BERM: Low-Overhead Prompt-Injection Detection via In-Situ Benign Representation Modeling

Maihao Guo[1] , **Chaoyang Zhao**[1] and **Jinqiao Wang**[1,2*]
[1] Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
maihao.guo@ia.ac.cn, {chaoyang.zhao, jqwang}@nlpr.ia.ac.cn

Abstract

Real-world deployment of large language models (LLMs) necessitates a robust and low-latency approach to detect prompt injections; existing low-overhead methods fail to simultaneously boost robustness and reduce latency. Current defenses for prompt injection either rely on brittle heuristics or invoke costly auxiliary models, imposing a significant runtime burden. We introduce BERM, a lightweight framework that performs in-situ detection by modeling a host LLM's internal representations extracted during prefill, adding negligible overhead. Our approach trains a lightweight classifier atop the LLM by learning a compact manifold of benign representations via joint contrastive learning to maximize the separation from malicious representations. At inference, this pre-trained classifier enables in-situ detection without invoking auxiliary guard models. On a diverse landscape of prompt injection attacks, our framework establishes a new state-of-the-art, achieving an F1-score 5.2 percentage points (pp) higher than the best prior work. Critically, BERM achieves this while being over 12x faster, reducing incremental inference overhead to near-zero.

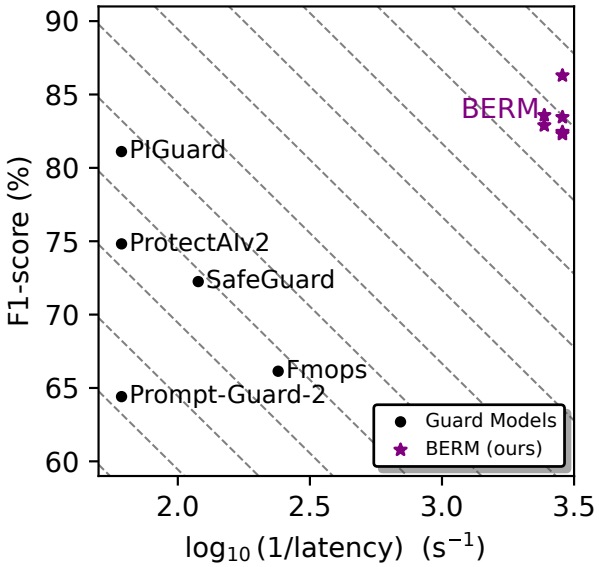

Figure 1: **Latency–performance frontier for prompt-injection detection.** Moving toward the upper-right is strictly better. BERM achieves an absolute **+5 pp F1** while being **12–47× faster** than existing guard models (black dots), establishing a new Pareto frontier.

1 Introduction

Large language models (LLMs) power mission-critical services such as chatbots, code assistants and search engines, answering billions of queries every day. Yet their remarkable flexibility also exposes a rapidly evolving threat: *prompt injection*. In these exploits, attackers cleverly embed harmful directives within apparently benign prompts, tricking models into revealing private information or bypassing built-in safety restrictions. According to the OWASP, prompt injection represents the most critical security vulnerability for LLM deployments[OWASP, 2023]. Current defense strategies typically involves a difficult trade-off between robustness and model performance, a central dilemma highlighted in recent security standards [Vassilev *et al.*, 2025].

The overhead–robustness dilemma. Security teams implementing LLM protections routinely face a wrenching tradeoff: strengthening defenses inevitably slows down response times. Lightweight keyword filters are fast but crumble under sophisticated attacks; meanwhile, robust model-based guards–typically DeBERTa classifiers [He *et al.*, 2021], follow an *external-guard* paradigm that adds ≥ 16 ms latency and is often limited to a 512-token context window. These technical limitations critically undermine their real-world deployability where user experience demands minimal response times and prompts routinely exceed these context windows.

Our approach: In-Situ Benign Representation Modeling (BERM). We address this dilemma with **BERM**, a novel framework that casts prompt injection as an out-of-distribution shift in the host LLM's own representation space. By harnessing the hidden-state representations extracted in-

*Corresponding author.

Method	#Injection Training Samples	Overall				Unforeseen				Overhead		
		Accu.	P	R	F1	Accu.	P	R	F1	GFLOPs	Inf. (ms)	Eff.
Guard Models												
Prompt-Guard-2 [2024]	–	77.21	96.78	53.57	64.41	62.64	94.06	32.56	45.55	60.45	16.33	0.04
Fmops [2024]	203	56.51	56.55	93.59	66.14	45.17	42.07	97.56	58.22	21.77	4.17	0.16
SafeGuard [2023]	6k	75.68	77.92	76.29	72.25	60.55	72.67	53.84	55.09	30.23	8.37	0.09
ProtectAIv2 [2024]	10k	77.28	87.79	70.91	74.82	60.86	77.83	58.60	59.74	60.45	16.33	0.05
PIGuard [2025]	15k	84.94	90.24	74.65	81.11	78.21	89.24	68.89	77.31	60.45	16.34	0.05
BERM (Ours)												
Mistral-7B-v0.1	6k	81.13	77.95	92.18	83.45	68.36	63.85	94.67	76.05	0.05	0.35	2.38
Llama-3.1-8B-Instruct	6k	84.01	81.46	93.83	**86.29**	71.82	66.41	95.27	**78.18**	**0.05**	**0.35**	**2.47**
L1-32B	6k	80.97	75.98	92.81	82.89	70.44	65.08	95.68	77.45	0.06	0.41	2.02
BERM@300 (Ours)												
Mistral-7B-v0.1	300	82.03	76.99	89.36	82.44	74.87	72.46	85.75	**78.47**	**0.05**	**0.35**	**2.36**
Llama-3.1-8B-Instruct	300	82.93	81.12	84.71	82.30	76.55	79.77	78.70	78.17	0.05	0.35	2.35
L1-32B	300	84.02	83.04	84.51	**83.58**	75.39	74.52	82.44	78.19	0.06	0.41	2.04

