

Beyond Over-Refusal: Defending Indirect Prompt Injection via Latent Instruction Manifolds

Jiahao Chen¹, Rui Yin¹, Xinfeng Li², Qianli Ma³, Tianyu Du¹

Zhihui Fu⁴, Jun Wang⁴, Zhaoxiang Wang⁴, Shouling Ji¹

¹Zhejiang University, ²Hong Kong Polytechnic University,

³University of Science and Technology of China, ⁴OPPO Research Institute
{xaddwell,ruiyin,zjradty,sji}@zju.edu.cn, qianli-ma@mail.ustc.edu.cn,
{lxfmakeit,hzzhzzf,junwang.lu,steven.wangzx}@gmail.com

Abstract

Large Language Models (LLMs) integrated into complex ecosystems (e.g., code agents) face critical safety threats from Indirect Prompt Injection (IPI) attacks, where malicious instructions are embedded in retrieved data. Existing defenses face an intractable safety-utility trade-off: sanitization-based guardrails incur high latency, while detection-based guardrails suffer from severe *over-refusal*. We first demonstrate, both theoretically and empirically, that LLMs can intrinsically separate instructions from data. Inspired by this insight, we propose **AEGIS** (Adaptive Ensemble Guard for Injection Shielding), a training-free defense that extracts **instruction-sensitive projectors** to identify malicious instructions and leverages a **Unified Multi-Layer Consensus** that aggregates topologically distinct signals across the network depth. Evaluations on eight IPI attacks and ten baselines show that **AEGIS** achieves state-of-the-art detection accuracy while reducing the false positive rate on benign tasks to below 1%. Code is available at [Github](#).

1 Introduction

Large Language Models (LLMs) have evolved from passive text generators into active agents that execute code, manage user data, and interact with external tools through Retrieval-Augmented Generation (RAG) (Taori et al., 2023; Ouyang et al., 2022; Shi et al., 2025). This integration, however, exposes a fundamental vulnerability: LLMs lack an explicit distinction between *instructions* (control flow) and *knowledge* (external data) within their context (Willison, 2022; Zhang et al., 2025).

This ambiguity enables Indirect Prompt Injection (IPI) attacks, where adversaries embed malicious commands into retrieved data streams, such as web pages or emails, to hijack the model’s behavior (Willison, 2022; Zhang et al., 2025), potentially causing data exfiltration (Lin et al., 2026) or the

propagation of harmful content (Hui et al., 2024; Collu et al., 2025). Existing countermeasures struggle to reconcile the *safety-utility* trade-off: sanitization approaches, such as structured queries (Chen et al., 2025a) and token-wise filtering (Geng et al., 2025; Liu et al., 2025a), degrade utility and incur high computational overhead, whereas detection guardrails, whether external classifiers (Alon and Kamfonas, 2023; BlueteamAI, 2024) or internal steering mechanisms (Hung et al., 2025), suffer from *over-refusal* (Li et al., 2025).

We challenge the assumption that instruction and data are indistinguishable for LLMs (Zverev et al., 2024). We argue that the key to resolving this trade-off lies in the intrinsic geometry of the latent space shaped by the two-stage training paradigm: pre-training forces knowledge representations to occupy a high-entropy, high-dimensional manifold, whereas **instruction tuning** collapses instruction representations into a compact operator subspace (Park et al., 2024; Zou et al., 2023a). Detection should therefore rely on signatures within this compact instruction manifold rather than black-box classification. We introduce **AEGIS** (Adaptive Ensemble Guard for Injection Shielding), a lightweight, training-free detection framework that isolates the core “imperative intent” behind superficial phrasing differences and aggregates evidence across multiple network depths via a *Unified Multi-Layer Consensus* mechanism. Our contributions are summarized as:

- We characterize the geometric separation between instruction and knowledge representations induced by instruction tuning, translating it into an **instruction-sensitive projector**.
- We propose **AEGIS**, combining distribution-aware threshold calibration with multi-layer consensus voting, with **theoretical error bound** for the ensemble false positive rate

(FPR) and a latent robustness radius against hidden-state perturbations.

