Quantifying Large Language Model Attacks Through the Lens of Model Cognition

Xiuming Liu[1]*, Chaoxiang He[1]*, Xuanran Yu[1], Jichen Chai[1], Feiyue Xu[1], Sheng Hang[1], Hanqing Hu[1], Bin Benjamin Zhu[2], Hongsheng Hu[1], Shi-Feng Sun[1], Dawu Gu[1], Shuo Wang[1]

[1]*Shanghai Jiao Tong University, Shanghai, China*
[2]*Microsoft Corporation, China*

Abstract

Large language models (LLMs) are vulnerable to malicious inputs that elicit harmful content. Current safety mechanisms, such as keyword filters or output moderation, largely ignore internal model dynamics. We show that safety-relevant features correlated with harmful prompting are strongly separable under lightweight probes in intermediate hidden states (up to 99% accuracy) *before* generation, revealing that such features persist internally even when models produce compliant outputs. Leveraging this observation, we introduce layer-wise toxicity probes and a multi-layer complementary detection framework that fuses signals from diverse depths. Our lightweight Sentinel (<5M parameters) halves false negatives compared to generation-level refusal and maintains over 94% detection accuracy under adversarial attacks—where baselines drop by 32%. Sentinel also outperforms *Llama-Guard-3-8B* on heterogeneous harmful prompting across seven open-weight LLMs (1.5B→72B) and multiple benchmarks (I2P, SneakyPrompt, MMA, Labelled, PIJ, ChatAlpaca, and Multi-turn Jailbreak). Beyond detection, our method provides the first quantitative, layer-resolved map of how safety-relevant signals emerge, propagate, and degrade within LLMs, enabling interpretable, inside-out alignment and diagnostics.

This paper contains potentially sensitive and offensive content, including but not limited to NSFW material, hate speech, discrimination, and other harmful text. Reader discretion is advised.

1 Introduction

Large Language Models (LLMs) now underpin a wide range of security-sensitive applications, ranging from conversational assistants to automated decision support systems. Built on transformer architectures [1] and trained on massive corpora, these models exhibit remarkable fluency and contextual understanding. This expressiveness, however, introduces a fundamental security vulnerability: LLMs are susceptible to adversarial and *jailbreak* prompts [2] that deliberately bypass built-in safety constraints and elicit the generation of harmful, illicit, or extremist content [3]. Notably, such attacks are often initiated through carefully crafted prompts that appear semantically benign or policy-compliant, thereby concealing adversarial goals and evading existing content moderation and safety mechanisms.

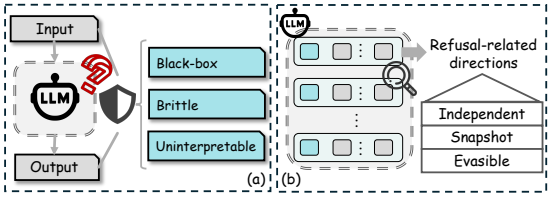


Figure 1: Limitations of existing LLM safety paradigms. (a) Surface-level filters treat LLMs as black boxes, relying on input–output behavior with limited robustness or interpretability under adversarial prompting. (b) Prior internal detectors monitor isolated layers or mechanisms (e.g., refusal states or safety concepts) without modeling how safety-relevant representations propagate and degrade across depth, making them vulnerable to attacks that override late-layer safeguards.

To mitigate these risks, modern LLMs rely on post-training alignment techniques such as supervised fine-tuning (SFT) [4], Reinforcement Learning from Human Feedback (RLHF) [5], and Direct Preference Optimization (DPO) [6]. While these approaches improve average-case safety, they provide limited robustness against adaptive or previously unseen jailbreak prompts. In particular, SFT is brittle under distributional shift beyond its curated demonstrations [7, 8]. RLHF further shapes behavior through preference-based feedback—typically via a learned reward model optimized using reinforcement learning—but remains constrained by feedback coverage and vulnerable to intentionally engineered prompts [9]. DPO eliminates the explicit reward model and reinforcement learning loop by optimizing directly from pref-

*The authors contribute equally to this paper.

erence comparisons; however, recent studies show that it still degrades under systematic adversarial evaluation [10,11]. Collectively, these results indicate that existing alignment methods primarily regulate nominal behavior and do not reliably enforce safety constraints in adversarial settings.