Table 1: **Overall vs. Unforeseen performance of prompt-injection methods.** *Overall* performance merges the test portion of the training dataset with the disjoint prompt-injection dataset, reflecting in-distribution accuracy. *Unforeseen* performance, comprising only previously unseen datasets, measures robustness out of the training distribution. *Overhead* performance reports the compute in GFLOPs, Inference latency (Inf.) in ms and F1-Efficiency (Eff.) = *Overall* F1 / Inf. adapted from the metric in PIGuard [Li *et al.*, 2025]. BERM variants (lower blocks) achieve a +5 pp improvement in *Overall* F1 score over the best guard model while maintaining competitive *Unforeseen* F1 scores. BERM@300, our few-shot variant with 300 benign / 300 injections, demonstrates superior sample efficiency, outperforming PIGuard in all metrics. All baselines evaluated on the same GPU. All BERM variants reduce inference latency by more than an order of magnitude compared to the best guard model.

situ during prefill, BERM bypasses the need for external guards. In their raw form, these representations are geometrically entangled (AUROC ≈ 0.59). The learned projector then re-embeds them in a more compact manifold where benign and injected prompts become separable. Figure 1 depicts BERM pushing the latency–performance frontier outward relative to prior guard models.

BERM leverages a joint contrastive learning scheme fusing (i) an unsupervised, cluster-based objective modeling the benign manifold and (ii) a supervised objective separating known attacks, allowing BERM to learn a robust notion of normalcy from as few as 300 injections.

Contributions.

1. **Framework.** BERM pairs strong prompt-injection detection with *near-zero* overhead. It improves overall F1 by **+5.2 pp** and cuts inference latency by $\geq 12\times$.

2. **Learning objective.** We propose a joint contrastive loss tailored for prompt-injection detection, achieving high sample efficiency by fusing unsupervised benign modeling with supervised ID–OOD separation.

3. **Analysis tool.** We develop the *Whitened Intrinsic Distance Estimator* (WIDE), a post-hoc diagnostic that audits representation quality and benign–injection overlap.

Road-map. Section 3 details BERM and WIDE; Section 4 reports experiments on four corpora and three backbones; Section 5 discusses limitations and future work.

2 Related Work

Prompt Injection Attacks. Prompt injection plants adversarial text into an LLM's context, prompting the model to override, ignore or leak the original prompt [Perez and Ribeiro, 2022]. Direct attacks are authored by the user [Willison, 2023]; indirect attacks can be embedded in third-party content fetched by a tool or search engine, taking effect once that content is appended to the prompt [Greshake *et al.*, 2023]. Techniques include fake completions [Willison, 2023], multi-strategy chains [Yi *et al.*, 2025] and gradient-searched suffixes [Liu *et al.*, 2024]. Recent large-scale evaluations find that even state-of-the-art models can be jailbroken reliably–often with near-perfect success–and that such prompts transfer across model families, highlighting the seriousness of the threat [Shen *et al.*, 2024].

Prompt Injection Defense. Existing defenses can be grouped into three lines of work: (i) Architecture-level fixes separate control from data, e.g., StruQ's dual channels [Chen *et al.*, 2025], instruction hierarchies [Wallace *et al.*, 2024], or task-specific de-instruction tuning [Piet *et al.*, 2024]. These require weight access and compute-intensive retraining, and may constrain free-form usage. (ii) Self-supervision adds meta-queries or reminder tokens so the model audits itself [Phute *et al.*, 2024; Xie *et al.*, 2023], straightforward to integrate but consumes context and lengthens latency. Hybrid pipelines that cascade detectors with architectural separation often inherit both cost and brittleness [Suo, 2024]. (iii) Detection filters route each prompt through a smaller surrogate classifier [Blueteam AI, 2024; ProtectAI.com., 2024; Shang *et al.*, 2023] or a second LLM

in zero-shot mode [Meta, 2024]. The low-overhead runner-up of this paradigm–PIGuard [Li *et al.*, 2025],relies on markedly more training data.

Out of Distribution (OOD) Detection. Our work frames prompt-injection detection as a fine-grained case of Out-of-Distribution (OOD) detection. Foundational works–supervised [Hendrycks *et al.*, 2019; Hendrycks *et al.*, 2020] and unsupervised like ContraOOD [Zhou *et al.*, 2021] achieved strong OOD detection by outlier exposure and contrastive learning to separate coarse-grained domain shifts (e.g., news vs. movie reviews). However, they are less effective for prompt injection. The distinction between benign and injected inputs is fine-grained and often hinges on adversarial phrasing within an otherwise in-distribution context. Within highly homogeneous domains like mathematical reasoning, Wang et al. [2025] leverage the volatility of this layer-wise hidden-state trajectory as a powerful OOD signal. This finding corroborates that a LLM's internal processing dynamics encode crucial cues for distinguishing in- from out-of-distribution inputs.

3 Methodology

We propose **BERM (Benign Representation Modeling)**, a framework that detects prompt-injection attacks by modeling the in-situ hidden state representations of a frozen large language model. Prompt injections can steer a model away from its intended instruction-following behavior. Our core intuition is that this deviation manifests as a geometric signature in the in-situ representation space, making it distinguishable from benign queries. BERM consists of an offline representation-learning stage and an online lightweight scoring stage. The methodology is built on three pillars:

(1) benign-manifold discovery via clustering, (2) joint contrastive learning that compactly encodes the benign manifold, and (3) whitened-distance auditing by Whitened Intrinsic Distance Estimator.