- Extensive evaluations on eight IPI attacks with ten baselines demonstrate that **AEGIS** achieves state-of-the-art detection performance while reducing the FPR on benign tasks to negligible levels ($< 1\%$).

2 Related Work

Instructions vs. data. Instruction fine-tuning aligns pre-trained LLMs with human intent (Taori et al., 2023; Ouyang et al., 2022) and fundamentally alters the geometry of the latent space (Zou et al., 2023a): SFT collapses imperative instructions into a low-dimensional operator subspace, distinct from the high-entropy knowledge manifold acquired during pre-training (Park et al., 2024). While whether LLMs can intrinsically separate instructions from data has been questioned (Zverev et al., 2024), we provide theoretical and empirical evidence in Section 3 that they can.

IPI attacks and defenses. IPI attacks have evolved from manual heuristics such as “Context Ignoring” and “Fake Completion” (Willison, 2022; Perez and Ribeiro, 2022; Liu et al., 2024b) to optimization-based methods such as GCG (Zou et al., 2023b), Universal Injection (Liu et al., 2024a), and NeuralExec (Pasquini et al., 2024), as well as agent-oriented attacks targeting tool selection (Shi et al., 2025; Zhang et al., 2025) and prompt leaking (Hui et al., 2024). Defenses fall into two lines: *sanitization* methods (e.g., StruQ (Chen et al., 2025a), PISanitizer (Geng et al., 2025), DRIP (Liu et al., 2025a), SecAlign (Chen et al., 2025b)) neutralize threats but impose utility penalties and high latency; *detection* methods, including external classifiers (PIBert (BlutteamAI, 2024), DeBERTa (ProtektAI.com, 2023)), perplexity-based filters (Alon and Kamfonas, 2023), and internal-state monitors (InjecGuard (Li and Liu, 2024), Attention Tracker (Hung et al., 2025)), often lack context-awareness or suffer from over-refusal (Li et al., 2025). **AEGIS** belongs to the detection line yet differs fundamentally: it is training-free and grounded in the spectral geometry of the latent space.

3 Methodology

3.1 Threat Model

The adversary can only inject malicious instructions I_{adv} into the external data segment D_{ext} (e.g.,

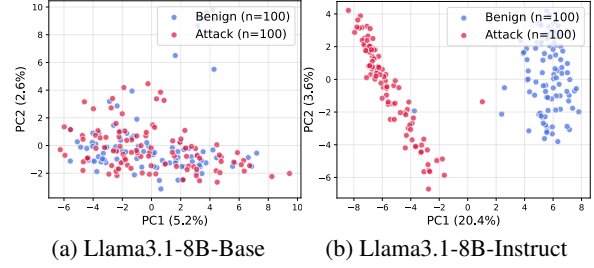


Figure 1: PCA visualization of the last-token hidden state at layer 25 on two versions of Llama3.1-8B.

poisoned web pages retrieved by the system) (Willison, 2022; Liu et al., 2024b), without access to the model weights, the system prompt, or the user query. The defender, as the model owner, has white-box access to the LLM at inference time and knows the segmentation boundaries of D_{ext} . The defense must satisfy *low-latency* and *utility-preservation* constraints (Li et al., 2025).

3.2 Spectral Asymmetry and Latent Decomposition

Pre-training forces knowledge representations h_K to occupy a high-dimensional, high-variance manifold $\mathcal{M}_K$ to encode diverse facts, while SFT collapses instruction representations h_I into a low-dimensional, compact subspace $\mathcal{M}_I$; Figure 1 shows that hidden states of the instruct model exhibit much higher separability than those of the base model. Let Σ_K, Σ_I be the covariance matrices of knowledge and instruction representations; we posit $\|\Sigma_K\|_2 \gg \|\Sigma_I\|_2$. Detecting an injection is thus equivalent to detecting an h_I buried in knowledge noise. Following the Linear Representation Hypothesis (Park et al., 2024), we model the activation at layer l as a superposition:

$$h^{(l)} = h_K^{(l)} + \alpha \cdot h_{I_{adv}}^{(l)} + \epsilon \quad (1)$$

where $\alpha \in \{0, 1\}$ indicates the presence of an attack. The goal is to find a projection $f : \mathbb{R}^d \rightarrow \{0, 1\}$ that maximizes the detection probability while bounding FPR $P(f(h) = 1 \mid \alpha = 0) \leq \tau$.