This limitation is reflected in the broader literature on LLM robustness. Most existing robustness evaluations, and many proposed defenses, adopt an output-centric perspective, assessing safety based on whether a model ultimately complies with or refuses a harmful request [12–14]. While effective for measuring externally observable failures, this predominantly black-box view obscures the internal representational dynamics that give rise to model behavior. As illustrated in Fig. 1, existing defenses, whether black-box filters or *single-layer or mechanism-specific internal probes*, fail to capture the continuous evolution of safety-relevant features, rendering them brittle against adaptive attacks. In transformer-based LLMs, safety-relevant features correlated with harmful prompting must be encoded, transformed, and propagated across multiple decoder layers before any output is generated [15]. As representations are iteratively refined across layers, increasingly abstract internal states emerge prior to generation [16]. This raises a central security question: *do jailbreaks succeed because LLMs fail to encode safety-relevant signals at all, or because such signals are present internally but not reliably enforced during decoding?* Distinguishing between these possibilities requires moving beyond output-only observation to examine internal representations directly.

Key insight. We identify a systematic *recognition–enforcement gap* within transformer-based LLMs: under many jailbreak prompts, intermediate hidden representations encode safety-relevant signals correlated with harmful intent, yet this information is not consistently enforced during generation. In particular, such signals often emerge in intermediate decoder layers but are attenuated, overridden, or repurposed in later layers by strong instruction-following and task-completion priors induced during post-training alignment. As a result, jailbreaks frequently arise not from the absence of internal safety cues, but from a failure to translate these cues into safe decoding behavior.

Inside-out safety. Rather than treating safety as an output-side decision, we frame it as an inside-out security problem: analyze how safety-relevant signals correlated with harmful prompting propagate across the transformer stack, track where they emerge, degrade, or are suppressed under adversarial prompting, and repurpose the most informative layers as internal safeguards that can preempt unsafe generation.

Guided by this perspective, we propose a layer-wise probing and fusion framework that both characterizes and exploits internal safety signals. Rather than relying on any single checkpoint, we analyze last-token hidden states across all decoder layers to measure how unsafe-intent information evolves, identify a compact set of complementary layers where such signals are strongest or most stable, and fuse them into an efficient pre-generation guard. This multi-layer design captures signals that arise at different depths and mitigates attacks that selectively target individual internal representations.

Our evaluation demonstrates that this inside-out approach serves a dual role as both a diagnostic lens and a defense mechanism. Across multiple open-weight LLMs and seven public red-teaming benchmarks, the resulting multi-layer sentinel detects jailbreak attempts with minimal latency overhead and maintains strong performance under adaptive attacks where output-level moderation substantially degrades. Beyond mitigation, our analysis provides a layer-resolved view of how safety and instruction-following interact inside modern LLMs, offering empirical evidence that many alignment failures arise from internal enforcement breakdowns rather than an absence of intent recognition.

Contributions. We make the following major contributions:

- **Layer-wise safety probing.** We introduce a layer-wise probing framework that trains lightweight MLP heads on each decoder layer to expose latent unsafe-intent representations and identify where safety recognition first emerges *prior to generation*.
- **Recognition–enforcement analysis under structured adversarial perturbations.** We show that many jailbreaks arise from internal safety signals being overridden during later stages of generation, and systematically characterize how this mismatch evolves across layers under syntactic, semantic, and obfuscation-based perturbations.
- **Complementary multi-layer detection.** We demonstrate that fusing a small set of complementary layers yields a compact, efficient detector that consistently outperforms single-layer and output-only defenses.
- **Robustness to adaptive attacks.** Evaluated on five public red-team benchmarks and under white-box adaptive prompting, our approach maintains $> 94\%$ detection accuracy while baseline moderation drops below 70%.
- **A cognitive map of jailbreak dynamics.** We provide a cross-model analysis of how internal safety signals rise, peak, and decay across the transformer stack, informing future inside-out alignment and security mechanisms.

2 Related Work

Adversarial attacks on LLMs. Early adversarial attacks in NLP primarily targeted classification models, using small token-level perturbations to induce errors in tasks such as sentiment analysis or natural language inference [17]. As LLMs evolved into capable generative systems, adversarial efforts shifted toward attacks that exploit their generative capacity to induce unsafe outputs, including hate speech, illicit

instructions, or privacy violations [18]. A prominent class of such attacks, known as *jailbreaks*, relies on instruction manipulation, role-playing, or obfuscated prompting rather than imperceptible changes [19], exploiting models' instruction-following behavior to evade safety mechanisms.