3.1 BERM Backbone

Let f_θ denote a frozen host LLM. For a prompt x, we take the token-level hidden states $h(x) \in \mathbb{R}^{T\times D}$ from the **final layer**. The final layer is directly upstream of next-token prediction, since its hidden states feed the language-model head used for token generation [Ethayarajh, 2019]. For a prompt x of length T, we compute a prompt-level embedding by mean-pooling token-level hidden states, following common sentence-embedding practices [Reimers and Gurevych, 2019; Gao *et al.*, 2021]:

$$\mathbf{z}(x) = \frac{1}{T}\sum_{t=1}^{T} h_t(x), \qquad \tilde{\mathbf{z}}(x) = \frac{\mathbf{z}(x)}{\|\mathbf{z}(x)\|_2}. \tag{1}$$

This D-dimensional vector is the sole input to the BERM projector.

3.2 Benign Manifold Discovery

GMM Clustering. Given benign prompts $\mathcal{D}_{\text{id}} = \{x_i\}_{i=1}^N$, we extract their embeddings $\{\tilde{\mathbf{z}}_i\}$ and model them with a Gaussian mixture of C components. Each sample produces a soft assignment $p(c \mid \tilde{\mathbf{z}}_i)$; we collapse it to the MAP label $y_i^{\text{clus}} = \arg\max_c p(c \mid \tilde{\mathbf{z}}_i)$.

3.3 Joint Contrastive Learning

Training only on labeled injection examples tends to overfit to dataset-specific attack templates. In contrast, generic unsupervised anomaly detection can miss fine-grained malicious instructions embedded in otherwise benign prompts. We address this gap with a *joint* objective. The unsupervised cluster-contrast loss ($\mathcal{L}_{\text{unsup}}$) tightens the benign manifold to support outlier detection, while the supervised ID–OOD contrast ($\mathcal{L}_{\text{sup}}$) repels known attacks from benign clusters.

Projector Network. A three-layer residual MLP $g_\phi : \mathbb{R}^D \to \mathbb{R}^D$ maps

$$\mathbf{e} = g_\phi(\tilde{\mathbf{z}}), \qquad \hat{\mathbf{e}} = \frac{\mathbf{e}}{\|\mathbf{e}\|_2}. \tag{2}$$

Unsupervised Cluster Contrast. An instance of the InfoNCE loss [van den Oord *et al.*, 2018], minimises

$$\mathcal{L}_{\text{unsup}} = -\frac{1}{|B|}\sum_{i\in B}\log\frac{\sum_{j\in B,\, y_j^{\text{clus}}=y_i^{\text{clus}}}\exp(\hat{\mathbf{e}}_i^\top \hat{\mathbf{e}}_j/\tau)}{\sum_{k\in B,\, k\neq i}\exp(\hat{\mathbf{e}}_i^\top \hat{\mathbf{e}}_k/\tau)}, \tag{3}$$

for mini-batch $B \subseteq \mathcal{D}_{\text{id}}$,with temperature $\tau = 0.15$.

Supervised ID-OOD Contrast. With a labeled set $\mathcal{D}_{\text{ood}} = \{(x,\ell)\}$, $\ell \in \{0,1\}$, we optimize a NT-Xent objective [Chen *et al.*, 2020], treating all benign embeddings as positives and all malicious ones as negatives.

Binary Classification Head. A linear layer $q_\psi : \mathbb{R}^D \to \mathbb{R}^2$ returns logits $\mathbf{o}_i$; we minimize cross-entropy

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|B|}\sum_{i\in B}\log\Big(\text{softmax}(\mathbf{o}_i)_{y_i}\Big). \tag{4}$$

Joint Objective. Training minimizes a weighted combination of the three terms:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{unsup}} + \beta\mathcal{L}_{\text{sup}} + \theta\mathcal{L}_{\text{CE}}. \tag{5}$$

Unless noted otherwise, we set $\alpha{:}\beta{:}\theta = 1{:}3{:}1$.

3.4 Inference Path and Latency

During LLM prefill we extract the in-situ prompt representation $\tilde{\mathbf{z}}_{\text{test}}$ and score it with:

$$p_{\text{attack}} = \text{softmax}\big(q_\psi(g_\phi(\tilde{\mathbf{z}}_{\text{test}}))\big)_1 \tag{6}$$

With $D = 5120$ the projector imposes ≤ 0.06G FLOPs overhead, delivering a $12\times$ latency reduction compared to the leanest guard baselines (Table 1)

3.5 WIDE: Whitened Intrinsic Distance Estimator

To assess the representation quality of BERM and other prompt-injection detectors, we introduce **WIDE (Whitened Intrinsic Distance Estimator)**, a lightweight post-hoc analysis tool. WIDE takes a set of embeddings and summarizes how tightly benign samples cluster by measuring distances to a fitted benign manifold in a whitened space. Rather than acting as a detector by itself, WIDE serves as a diagnostic: it quantifies separability and generalization capacity of the representation space itself. Given benign (in-distribution) embeddings $\mathcal{Z}_{\text{id}} = \{\mathbf{z}_i\}_{i=1}^N \subset \mathbb{R}^D$, WIDE assigns each test embedding $\mathbf{z}_{\text{test}}$ an anomaly score based on its whitened distance to the benign manifold.

