shift: +50% positive	+0.20	+0.30	+0.1	+0.00	+0
Label noise: 2% flips	+0.07	+0.10	+0.2	+0.00	+0
Label noise: 5% flips	+0.22	+0.32	+0.5	+0.00	+0
Prototype bank: K = 128	—	+0.22	—	—	+1
Prototype bank: K = 256	—	−0.05	—	—	0
Narrative parser (alt.)	+0.08	+0.20	+0.1	+0.00	−1
Graph noise: 10% edges ↓	+0.12	+0.30	+0.2	+0.00	+0
Graph noise: 20% edges ↓	+0.28	+0.70	+0.5	+0.00	+0
Round-trip language set: 5 languages	—	—	—	+0.05	−1
Markup features off	+0.09	+0.30	+0.3	+0.02	−1
Train size: −25%	+0.20	+0.31	+0.2	+0.01	+0
Train size: +25%	−0.06	−0.10	−0.1	−0.01	+0

For the prototype-bank and reduced-language-set rows, values are reproduced only where the corresponding metric is explicitly identified in the manuscript text. An em dash indicates that a value was not reported unambiguously in the supplied files; it does not mean zero. The reduced-language-set ECE increase of 0.5 percentage points reported in the text concerns OOD evaluation and is not entered in the frontier ECE column.