Interpretability, cognition, and internal representations. Interpretability methods have been widely applied to analyze how transformer models encode linguistic features [20, 21], including attention-based analyses [22] and probing classifiers that localize syntactic and semantic information across layers [23]. Prior work observes a coarse, distributed progression from lexical to semantic features across transformer layers [24], and recent perspectives frame these layers as stages in an implicit, non-anthropomorphic reasoning process where representations are progressively refined [25, 26]. However, how such internal representational dynamics behave under adversarial or safety-critical prompting remains largely unexplored.

Internal-representation-based detection. Several recent defenses leverage internal representations of LLMs to detect malicious or jailbreak prompts, most notably HiddenDetect [27] and JBShield [28]. HiddenDetect monitors refusal-related directions in hidden states, assuming jailbreaks succeed by suppressing explicit refusal mechanisms; this can miss attacks that preserve safety-relevant signals while overriding them during later stages of generation. JBShield detects activated safety concepts within individual layers, but largely treats layers as independent detection points and does not model how safety-relevant information propagates or degrades across depth, leaving it vulnerable to layer-specific evasion under adaptive attacks.

In contrast, our approach explicitly models the layer-wise trajectories of safety-relevant features correlated with harmful prompting across the transformer stack. Rather than relying on a single refusal direction or concept activation, we quantify how harmful and adversarial prompts drift relative to benign baselines across layers and fuse complementary signals from multiple depths. This framing characterizes jailbreaks as enforcement failures—where latent safety-related features are present but not reliably enforced during decoding—and motivates a multi-layer sentinel that improves robustness under both black-box and adaptive white-box attacks.

Research gap. Most existing robustness evaluations and jailbreak defenses adopt an output-centric, black-box perspective, relying on prompt rewriting or post-hoc moderation while ignoring internal representational dynamics [29–34]. Although intermediate transformer layers are known to encode task-relevant signals more distinctly than the final layer [15, 35], how safety-relevant features correlated with harmful prompting arise, propagate, and degrade across layers remains undercharacterized. Moreover, probing methods may recover correlated features rather than signals causally used for generation [36], and recent safety analyses identify refusal-related layers without modeling the full trajectory of signal evolution under attack [37]. Our work addresses this gap by systematically tracing layer-wise representational dynamics of benign and harmful prompts and leveraging distributed internal signals to build more robust, representation-level defenses.

3 Methodology

3.1 Rationale and Design Principles

Our methodology is grounded in a conservative premise: transformer models encode safety-relevant features correlated with harmful prompting in their internal representations, and the evolution of these features across layers provides diagnostic signals about jailbreak behavior. We operationalize this premise by (*i*) extracting last-token representations at each layer, (*ii*) training low-capacity, per-layer probes to assess separability between benign and harmful prompts, (*iii*) tracking how adversarial perturbations alter these representations across depth, and (*iv*) fusing a compact, non-redundant set of layers into a multi-layer sentinel for detection. The corresponding design principles are as follows:

- **Where to look: intermediate-layer safety signals.** Transformer stacks exhibit a coarse, distributed hierarchy: lower layers emphasize lexical cues, intermediate layers consolidate semantic and contextual information, and upper layers are increasingly shaped by next-token prediction [36, 38]. Prior alignment analyses further suggest that safety-related behavior is associated with a contiguous band of intermediate layers [37]. These observations motivate our layer-wise analysis and our focus on intermediate depths, where safety-relevant features are most distinguishable while remaining less entangled with decoding-specific behavior.
- **What to read: last-token hidden states.** We extract the last-token hidden state at each layer as a compact summary of the full prompt processed so far. This choice aligns with auto-regressive decoding—where the next token is conditioned on the final position—and is length-agnostic, avoiding heuristic pooling schemes. Empirically, we find that these representations are sufficient to capture safety-relevant differences even for multi-turn or instruction-suffixed prompts, as the last position integrates the entire preceding context through self-attention.
- **How to test: low-capacity probing for separability.** Intermediate-layer representations often provide a favorable trade-off between compressing surface variation and preserving task-relevant structure [15]. For each layer, we attach a lightweight two-layer MLP probe to evaluate how well benign and harmful prompts can be distinguished by a simple classifier operating on that representation space. We use AUROC as a threshold-free measure of probe separability to identify layers that encode strong safety-relevant signals, without claiming that these features are causally used during generation.