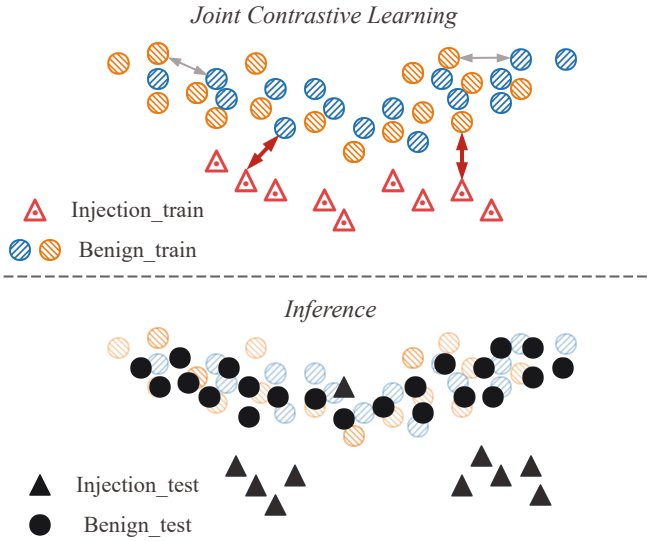

Figure 2: **Joint contrastive learning and its geometric effect.** This figure illustrates a 2-D projection of the learned latent representation space. **Top:** during training, grey thin arrows depict the *unsupervised cluster contrast*, which enlarges the gap between benign clusters, while red arrows shows the *supervised ID–OOD contrast* that repels injected prompts from *every* benign cluster. **Bottom:** once training converges, benign test samples (solid dots) concentrate on or near the compact benign manifold, whereas most novel injections (solid triangles) are projected farther away. Collectively, these objectives reorganize the projected space , which simplifies inference-time scoring.

Step 1: Benign Manifold Statistics. We start by summarizing the benign embeddings with their second-order statistics. Specifically, we estimate the sample mean

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{z}_i, \tag{7}$$

and the sample covariance matrix $\mathbf{\Sigma}$, which describes the manifold's shape, orientation, and variance along its principal axes.

Step 2: Denoising via Principal Component Analysis. High-dimensional embeddings can contain noisy or uninformative directions. We eigendecompose the covariance matrix, $\mathbf{\Sigma Q} = \mathbf{Q\Lambda}$, where $\mathbf{Q}$ contains the eigenvectors and $\mathbf{\Lambda}$ is the diagonal matrix of corresponding eigenvalues. We retain the top-k principal components that together explain a ratio η (e.g., 99.5%) of the total variance, thereby projecting the data onto a lower-dimensional denoised subspace prior to whitening.

Step 3: Whitening Transformation. Based on the denoised principal components $(\mathbf{Q}_k, \mathbf{\Lambda}_k)$, we construct a whitening matrix:

$$\mathbf{W} = (\mathbf{\Lambda}_k + \varepsilon\mathbf{I})^{-\beta/2}\mathbf{Q}_k^\top. \tag{8}$$

This transformation rotates a deviation vector into the principal component space and re-scales it, amplifying deviations along compact, low-variance dimensions of the benign manifold while suppressing them along high-variance ones. The parameter $\beta \in (0, 1]$ controls this suppression strength, while $\varepsilon > 0$ ensures numerical stability.

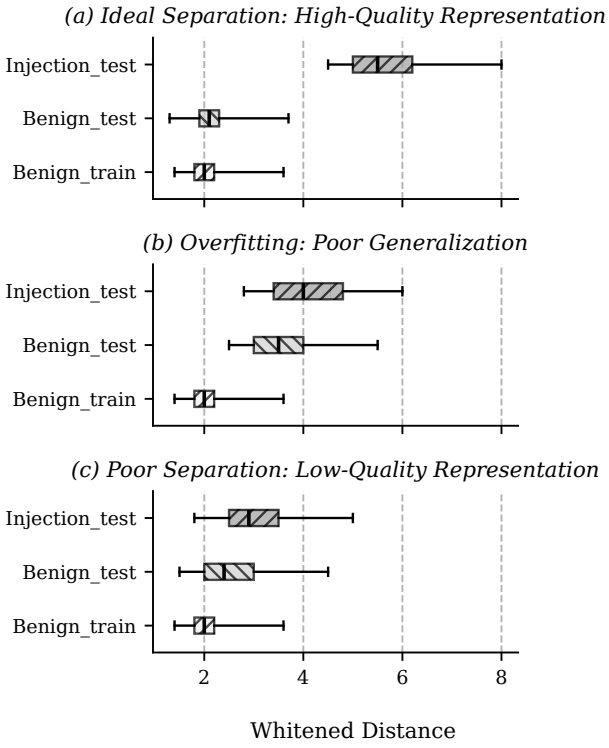

Figure 3: Interpreting model behavior with Whitened Intrinsic Distance Estimator(WIDE). Each panel illustrates how score distributions diagnose representation quality. (a) Successful learning: The benign and malicious clusters distribute distinctly apart. (b) Overfitting detected: Benign test samples are anomalously scored higher than their training counterparts, conflating with the adversarial cluster. (c) Model Collapse: The profound intermingling across datasets signals a collapse in learning, where the model fails to carve out distinguishable representations of the underlying manifolds.

Step 4: Anomaly Scoring. For any test embedding $\mathbf{z}_{\text{test}}$, we compute the final anomaly score as its Mahalanobis-like distance in this transformed space:

$$s(\mathbf{z}_{\text{test}}) = \|\mathbf{W}(\mathbf{z}_{\text{test}} - \mu)\|_2. \tag{9}$$

A large score s intuitively indicates that $\mathbf{z}_{\text{test}}$ is improbably far from the benign manifold, particularly along dimensions where benign samples show little variation.