3.3 The AEGIS Framework

Instruction-Sensitive Projector. A naïve separation of centroids is insufficient because instruction representations vary along “synonymy axes” (e.g., “Summarize” vs. “TL;DR”) while remaining invariant along the “intent axis.” We therefore seek a projection direction w that maximizes the separation between instruction and knowledge centroids

while minimizing the variance *within* the instruction class:

$$w^* = \arg \max_w \frac{(w^T(\mu_I - \mu_K))^2}{w^T \Sigma_I w}, \quad (2)$$

where μ_I, μ_K and Σ_I, Σ_K denote the class centroids and covariances, estimated with Ledoit–Wolf shrinkage. Solving via Lagrange multipliers yields the closed-form **Instruction-Sensitive Projector**:

$$w^* \propto \Sigma_I^{-1}(\mu_I - \mu_K) \quad (3)$$

The term Σ_I^{-1} acts as a **Semantic Focuser**: it down-weights dimensions where instructions vary lexically and up-weights dimensions where instructions are compact and distinct from knowledge, locking the detector onto the stable semantic core of an imperative command.

Distribution-Aware Safety Calibration. A fixed geometric margin is hazardous due to the heavy-tailed nature of the knowledge distribution. Without parametric assumptions about the knowledge manifold, we anchor the decision boundary to the observed projection statistics of the benign knowledge set $\mathcal{D}_{benign}$: for a required safety level β (e.g., guaranteeing 99% utility retention), we set

$$b = -\mathcal{Q}_{1-\beta}(\{w^{*T}h \mid h \in \mathcal{D}_{benign}\}) \quad (4)$$

where $\mathcal{Q}_p(\cdot)$ denotes the p -th quantile function. The detector thus triggers on at most a β fraction of benign queries, mathematically anchoring its sensitivity to the model’s actual behavior on safe inputs.

Multi-Layer Consensus Voting. Single-layer detectors, while calibrated, may exhibit localized sensitivity to noise or “polysemantic” overlaps. We model the layers as a committee: let $S_l(x) = w_l^{*T}h^{(l)}$ be the raw projection score at layer $l \in \mathcal{L}_{vote}$; the ensemble score aggregates transformed layer-wise signals,

$$\mathcal{S}_{ens}(x) = \sum_{l \in \mathcal{L}_{vote}} \phi_l(S_l(x)), \quad (5)$$

with the final decision $\hat{y} = \mathbb{I}(\mathcal{S}_{ens}(x) \geq \Gamma)$. We support two modes: (1) *hard voting*, $\phi_{hard}(s) = \mathbb{I}(s > b_l)$, where b_l is the FPR-calibrated threshold, $\mathcal{S}_{ens}$ is the raw vote count, and Γ the consensus quorum; and (2) *soft voting*, $\phi_{soft}(s) = (s - \mu_{benign}^{(l)}) / \sigma_b^{(l)}$, where $\mu_b^{(l)}$ and $\sigma_b^{(l)}$ are the mean and standard deviation of benign scores at layer l .

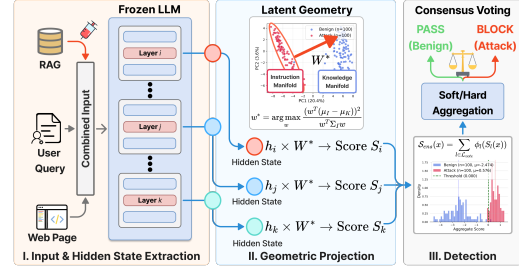


Figure 2: Conceptual illustration of the AEGIS.

An input is flagged only if a persistent malicious signal appears throughout the network depth, eliminating false positives caused by transient noise. Figure 2 illustrates the AEGIS framework.