- **How to summarize perturbation effects: drift and ACI.** To quantify how adversarial perturbations affect internal representations, we measure cosine similarity between original and perturbed prompts at each layer. Cosine distance is scale-invariant and comparable across depth, enabling model-wide drift analysis. We further aggregate these layer-wise drifts into a single summary statistic, the Attack Consistency Index (ACI; Section 3.3), which serves as a diagnostic measure of how aggressively a perturbation alters internal representations, rather than as a theoretical bound on attack strength.
- **How to deploy: a compact multi-layer sentinel.** Adjacent transformer layers are often highly redundant due to residual connections [39], while non-adjacent layers can encode complementary information [40]. We therefore prune redundant layers using inter-layer similarity and fuse probes from a small, diverse subset into a low-capacity classifier. This multi-layer sentinel aggregates distributed safety-relevant signals across depth, yielding an efficient and robust detection mechanism.

3.2 Problem Formulation

In this paper, we consider each layer ℓ of a decoder-only transformer comprises (*i*) multi-head self-attention (MHSA), (*ii*) a position-wise feed-forward network (FFN), and (*iii*) two residual–layer-norm wrappers. Denoting the input representation of layer ℓ by $\mathbf{X}^{(\ell-1)} \in \mathbb{R}^{n\times d}$, the forward pass proceeds as

$$\tilde{\mathbf{X}} = \mathbf{X}^{(\ell-1)} + \mathrm{LN}_1^{(\ell)}\big(\mathrm{MHSA}(\mathbf{X}^{(\ell-1)})\big), \tag{1}$$

$$\mathbf{X}^{(\ell)} = \tilde{\mathbf{X}} + \mathrm{LN}_2^{(\ell)}\big(\mathrm{FFN}(\tilde{\mathbf{X}})\big), \tag{2}$$

where $\mathrm{LN}(\cdot)$ is layer normalization, $\mathrm{MHSA}(\cdot)$ aggregates contextual information via scaled dot-product attention, and $\mathrm{FFN}(\cdot)$ applies a position-wise 2-layer MLP with nonlinearity. Equation (2) yields the full representation $\mathbf{X}^{(\ell)}$. We extract its last-token row,

$$\mathbf{h}^{(\ell)} = \mathbf{X}_n^{(\ell)} \in \mathbb{R}^d, \tag{3}$$

as the summary of how the prompt has been integrated up to layer ℓ. Stacking $\{\mathbf{h}^{(\ell)}\}_{\ell=1}^{L}$ gives the trajectory of internal cognition that our probes analyse, where L denotes the total number of hidden layers.

To systematically analyze the model's internal safety mechanisms, we construct two distinct types of prompt sets: (*i*) Training pairs $(P_{\text{benign}}, P_{\text{harmful}})$. Here, P_{harmful} denotes malicious queries explicitly targeting prohibited topics (e.g., illegal acts, hate speech), while P_{benign} consists of standard, safe user instructions. These pairs serve as the labeled data for supervision, enabling us to train layer-wise binary probes that map the linear decision boundary between safe and unsafe representational subspaces. (*ii*) Evaluation pairs $(P_{\text{ori}}, P_{\text{per}})$. In this setting, P_{ori} represents an original harmful prompt that a safety-aligned model successfully refuses. P_{per} is its perturbed variant, generated via adversarial attacks or semantic rephrasing, intended to induce a harmful completion. These pairs are crucial for *inference-time analysis*, allowing us to track how input perturbations shift the model's internal trajectory from a "refusal" state to a "compliance" state.

3.3 Framework Design and Components

Our framework comprises four components: (P1) layer-wise safety profiling; (P2) representational drift measurement; (P3) attack capability quantification; and (P4) sentinel construction. Fig. 2 summarizes the workflow.

P1: Layer-wise safety profiling. Given labeled benign and harmful prompt sets $(P_{\text{benign}}, P_{\text{harmful}})$, we extract last-token hidden states $\{\mathbf{h}^{(\ell)}\}_{\ell=1}^{L}$ and train a lightweight MLP probe for each layer. Layers achieving high AUROC indicate strong local separability between benign and harmful prompts, providing a diagnostic signal of safety-relevant feature encoding rather than evidence of causal intent recognition.

P2: Representational drift measurement. For original–perturbed prompt pairs $(P_{\text{ori}}, P_{\text{per}})$, we compute cosine similarity between their layer-wise hidden states, yielding a consistency profile that highlights where internal representations diverge under adversarial perturbation.

P3: Attack capability quantification. We aggregate layer-wise divergence into the *Attack Consistency Index (ACI)*, defined as the average per-layer cosine distance. ACI provides a compact, comparative summary of how different attacks alter internal representations and should be interpreted as a diagnostic statistic rather than a formal measure of attack strength.