WIDE as a Diagnostic Tool. To translate geometric insights into actionable diagnostics, we derive two metrics from the anomaly score distributions shown in Fig. 3: **Whiten-AUROC** and **Whiten-FPR**$_{95}$.

- **Ideal Representation (Fig. 3a):** Benign and malicious score distributions are clearly separated (Whiten-AUROC $\approx$ 1.0). This signifies a high-quality embedding space where the model has learned a compact and generalizable manifold for benign prompts. As shown in Table 1, our BERM model trained on the full Safe Guard Prompt Injection dataset achieves this on in-distribution data (Whiten-AUROC=0.999).

- **Overfitting Diagnosis (Fig. 3b):** A key diagnostic is the divergence between in-distribution and out-of-distribution test performance. If the Whitened distances for the benign training set are low, yet the distance distribution of the benign test set largely overlaps with that of the injection test set, this provides a strong quantitative signal of overfitting.
- **Predicting Performance Bottlenecks:** WIDE metrics expose geometric constraints in representation spaces that shape classifier capabilities. As Fig. 3c demonstrates, collapsed Whiten-AUROC values reveal substantial overlap between benign and malicious patterns, exposing that effective detection requires first untangling these representations at the embedding level. This geometric diagnosis uncovers which representation spaces preserve sufficient discriminative integrity to support reliable security decisions.

Beyond BERM. WIDE transcends its origins as a BERM analysis companion and functions as a general geometric probe for representation spaces. By calculating whitened distances from test samples to an authentic in-distribution manifold, WIDE exposes the representation quality of any embedding architecture: (a) WIDE distills manifold geometry into a singular diagnostic metric whose values presage the feasibility of linear decision boundaries; (b) per-sample distance visualizations could expose manifold collisions and potential representation flaws.

3.6 Complexity

WIDE incurs an offline cost of $\mathcal{O}(ND^2 + D^3)$ operations with $\mathcal{O}(D^2)$ memory footprint. During deployment, online inference incurs an $\mathcal{O}(kD)$ overhead, which is submillisecond on GPU for $D \leq 5120$ in our setting. This computational frugality helps BERM maintains its real-time security posture without compromising system responsiveness.

3.7 Summary of Methodology

BERM optimizes a joint contrastive objective to sculpt a compact manifold of benign prompts. At inference time, each new prompt is scored with just one extra pass through a small projector head computation, adding only a tiny incremental cost. WIDE provides an geometric readout of how distinctly benign and injected prompts diverge, so we can assess representation quality decoupled from the classifier's decision logic.

4 Experiments

Our empirical analysis rigorously scrutinizes BERM across the robustness–efficiency spectrum. The goal is to quantify how much robustness can be achieved when both incremental compute and labeled injections are constrained. We therefore organized the evaluation to probe: (1) How does BERM compare with state-of-the-art guard models across diverse benchmarks? (2) How sample-efficient is BERM in low-data regimes? (3) What is the contribution of each component in our joint contrastive objective?

4.1 Experimental Setup

Datasets. Our evaluation draws on four public corpora that are commonly used in recent prompt-injection studies: **SafeGuard** (SG) [Shang *et al.*, 2023], **Jailbreak Classification** (JC) [Shen *et al.*, 2024], **DeepSet Prompt Injections** (DS) [Deepset, 2024], and **Prompt-Injection-Safety** (PS) [Jayavibhavnk, 2024]. SG and JC supply labelled *training* splits; DS and PS serve as test partitions and are solely used for out-of-distribution (OOD) evaluation.

Protocol rationale. PIGuard classifier [2025], the latest over-defense mitigating DeBERTa, is trained on a corpus with 61 k benign + 15 k injections, and evaluated partially on private corpus, together contains $>$1.5 k non-English samples. We limit the study to publicly available, English-only corpora to keep all methods on equal footing and to focus on the *Overhead-Robustness dilemma*. We train **BERM** on SG (6k) to fairly compare with baselines trained on thousands of injections; **BERM@300** on a random 300-sample subset to mirror low-label settings (§4.2).

Baselines. Five DeBERTa-based guards: **Fmops** [2024], **SafeGuard** [2023], **ProtectAI-v2** [2024], **Prompt-Guard-2** [2024],**PIGuard** [2025].

Metrics. Accuracy (A), Precision (P), Recall (R), F1, AUROC, FPR_{95}, GFLOPs, and latency (ms). All results are averaged over five seeds (2022–2026).

Implementation. We instantiate BERM on three frozen backbones: *Mistral-7B-v0.1* [Jiang *et al.*, 2023], *Llama-3.1-8B-Instruct* [Meta AI, 2024], *L1-32B* [Wan *et al.*, 2025]. [1] We extract in-situ hidden-state representations during the host LLM's native prefill stage, using the final-layer hidden states for their accessibility in most LLM inference stacks.

We use a fixed set of hyperparameters rather than exhaustive tuning: For the Bayesian GMM, we set the Dirichlet prior to 0.1, and run EM for 1000 steps, using 2 clusters for BERM@300, 5 clusters for BERM(@6k); Adam optimizer learning rate 5×10^{-3}, batch size 256; ResMLP hidden size 4096; InfoNCE $\tau = 0.15$. We train BERM for 5 epochs with Adam. Training runs on two NVIDIA RTX-A6000 (48 GB) GPUs. Maximum wall-time for 5K prompts is 30 minutes.