3.4 Theoretical Guarantees

For **Hard Voting**, define $I_l = \phi_{hard}(S_l(x))$ with $P(I_l = 1) \leq \beta$ for benign inputs. Assuming approximate independence between non-adjacent layers (González et al., 2025), the ensemble FPR $Q_{ens} = P(\mathcal{S}_{ens} \geq \Gamma)$ follows the tail of a Binomial, and **Hoeffding’s inequality** (Hoeffding, 1963) yields an exponential bound:

$$Q_{ens} \leq \exp \left(-2|\mathcal{L}_{vote}| \left(\frac{\Gamma}{|\mathcal{L}_{vote}|} - \beta \right)^2 \right) \quad (6)$$

As long as the quorum ratio $\Gamma/|\mathcal{L}_{vote}|$ (e.g., 0.5) exceeds the single-layer error rate β (e.g., 0.1), the false-alarm probability decays *exponentially* with the number of voting layers, allowing a looser per-layer β (higher recall) without compromising global safety.

We further analyze robustness against a *latent adversary* that adds a perturbation δ directly to hidden states. With unit-normalized projectors, the per-layer margin to the decision hyperplane is $R_l(x) = b_l - S_l(x)$; the ensemble robustness radius is exactly the Γ -th smallest per-layer margin, $R_{ens}(x) = R_{(\Gamma)}$. Any perturbation with $\|\delta\|_2 < R_{(\Gamma)}$ is thus guaranteed to preserve safe classification: the ensemble is as strong as its “median” link, forcing an attacker to simultaneously satisfy geometric constraints across Γ distinct semantic spaces.

4 Experiments

4.1 Experimental Settings

Following prior work (Liu et al., 2024b; Hung et al., 2025), we use the *OpenPromptInjection* benchmark with 7 NLP tasks (100 instruction-data pairs each). We evaluate AEGIS against 8 IPIs: five

Attack ↓	Defense →	AttTracker	DeBERTa	InjecGuard	KA	LLM Naive	PIBert	PPL ¹	PPL ²	PrGuard	PIShield	AEGIS
Combined Attack	ACC ↑	0.8071	0.6219	0.9836	0.7195	0.6710	0.6050	0.4508	0.5182	0.5371	0.5758	0.9848
	FPR ↓	0.3857	0.0057	0.0243	0.4931	0.6069	0.7900	0.5492	0.7100	0.9257	0.8171	0.0290
	FNR ↓	0.0000	0.7504	0.0086	0.0680	0.0510	0.0000	0.5492	0.2537	0.0000	0.0312	0.0014
Escape Character	ACC ↑	0.8071	0.4996	0.6005	0.6701	0.5560	0.6050	0.3996	0.4597	0.5371	0.5790	0.9852
	FPR ↓	0.3857	0.0057	0.0243	0.4967	0.6029	0.7900	0.6004	0.7100	0.9257	0.8171	0.0290
	FNR ↓	0.0000	0.9951	0.7747	0.1631	0.2851	0.0000	0.6004	0.3706	0.0000	0.0249	0.0006
Fake Completion	ACC ↑	0.8071	0.4999	0.6124	0.7082	0.6254	0.6050	0.4347	0.5051	0.5371	0.5769	0.9848
	FPR ↓	0.3857	0.0057	0.0243	0.4963	0.6024	0.7900	0.5653	0.7100	0.9257	0.8171	0.0290
	FNR ↓	0.0000	0.9945	0.7508	0.0873	0.1467	0.0000	0.5653	0.2798	0.0000	0.0290	0.0013
Context Ignoring	ACC ↑	0.8071	0.6873	0.9823	0.7056	0.4616	0.6050	0.4535	0.4990	0.5371	0.5755	0.9850
	FPR ↓	0.3857	0.0057	0.0243	0.4971	0.6155	0.7900	0.5465	0.7100	0.9257	0.8171	0.0290
	FNR ↓	0.0000	0.6196	0.0110	0.0916	0.4612	0.0000	0.5465	0.2920	0.0000	0.0318	0.0021
Naive Attack	ACC ↑	0.8071	0.4996	0.6005	0.7018	0.4585	0.6050	0.4231	0.4752	0.5371	0.5783	0.9850
	FPR ↓	0.3857	0.0057	0.0243	0.4994	0.6051	0.7900	0.5769	0.7100	0.9257	0.8171	0.0290
	FNR ↓	0.0000	0.9951	0.7747	0.0969	0.4780	0.0000	0.5769	0.3396	0.0000	0.0263	0.0010
NeuralExec	ACC ↑	0.8071	0.5620	0.9879	0.7404	0.6964	0.6050	0.7867	0.6413	0.5371	0.5786	0.9847
	FPR ↓	0.3857	0.0057	0.0243	0.4957	0.6020	0.7900	0.2133	0.7100	0.9257	0.8171	0.029
	FNR ↓	0.0000	0.8702	0.0000	0.0235	0.0051	0.0000	0.2133	0.0073	0.0000	0.0257	0.0016
PLeak	ACC ↑	0.8071	0.8170	0.8429	0.6180	0.6391	0.6050	0.6271	0.6106	0.5371	0.5893	0.9855
	FPR ↓	0.3857	0.0057	0.0243	0.4945	0.6035	0.7900	0.3729	0.7100	0.9257	0.8171	0.0290
	FNR ↓	0.0000	0.3602	0.2900	0.2696	0.1184	0.0000	0.3729	0.0688	0.0000	0.0043	0.0000
Universal Injection	ACC ↑	0.8071	0.5304	0.9879	0.7371	0.6118	0.6050	0.5963	0.6029	0.5371	0.5774	0.9850
	FPR ↓	0.3857	0.0057	0.0243	0.4922	0.6035	0.7900	0.4037	0.7100	0.9257	0.8171	0.0290
	FNR ↓	0.0000	0.9335	0.0000	0.0335	0.1729	0.0000	0.4037	0.0843	0.0000	0.0280	0.0010