P4: Sentinel construction. Finally, we select a low-redundancy subset of intermediate layers based on similarity of probe responses across attack strategies and fuse their outputs into a compact logistic-regression sentinel. This design captures complementary safety-relevant signals across depth while remaining efficient for inference-time deployment.

The four parts are detailed as follows:

P1: Layer-wise safety profiling. To quantify layer-wise safety-relevant signals, we construct balanced benign and harmful prompt sets $(P_{\text{benign}}, P_{\text{harmful}})$ and extract their last-token hidden states $\{\mathbf{h}^{(\ell)}\}_{\ell=1}^{L}$. These representations are labeled according to prompt type and used to train, for each layer ℓ, a lightweight two-layer MLP probe p_ℓ that maps $\mathbf{h}^{(\ell)}$ to a binary prediction probability:

$$p_\ell(y \mid \mathbf{h}^{(\ell)}) = \sigma\big(\mathbf{w}_\ell^{(2)\top}\ \mathrm{ReLU}\big(\mathbf{W}_\ell^{(1)}\mathbf{h}^{(\ell)} + \mathbf{b}_\ell^{(1)}\big) + b_\ell^{(2)}\big), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid activation function and y is the label. As this is a classification problem, layers exceeding a high AUROC threshold on a held-out set indicate a robust intrinsic capacity to distinguish harmful from benign inputs.

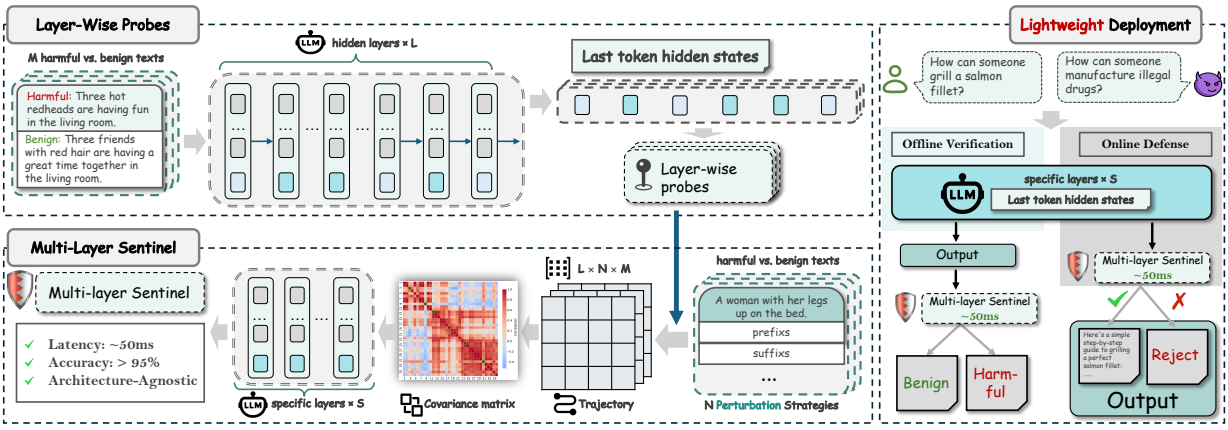


Figure 2: Overview of the Sentinel framework. The training phase (left) uses layer-wise probes to identify depth-specific safety signals and select a non-redundant set of layers for a fused multi-layer sentinel. The deployment phase (right) supports both real-time interception and offline auditing. By reusing the LLM's hidden states, Sentinel incurs zero additional inference passes and negligible latency (∼50 ms).

This profiling process directly addresses our first objective (**O1**): obtaining per-layer safety feedback that reflects the model's internal detection of potential toxicity.

P2: Cognitive drift measurement. Having established a baseline for safety detection, we aim to capture the mechanistic path of a harmful prompt by quantifying how adversarial perturbations shift the model's internal cognition away from its refusal state. First, to evaluate whether a layer maintains its ability to recognize toxicity under perturbation, we utilize the probe outputs and define the Toxicity Detection Rate (TDR) at layer ℓ, denoted as $R_{\text{tox}}^{(\ell)}$, to measure the proportion of harmful inputs correctly identified by the layer's internal state. Given a set of harmful prompts $\mathcal{P}_{\text{harm}}$ and a decision threshold τ (typically set to 0.5), the rate is calculated as:

$$R_{\text{tox}}^{(\ell)} = \frac{1}{|\mathcal{P}_{\text{harm}}|} \sum_{\mathbf{x} \in \mathcal{P}_{\text{harm}}} \mathbb{I}\left(p_\ell(y = \text{harm} \mid \mathbf{h}^{(\ell)}(\mathbf{x})) > \tau \right), \quad (5)$$