4.2 Main Results

Table 1 benchmarks **BERM** with five premier stand-alone guard models. We instantiate BERM on three host LLMs: *Mistral-7B-v0.1* [2023], *Llama-3.1-8B-Instruct* [2024], and *L1-32B* [2025], to validate the universality of benign-representation modeling across model families and scales.

Accuracy and OOD robustness. On the four-corpus aggregate, **BERM (8B)** reaches an *Overall* F1 of **86.29%**, outperforming the previous best PIGuard by **+5.2 pp**. The margin persists on the *Unforeseen* split (**78.18%** F1), suggesting that BERM does more than track dataset-specific templates and can generalize to unseen corpora. Scaling the host to **L1-32B** still delivers 82.89 % Overall and 77.45 % Unforeseen F1, indicating that BERM transfers to substantially larger (RL-tuned) backbones.

[1] Code available at https://github.com/CytAI/BERM

Low and scalable overhead. Guard baselines require ≈ 60 GFLOPs and 16 ms per prompt, whereas BERM adds only **0.05 GFLOPs** and **0.35 ms**, a **400–1200**$\times$ reduction in (incremental) compute and a **12–47**$\times$ reduction in latency. The projector cost is governed by the hidden dimension ($D \leq 5120$) rather than host parameter count. The same overhead applies to **BERM (32B)**, keeping throughput essentially unchanged as the host scales up.

Sample efficiency. The few-shot variant **BERM@300** uses only **300** injected prompts and still secures 83.58 % Overall F1 on L1-32B, surpassing PIGuard, which relies on 15k injections. This mirrors the trend in Fig. 4 and supports the view that explicitly modeling benign structure can reduce the amount of injection samples needed (see §4.3).

4.3 Few-Shot Sample Efficiency

A cardinal advantage of our joint-learning framework is its capacity to exploit scarce malicious data, a common constraint in production environments. To investigate this, we evaluate BERM's performance when trained with a varying number of malicious samples, ranging from just 100 up to the full 6k set. We probe this by sweeping the injection sample size from 100 to 6k. Figure 4 plots the F1 trajectories for both in-distribution (ID) and out-of-distribution (OOD) test sets.

The results underscore BERM's acute sample efficiency. The in-distribution performance (solid line) demonstrates a steep learning curve, clearing 94 % F1-score with only 100 samples and steadily improving thereafter. Crucially, the OOD performance (dashed line), a proxy for generalization to unforeseen attacks, initializes at a remarkably high baseline of nearly 80 % F1-score. This indicates that our core approach of modeling the benign manifold confers potent intrinsic robustness, even before extensive exposure to attack patterns.

As the number of malicious samples increases, we observe a slight, transient dip in OOD performance. This is a familiar trade-off: with more labeled attacks, the classifier can lean toward patterns specific to the seen injections, which slightly tightens the decision rule around the training distribution and yields a mild overfitting effect. However, the performance quickly stabilizes to a plateau, consistent with the joint contrastive objective acting as a regularizer that preserves cross-dataset generalization.

Our BERM@300 model, trained on just 300 injection samples, already achieves 82.3 % F1-score overall and 78.2 % on unforeseen datasets (Table 1). This notably represents more than 95 % of the model's eventual OOD performance and exceeds PIGuard [2025], which is trained with 15k injections. This few-shot experiment suggest that BERM can deliver strong detection performance under tight labeling budgets, while retaining its low runtime cost.

4.4 Out-of-Distribution (OOD) Robustness

We stress-test the model's adaptability via a *leave-one-dataset-out* regime, training on a solitary dataset and probing performance on the unforeseens.

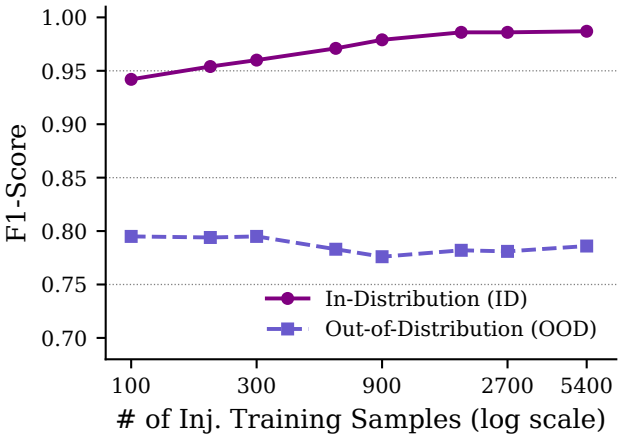

Figure 4: Performance as a function of malicious training sample count. The F1-score is shown for both in-distribution (ID) and out-of-distribution (OOD) test sets. BERM can achieve strong performance with only a few hundred samples. The key takeaway is BERM's high initial OOD robustness and rapid convergence, confirming its exceptional sample efficiency.

Protocol. Under this protocol, a detector is fit on a single corpus and evaluated on (i) on its in-distribution split (IN) and (ii) on all OUT-of-distribution corpora held out from training. We report the same metrics as in §4.1: WIDE-AUROC (AUROC), WIDE-FPR$_{95}$ (FPR$_{95}$), and the downstream F1. Higher AUROC/F1 and lower FPR$_{95}$ indicate stronger robustness. We evaluate BERM (Llama-3.1-8B) against the corpus-specific detectors released with each dataset, ensuring identical training splits.

Findings. BERM exhibits a consistent lead over dataset-specific baselines on held-out corpora, regardless of the training source. Trained on SG, BERM boosts F1 by **+23.1 pp** on average (**+38.5 pp** on DS), while *lowering* FPR$_{95}$. We see the same overall trend when training on JC, even though AUROC on DS drops slightly, BERM still yields **+18.7 pp** higher F1 in aggregate. Such consistent gains validate that BERM's joint objective captures signals that transfer across datasets rather than keying on corpus-specific artifacts.