Table 1: Overall detection performance (ACC, FPR and FNR) comparison between AEGIS and baselines.

heuristic attacks (Naive (Harang, 2023), Escape Characters (Perez and Ribeiro, 2022), Context Ignoring (Willison, 2022), Fake Completion (Willison, 2022), Combined (Liu et al., 2024b)) and three optimization-based attacks (Universal Injection (Liu et al., 2024a), NeuralExec (Pasquini et al., 2024), PLeak (Hui et al., 2024)). We report accuracy (ACC), FPR, FNR, TPR, F1, and computational overhead on Qwen3-4B (Team, 2025) (default), Llama3-8B (AI@Meta, 2024), and Llama3.1-8B (Grattafiori et al., 2024). Baselines comprise PIBert (BlueteamAI, 2024), DeBERTa (ProtectAI.com, 2023), PromptGuard (PrGuard) (Meta, 2024), PPL (Alon and Kamfonas, 2023) (GPT2 for PPL¹ and Vicuna-7B for PPL²), Known-Answer (KA) (Nakajima, 2022), InjecGuard (Li and Liu, 2024), Attention Tracker (AttTracker) (Hung et al., 2025), LLM-Naive (Armstrong, 2023), and PIShield (Zou et al., 2025), all with thresholds tuned on the same validation benign set. **AEGIS** estimates projectors from only 200 benign and 200 malicious calibration samples; by default it aggregates with soft voting over a window starting at 50% network depth ($|\mathcal{L}_{vote}| = 3$ for Qwen3-4B and 7 for the Llama models), and all results are averaged over 5 random seeds.

4.2 Main Results

Table 1 reports detection performance across the 8 IPIs. Existing defenses gravitate toward one extreme: AttTracker and PrGuard achieve low FNR

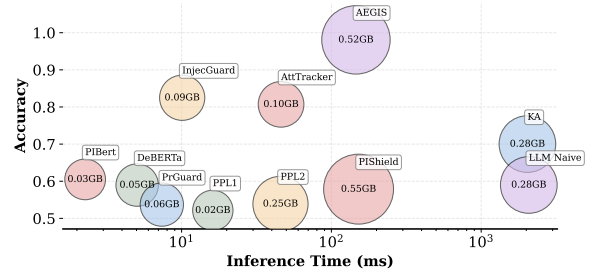


Figure 3: Efficiency-accuracy trade-off comparison.