where $\mathbb{I}(\cdot)$ is the indicator function. Then, to capture from which layer the internal representations of the harmful input shift from benign ones, for prompt pairs $(P_{\text{ori}}, P_{\text{per}})$, we compute the layer-wise cosine similarity between their hidden states of a specific layer:

$$C(\ell) \;=\; \text{CosSim}\big(\mathbf{h}_{\text{ori}}^{(\ell)}, \mathbf{h}_{\text{per}}^{(\ell)}\big). \quad (6)$$

Aggregating these scores from each layer yields a *consistency profile* $\mathcal{C} = \{C(1), \ldots, C(L)\}$ of a harmful prompt. Lower values in this profile pinpoint the exact depths where the perturbed representation diverges from the original refusal trajectory. This measurement fulfills our second objective (**O2**): revealing exactly where and how internal representations of a harmful prompt drift under adversarial pressure.

P3: Attack capability quantification. While layer-wise drift provides fine-grained insights, comparing how different attack methods alter model internals requires a holistic metric. To this end, we aggregate layer-wise drifts into a single scalar metric, the *Attack Consistency Index (ACI)*, to summarize the cumulative impact of a perturbation on the model's latent space. The index is defined as the average divergence across all layers relative to a specific reference baseline:

$$\text{ACI} \;=\; \frac{1}{L} \sum_{\ell=1}^{L} \big(1 - C(\ell)\big). \quad (7)$$

Crucially, the magnitude of ACI serves as a diagnostic proxy for the attack's operational mode rather than a simple measure of strength. A lower ACI(relative to the original harmful contents) indicates the strict preservation of malicious intent necessary to elicit harmful responses, whereas a higher ACI signifies brute-force disruption. This quantification fulfills our third objective (**O3**): enabling the standardized comparison of diverse attack strategies via a unified numerical benchmark.

P4: Sentinel construction and deployment. Finally, to translate these interpretability insights into an efficient defense for detecting harmful prompts, we aim to construct a compact monitor that maximizes detection coverage while eliminating redundant computations. To achieve this, we first model the responses of intermediate layers to M distinct harmful strategies (e.g., prompt injection). For each strategy j, the model produces a toxicity trajectory across layers, defined as

$$T_j = [s_{1j}, \ldots, s_{Lj}], \quad (8)$$

where $s_{\ell j}$ is the probe score at layer ℓ. To capture the distinct role of each layer ℓ, we construct a behavior vector

$$B_\ell = [s_{\ell 1}, \ldots, s_{\ell M}]^\top. \quad (9)$$

Optimal subset selection and sentinel training are described in **Algorithm 1**. In Stage 3, we enforce low redundancy by

Algorithm 1 Sentinel Layer Selection and Training

Require: All layers L, Training Data $\mathcal{D}_{tr}$, Validation Strategies $\mathcal{D}_{val}$ (size M), Thresholds τ_{acc}, τ_{corr}
Ensure: Sentinel $\mathcal{P}$, Selected Layers $\mathcal{S}$

Stage 1: Probing & Filtering
Train probes p_l for all $l \in L$ on $\mathcal{D}_{tr}$.
Select candidates $C \leftarrow \{l \in L \mid \text{AUROC}(p_l) > \tau_{acc}\}$, sorted by AUROC desc.
Stage 2: Behavior Profiling
for each $l \in C$ **do**
  Compute behavior vector $B_l \in \mathbb{R}^M$ where $B_l[j] = p_l(\mathcal{D}_{val}^{(j)})$.
end for
Stage 3: Non-Redundant Selection
Initialize $\mathcal{S} \leftarrow \emptyset$.
for each candidate $c \in C$ **do**
  {Check max similarity with already selected layers}
  if $\mathcal{S} = \emptyset$ **or** $\max_{s \in \mathcal{S}} \text{CosSim}(B_c, B_s) < \tau_{corr}$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{c\}$
  end if
end for
Stage 4: Sentinel Fusion
Construct features $H_{fused} = \text{Concat}(\{h^{(l)} \mid l \in \mathcal{S}\})$.
Train classifier $\mathcal{P}$ on H_{fused} to minimize CrossEntropy.
return $\mathcal{P}, \mathcal{S}$

computing cosine similarity between layer behavior vectors $\{B_\ell\}$ and pruning highly correlated layers via greedy selection, yielding a compact yet representative subset. In Stage 4, the *Sentinel* $\mathcal{P}$ is constructed as a logistic regression classifier over the concatenated hidden states of the selected layers. This design enables both efficient prompt detection and fine-grained tracing of where harmful prompts diverge from benign ones, fulfilling our fourth objective (**O4**). Additional justification for the layer selection strategy is provided in Appendix C.