4.5 Ablation Study

To dissect the distinct contribution of each module within the BERM framework, we conduct a rigorous four-level ablation study on our best-performing setting with *Llama-3.1-8B-Instruct* . Table 3 reports the results. For variants with the final classifier ablated, F1 is no longer well-defined; in those cases we rely on WIDE (see Section 3.5) as a consistent and geometrical probe. This allows us to standardize evaluation across all variants using AUROC and FPR$_{95}$ derived from whitened distances. The study systematically dissect BERM to make the contribution of each component explicit.

The Foundation: Raw Representations. We begin with the most basic variant, *(c) Raw LLM representation*, where we apply WIDE directly to the host LLM's raw hidden states without any BERM components. On this raw space, WIDE

Train	Test	Method	AUROC ↑	FPR_{95} ↓	F1 ↑
(a) Train = SafeGuard (SG)					
SG	SG	SafeGuard	0.999	0.000	**0.994**
SG	SG	BERM	0.999	0.000	0.987
SG	DS	SafeGuard	0.721	0.607	0.405
SG	DS	BERM	0.692	**0.571**	**0.790**
SG	PS	SafeGuard	0.702	0.884	0.697
SG	PS	BERM	**0.819**	**0.627**	**0.774**
Unforeseen avg.		SafeGuard	0.712	0.746	0.551
		BERM	**0.756**	**0.599**	**0.782**
(b) Train = Jailbreak Classification (JC)					
JC	JC	ProtectAIv2	0.979	0.070	0.886
JC	JC	BERM	**0.999**	**0.004**	**0.979**
JC	DS	ProtectAIv2	**0.902**	**0.571**	0.537
JC	DS	BERM	0.878	0.679	**0.826**
JC	PS	ProtectAIv2	**0.722**	0.960	0.658
JC	PS	BERM	0.675	**0.806**	**0.742**
Unforeseen avg.		ProtectAIv2	**0.812**	0.766	0.598
		BERM	0.776	**0.742**	**0.784**

Table 2: Out-of-Distribution robustness. Upper block: models trained on SafeGuard (SG) and evaluated on DeepSet (DS) and Prompt-Injection-Safety (PS). Lower block: models trained on Jailbreak Classification (JC) and evaluated on DS and PS. BERM uses Llama-3.1-8B; SafeGuard and ProtectAIv2 are the respective baselines. Higher AUROC/F1 and lower FPR_{95} indicate better detection.

Variant	GMM	C.L.	Cls.	AUROC ↑	FPR_{95} ↓
Full BERM	✓	✓	✓	0.999	0.000
(a) w/o classifier	✓	✓	×	0.968	0.197
(b) GMM only	✓	×	×	0.722	0.647
(c) Raw LLM repr.	×	×	×	0.590	0.851

Table 3: Ablation study on the components of BERM with a Llama-3.1-8B host. Columns indicate enabling (✓) or disabling (×) of the GMM pre-clustering, the joint contrastive learning (C.L.) projector, and the final classifier (Cls.). Performance drops monotonically as we strip away components, underscoring the contribution of each stage.

yields an AUROC of just 0.590, barely above random chance. In other words, while the vanilla representation geometry may carry some signal, it does not separate benign and injected prompts in a way that makes detection straightforward.

First Layer of Gain: Unsupervised Clustering. Our first addition, *(b) GMM only*, applies Gaussian Mixture Model clustering to the raw representations prior to scoring. This simple, unsupervised step of identifying and separating submanifolds within the benign distribution provides a notable performance lift. The AUROC improves by $+0.132$ to 0.722. This suggests that benign prompts exhibit heterogeneous structure in the representation space, and modeling this structure with clusters can improve separability.

The Core Contribution: joint contrastive learning. The most significant performance leap occurs when we introduce our core innovation, the joint contrastive learning projector. In configuration *(a) w/o Classifier*, the projector is optimized using both the unsupervised cluster-contrastive and supervised ID–OOD contrastive terms. This step dramatically transforms the representation space, catapulting the AUROC from 0.722 to 0.968–a gain of $+0.246$. This demonstrates that the contrastive learning objective is the dominant factor in BERM's success: it actively sculpts the embedding space, projecting the entangled raw representations into a highly structured manifold where the distance between benign and malicious clusters is maximised.

Final Polish: The Classification Head. Finally, the *Full BERM* model adds the binary classification head supervised by a cross-entropy loss. While the primary goal of this component is to provide a direct classification output aligned with the prompt-injection task, its training signal also sharpens the learned representations. This brings the AUROC to a near-perfect 0.999 and the FPR_{95} to 0.000, representing a final gain of $+0.031$ in AUROC over configuration (a).

Overall Impact. Starting from the nearly inseparable raw representations of a LLM (AUROC 0.590), BERM's components progressively structure and separate the representation space. The final framework achieves an overall improvement of $+0.409$ in AUROC and -0.851 in FPR_{95}, projecting seemingly elusive LLM representations into separable, robust detection signals. Across these ablations, performance degrades monotonically as we remove components, underscoring that each stage contributes to the final separability observed in Table 3.

5 Discussion

Robust AI safety defenses can emerge from learning better representations. The large performance gaps in our ablation study (Table 3) show that BERM's gain comes primarily from learning a contrastive projector that re-embeds the LLM representations into a space where benign manifolds and malicious prompts become easier to separate. Besides being robust and symbiotically efficient, BERM integrates cleanly into existing inference stacks. It reuses hidden states already produced during prefill and avoids an auxiliary guard model call. Our future work will expand evaluation to broader real-world settings to validate BERM's effectiveness in these scenarios. We are committed to accelerate the adoption of efficient, representation-based defenses in the broader academic and industrial communities.