but over-refuse benign queries (FPR 38.57% and 92.57%), while shallow classifiers such as DeBERTa and PIBert keep low FPR ($\approx 0.5\%$) but miss heuristic attacks (FNR up to 99.51% on Naive Attacks). In contrast, **AEGIS** achieves an average ACC above 98% across most tasks with a consistent FPR of **0.29%**, confirming that malicious intent can be identified without compromising the processing of safe queries. **AEGIS** also generalizes across both attack paradigms, exceeding 98% accuracy where perplexity-based methods struggle (e.g., PPL¹ reaches only 59.63% on Universal Injection), validating the multi-layer consensus in capturing deep semantic anomalies that persist across network depth.

Figure 3 visualizes the accuracy-latency trade-off on Qwen3-4B, with bubble size encoding peak GPU memory. External guardrails such as PIBert and DeBERTa are cheap (< 10 ms, < 0.1 GB) but insufficiently accurate ($\approx 60\%$), while LLM-based judges incur prohibitive latency (> 1000 ms).

AEGIS occupies the optimal position on the Pareto frontier: with a marginal latency increase (≈ 120 ms), it achieves near-perfect accuracy ($> 98\%$), demonstrating a favorable balance between robust safety and deployment feasibility.

4.3 Ablation Studies

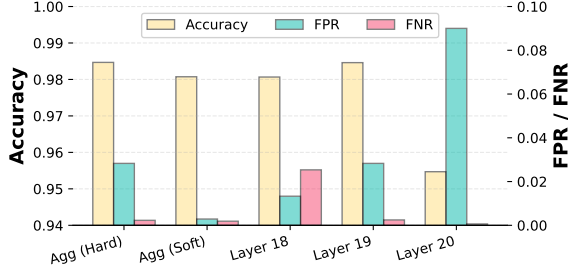


Figure 4: Ablation study of detection settings.

Multi-layer consensus. Figure 4 compares the ensemble against single-layer detectors on Qwen3-4B: while individual layers (e.g., Layer 19) achieve high accuracy, they exhibit instability in FPR or FNR, whereas both *Agg (Hard)* and *Agg (Soft)* stabilize the FPR while maintaining accuracy above 97%. Aggregating topologically distinct signals thus filters out layer-specific noise, creating a robust detection boundary.

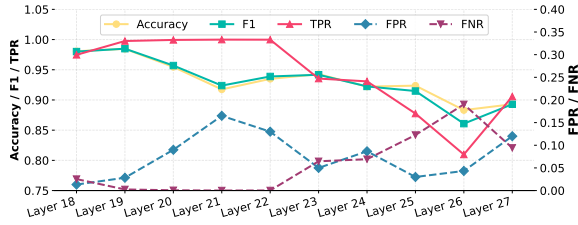


Figure 5: Detection performance of different layers.

Performance across layers. A core premise of our work is that instruction and knowledge representations exhibit distinct spectral signatures at specific depths. Figure 5 shows detection performance across layers 18–27 on Qwen3-4B: accuracy peaks and FPR is minimized in the middle layers, while FNR increases in deeper layers as the instruction manifold collapses into the token generation subspace. Starting aggregation at 50% network depth with 3–7 voting layers is optimal. This justifies our strategy of targeting intermediate layers for projector extraction.

Cross-model generalization. To verify that **AEGIS** captures fundamental properties of LLMs rather than architecture-specific artifacts, we evaluate it on three model families. Figure 6 shows

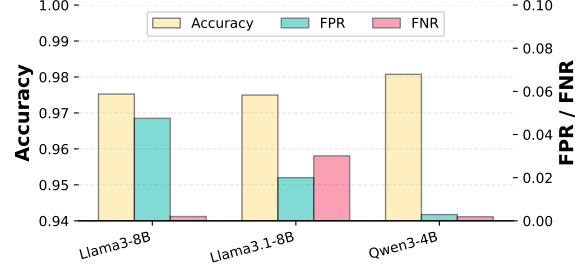


Figure 6: Detection performance of different models.

that **AEGIS** maintains accuracy above 97% and FPR below 5% on Llama-3-8B, Llama-3.1-8B, and Qwen3-4B, suggesting that the spectral asymmetry between instruction and knowledge is a generic structural invariant introduced by alignment rather than an architecture-specific artifact.