4 Experimental Evaluation

All experiments, including all baselines, were conducted on a single NVIDIA A100 GPU using PyTorch 2.6.0.

Task setting. We study the detection of malicious inputs intended to induce large language models to generate harmful outputs. The task is formulated as a binary classification problem (safe vs. unsafe) using *Sentinel* in a white-box setting, where intermediate hidden states are monitored during inference to enable proactive and interpretable safety assessment.

Target models. To evaluate the generality of the proposed framework, we conduct experiments across a diverse set of LLM families with varying model sizes and architectural designs, as summarized in Table 1. This diversity enables a systematic assessment of scalability and robustness under adversarial prompting across practical deployment scenarios.

Table 1: Target LLMs evaluated under a white-box threat model. The table summarizes model families, parameter scales, layer depths, and developers, enabling assessment of scalability and robustness across diverse architectures.

Model Name	Param.	# of Layers	Developer
Llama-{3.2-3B, 3.1-8B}-Instruct	{3B, 8B}	{28, 32}	Meta
Qwen2.5-{1.5B, 7B, 72B}-Instruct	{1.5B, 7B, 72B}	{28, 28, 80}	Ali Cloud
Qwen3-{4B, 8B}	{4B, 8B}	{36, 36}	Ali Cloud

Datasets. We employ a diverse collection of datasets covering explicit toxicity, adversarially obfuscated prompts, jailbreak attacks, and multi-turn interactions:

- **NSFW-56k [41]:** 56K textual prompts generated from web-crawled explicit images via an image-to-text pipeline, refined with CLIP-based re-ranking.
- **SneakyPrompt [42]:** 200 adversarial NSFW prompts generated using reinforcement learning to replace sensitive tokens with benign alternatives while preserving malicious goals.
- **I2P [43]:** A benchmark containing 4,700 real-world prompts spanning nudity, violence, hate, and self-harm.
- **MMA [44]:** Prompts modified to bypass NSFW filters while maintaining harmful semantics.
- **Labelled [45]:** 5,350 labeled instances collected from tweets, harmful behavior datasets, and synthetic sources.
- **Prompt Injection Benchmark (PIJ) [46]:** 5,000 jailbreak prompts involving policy-violating and illegal instructions.
- **Multi-turn robustness (ChatAlpaca [47], Multi-turn Jailbreak [48]):** A mixed dataset of benign and adversarial multi-turn dialogues, where harmful intent is distributed across conversation turns.

To evaluate generalization beyond the training distribution, we organize benchmark datasets by evaluation objective: (*i*) Adversarial robustness: SneakyPrompt, I2P, and MMA; (*ii*) Domain generalization: Labelled; (*iii*) Jailbreak resistance: PIJ; (*iv*) Multi-turn robustness: ChatAlpaca and Multi-turn Jailbreak.

Training data. We train layer-wise probes using a balanced binary dataset with an 8:2 train–test split. The harmful class consists of 1,000 prompts sampled from NSFW-56k. For the benign class, we generate 1,000 "home scene" descriptions using OpenAI's `o1` model [49]. These benign prompts are structurally matched to harmful ones to serve as semantic controls (Appendix E), reducing superficial lexical biases.

Evaluation metrics. We report standard binary classification metrics including accuracy, AUROC, precision, recall (TPR), and F1-score, with particular emphasis on TPR at low FPR thresholds (0.1% and 1%) to reflect realistic safety constraints. In addition, we evaluate the proposed *Attack Consistency Index* (ACI), which quantifies internal representational drift and exposes the phenomenon of *adversarial camouflage*: successful attacks retain harmful features while subtly shifting representations toward benign regions to evade detection.

Baselines. We compare Sentinel against both black-box and white-box detection methods. Black-box baselines include in-context *Generative Judgment*, a fine-tuned NSFW text classifier [50], and *Llama-Guard-3-8B* [51]. White-box baselines include *HiddenDetect* [27], *JBShield* [28], and *GradSafe* [52]. All baselines are evaluated using configurations reported in their original works.