6 Conclusion

In this work we present BERM, a light-weight framework that detects prompt-injection attacks by modeling the intrinsic geometry of a host LLM's benign in-situ representations. BERM achieves state-of-the-art accuracy and out-of-distribution robustness while reducing computational cost to a level suitable for latency-sensitive deployments. Looking ahead, these findings suggest that in-situ representation modeling is a promising building block for LLM safety, especially as models and attack surfaces continue to evolve.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62276260 and the National Key Research and Development Program of China (No. 2024YFE0210600).

References

[Blueteam AI, 2024] Blueteam AI. fmops/distilbert-prompt-injection. https://huggingface.co/fmops/distilbert-prompt-injection, 2024.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2025] Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2383–2400, Seattle, WA, August 2025. USENIX Association.

[Deepset, 2024] Deepset. Deepset prompt injection dataset. https://huggingface.co/datasets/deepset/prompt-injections, 2024.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[Ethayarajh, 2019] Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics.

[Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[Greshake *et al.*, 2023] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. More than you've asked for: A comprehensive analysis of novel prompt injection threats to application-integrated large language models. *arXiv preprint arXiv:2302.12173*, 27, 2023.

[He *et al.*, 2021] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.

[Hendrycks *et al.*, 2019] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations*, 2019.

[Hendrycks *et al.*, 2020] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*, 2020.

[Jayavibhavnk, 2024] Jayavibhavnk. Prompt injection safety. https://huggingface.co/datasets/jayavibhav/prompt-injection-safety, 2024.

[Jiang *et al.*, 2023] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. arXiv: 2310.06825.

[Li *et al.*, 2025] Hao Li, Xiaogeng Liu, Ning Zhang, and Chaowei Xiao. PIGuard: Prompt injection guardrail via mitigating overdefense for free. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*, pages 30420–30437, Vienna, Austria, 2025. Association for Computational Linguistics.

[Liu *et al.*, 2024] Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. Automatic and universal prompt injection attacks against large language models. *arXiv preprint arXiv:2403.04957*, 2024.

[Meta AI, 2024] Meta AI. Introducing Llama 3.1: Our Most Capable Models to Date. Blog post, July 2024. Accessed 29 Jun 2026.

[Meta, 2024] Meta. Promptguard: Prompt injection guardrail. https://www.llama.com/docs/model-cards-and-prompt-formats/prompt-guard/, 2024.

[OWASP, 2023] Top OWASP. Owasp top 10 for large language model applications, 2023.

[Perez and Ribeiro, 2022] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

[Phute *et al.*, 2024] Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. Llm self defense: By self examination, llms know they are being tricked, 2024.

[Piet *et al.*, 2024] Julien Piet, Maha Alrashed, Chawin Sitawarin, Sizhe Chen, Zeming Wei, Elizabeth Sun, Basel Alomair, and David Wagner. Jatmo: Prompt injection defense by task-specific finetuning. In *European Symposium on Research in Computer Security*, pages 105–124. Springer, 2024.

[ProtectAI.com., 2024] ProtectAI.com. Fine-tuned deberta-v3-base for prompt injection detection. https://huggingface.co/ProtectAI/deberta-v3-base-prompt-injection-v2, 2024.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[Shang *et al.*, 2023] Chuyi Shang, Aryan Goyal, Lutfi Eren Erdogan, and Siddarth Ijju. Safeguard: A benchmark suite for evaluating attacks and defenses on llm safety. https://devpost.com/software/safeguard-a1hfp4, 2023.

[Shen *et al.*, 2024] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "Do Anything Now": Characterizing and Evaluating In-the-Wild Jailbreak Prompts on Large Language Models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1671–1685, Salt Lake City, UT, USA, 2024. Association for Computing Machinery.

[Suo, 2024] Xuchen Suo. Signed-prompt: A new approach to prevent prompt injection attacks against llm-integrated applications. In *AIP Conference Proceedings*, volume 3194, page 040013. AIP Publishing LLC, 2024.

[van den Oord *et al.*, 2018] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[Vassilev *et al.*, 2025] Aleksandr Vassilev, Alina Oprea, Andrew Fordyce, Hyrum Anderson, Xavier Davies, and Michael Hamin. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. NIST Trustworthy and Responsible AI NIST AI 100-2e2025, National Institute of Standards and Technology, Gaithersburg, MD, 2025.

[Wallace *et al.*, 2024] Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

[Wan *et al.*, 2025] Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. Qwenlong-l1: Towards long-context large reasoning models with reinforcement learning, 2025.

[Wang *et al.*, 2025] Yiming Wang, Pei Zhang, Baosong Yang, Derek F. Wong, Zhuosheng Zhang, and Rui Wang. Embedding trajectory for out-of-distribution detection in mathematical reasoning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS 2024)*, pages 42965–42999, Red Hook, NY, USA, 2025. Curran Associates Inc.

[Willison, 2023] Simon Willison. Delimiters won't save you from prompt injection. https://simonwillison.net/2023/May/11/delimiters-wont-save-you, 2023.

[Xie *et al.*, 2023] Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496, 2023.

[Yi *et al.*, 2025] Jingwei Yi, Yueqi Xie, Bin Zhu, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and defending against indirect prompt injection attacks on large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1809–1820, 2025.

[Zhou *et al.*, 2021] Wenxuan Zhou, Fangyu Liu, and Muhao Chen. Contrastive out-of-distribution detection for pre-trained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.