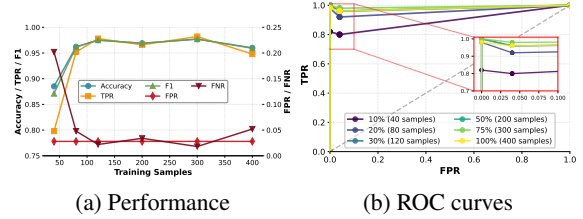


Figure 7: Data efficiency of **AEGIS** (Llama3.1-8B) with calibration set sizes from 40 to 400 samples.

Data efficiency. A practical defense must not require extensive datasets for calibration. Figure 7 shows that **AEGIS** converges to optimal performance with as few as 100 calibration samples: accuracy and F1 plateau quickly, and the ROC curves retain high AUC even in low-data regimes. The instruction manifold is evidently low-dimensional and stable, requiring minimal samples to characterize its principal components.

5 Conclusion

We analyzed the conflation of instructions and data that renders LLMs susceptible to prompt injection, and identified the spectral asymmetry between the compact instruction manifold and the high-entropy knowledge manifold as a critical leverage point. We proposed **AEGIS**, a training-free defense that extracts instruction-sensitive projectors and aggregates evidence through a Unified Multi-Layer Consensus mechanism with layer-wise calibration. Theoretical analysis yields an exponential ensemble FPR bound and a latent robustness radius, and empirical evaluations demonstrate state-of-the-art detection with negligible impact on benign utility.

Limitations

While **AEGIS** offers robust defense guarantees, we acknowledge several limitations: (1) **AEGIS** requires access to internal hidden states, limiting direct applicability to closed-source commercial models; (2) the error bound relies on an approximate layer-independence assumption (González et al., 2025), making Equation 6 a potentially optimistic upper bound; and (3) detection introduces non-zero computational overhead that may affect latency in ultra-high-throughput scenarios. These limitations are shared by other IPI defenses (Liu et al., 2025b; Jia et al., 2026; Chen et al., 2025a; Geng et al., 2025).

Ethical Considerations

We adhere to the ACM Code of Ethics and the ACL Code of Conduct. Our experiments rely exclusively on established public benchmarks; we release no new datasets containing harmful content, and **AEGIS** will be open-sourced to foster verification and replication. We acknowledge the potential for dual use in security research and encourage responsible use of **AEGIS** for defensive purposes only.

LLM Usage Considerations

We utilized Large Language Models (specifically Gemini3-Pro and GPT-5) to assist with checking grammatical errors and refining sentence structure. All scientific claims, experimental designs, theoretical derivations, and final content decisions remain the sole responsibility of the human authors.

References

- AI@Meta. 2024. Llama 3 model card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.
- Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity.
- R Gorman Stuart Armstrong. 2023. Using GPT-eliezer against ChatGPT jailbreaking. <https://www.alignmentforum.org/posts/pNcFYZnPdXyL2RfgA/using-gpt-eliezer-against-chatgpt-jailbreaking>.
- BlueteamAI. 2024. fmops/distilbert-prompt-injection. <https://huggingface.co/fmops/distilbert-prompt-injection>.
- Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. 2025a. StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2383–2400.
- Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, David Wagner, and Chuan Guo. 2025b. SecAlign: Defending against prompt injection with preference optimization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 2833–2847.
- Matteo Gioele Collu, Umberto Salviati, Roberto Confalonieri, Mauro Conti, and Giovanni Apruzzese. 2025. Publish to perish: Prompt injection attacks on LLM-assisted peer review.
- Runpeng Geng, Yanting Wang, Chenlong Yin, Minhao Cheng, Ying Chen, and Jinyuan Jia. 2025. PISanitizer: Preventing prompt injection to long-context LLMs via prompt sanitization.
- Ramón Calvo González, Daniele Paliotta, Matteo Pagliardini, Martin Jaggi, and François Fleuret. 2025. Leveraging the true depth of LLMs. *CoRR*, abs/2502.02790.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models.
- Rich Harang. 2023. Securing LLM systems against prompt injection. <https://developer.nvidia.com/blog/securing-llm-systems-against-prompt-injection>.
- Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30.
- Bo Hui, Haolin Yuan, Neil Gong, Philippe Burlina, and Yinzhi Cao. 2024. PLeak: Prompt leaking attacks against large language model applications. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3600–3614.
- Kuo-Han Hung, Ching-Yun Ko, Ambrish Rawat, I-Hsin Chung, Winston H Hsu, and Pin-Yu Chen. 2025. Attention tracker: Detecting prompt injection attacks in LLMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2309–2322.
- Yuqi Jia, Yupei Liu, Zedian Shao, Jinyuan Jia, and Neil Zhenqiang Gong. 2026. PromptLocate: Localizing prompt injection attacks. In *IEEE Symposium on Security and Privacy*.
- Hao Li and Xiaogeng Liu. 2024. InjecGuard: Benchmarking and mitigating over-defense in prompt injection guardrail models.
- Hao Li, Xiaogeng Liu, Ning Zhang, and Chaowei Xiao. 2025. PIGuard: Prompt injection guardrail via mitigating overdefense for free. In *Proceedings of the*

- 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30420–30437, Vienna, Austria. Association for Computational Linguistics.
- Ruixiao Lin, Qingming Li, Jiahao Chen, Chunyi Zhou, and Shouling Ji. 2026. SOPE: Situation-aware and statistically indistinguishable privacy exfiltration for MCP-enabled agents. In *Forty-third International Conference on Machine Learning*.
- Ruofan Liu, Yun Lin, and Jin Song Dong. 2025a. DRIP: Defending prompt injection via de-instruction training and residual fusion model architecture.
- Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. 2024a. Automatic and universal prompt injection attacks against large language models.
- Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. 2024b. Formalizing and benchmarking prompt injection attacks and defenses. In *USENIX Security Symposium*.
- Yupei Liu, Yuqi Jia, Jinyuan Jia, Dawn Song, and Neil Zhenqiang Gong. 2025b. DataSentinel: A game-theoretic detection of prompt injection attacks. In *IEEE Symposium on Security and Privacy*.
- Meta. 2024. meta-llama/Prompt-Guard-86M. <https://huggingface.co/meta-llama/Prompt-Guard-86M>.
- Yohei Nakajima. 2022. Yohei’s blog post. <https://twitter.com/yoheinakajima/status/1582844144640471040>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, pages 39643–39666. PMLR.
- Dario Pasquini, Martin Strohmeier, and Carmela Troncoso. 2024. Neural exec: Learning (and learning from) execution triggers for prompt injection attacks. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, pages 89–100.
- Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models.
- ProtectAI.com. 2023. Fine-tuned DeBERTa-v3 for prompt injection detection. <https://huggingface.co/ProtectAI/deberta-v3-base-prompt-injection>.
- Jiawen Shi, Zenghui Yuan, Guiyao Tie, Pan Zhou, Neil Zhenqiang Gong, and Lichao Sun. 2025. Prompt injection attack to tool selection in LLM agents.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following LLaMA model.
- Qwen Team. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Simon Willison. 2022. Prompt injection attacks against GPT-3.
- Yitong Zhang, Ximo Li, Liyi Cai, and Jia Li. 2025. Realistic environmental injection attacks on GUI agents.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xu Wang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023a. Representation engineering: A top-down approach to AI transparency.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023b. Universal and transferable adversarial attacks on aligned language models.
- Wei Zou, Yupei Liu, Yanting Wang, Ying Chen, Neil Gong, and Jinyuan Jia. 2025. PIShield: Detecting prompt injection attacks via intrinsic LLM features.
- Egor Zverev, Sahar Abdelnabi, Mario Fritz, and Christoph H Lampert. 2024. Can LLMs separate instructions from data? and what do we even mean by that? In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.