Implementation details of our framework. As illustrated in Fig. 2, the pipeline follows a structured life cycle from internal state extraction to cognitive quantification and Sentinel deployment:

Feature extraction. For each benign and harmful prompt, we perform a single forward pass and extract hidden states from all transformer layers. We use the last-token representation as the layer-wise summary, as ablation results (Appendix B) show that mean pooling dilutes safety-relevant signals concentrated near the generation boundary.

Probing & behavior profiling. To measure layer-wise separability between benign and harmful inputs, we train a lightweight two-layer MLP probe (100 hidden units) for each layer using cross-entropy loss with early stopping (maximum 500 epochs). In parallel, we apply a multi-level perturbation suite (Appendix A) spanning element-level insertions, syntactic transformations, and semantic role-playing prompts to elicit diverse internal responses.

Cognitive quantification & selection. The probing responses across perturbation types form layer-specific toxicity trajectories. We summarize representational drift using the Attack Consistency Index (ACI), which aggregates layer-wise cosine distances to quantify internal deviation under attack. Based on correlation between trajectories, redundant layers are pruned to construct a compact and complementary subset.

Sentinel implementation. Finally, we instantiate the Sentinel as a two-layer MLP with 100 hidden units (as selected by Algorithm 1), operating on the concatenated hidden states from the selected layer subset. This design preserves computational efficiency while leveraging distributed, non-linear safety-relevant signals encoded in the model internals.

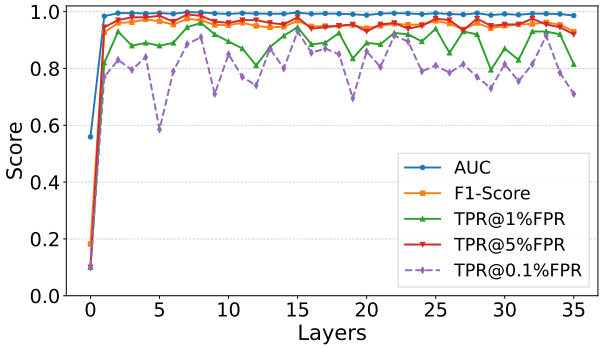


Figure 3: Layer-wise performance of safety probes on Qwen3-4B. Detection capability is minimal at the embedding layer (L0) but emerges sharply at the first transformer layer (L1) and remains strong across subsequent layers, with stability and robustness increasing at greater depths. This indicates that safety-relevant features correlated with harmful prompting arise early and are progressively refined throughout the forward pass.

4.1 Results and Analysis

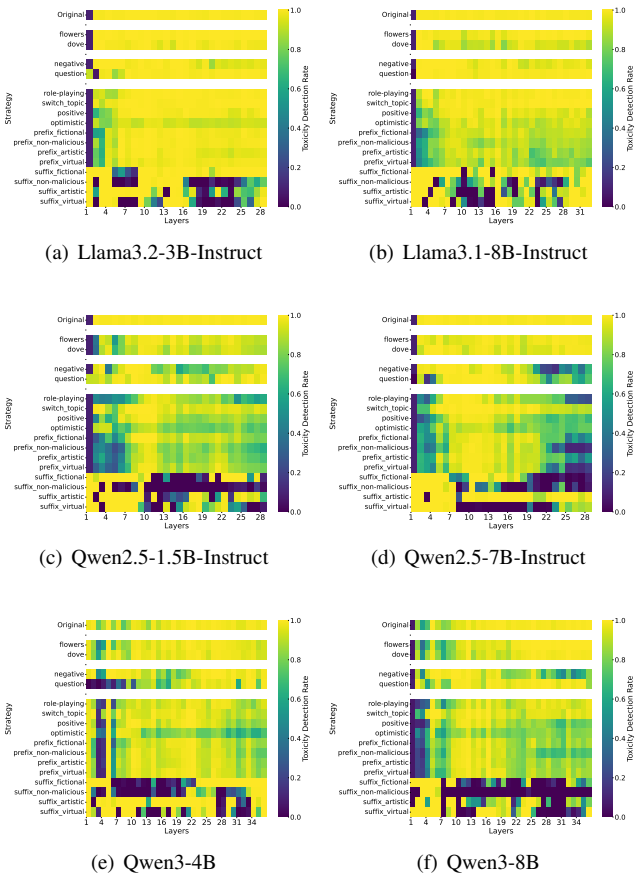
4.1.1 Layer-wise Toxicity Probing

We train safety probes at each layer to examine how safety-relevant features correlated with harmful prompting are encoded throughout the forward pass. As shown in Fig. 3 using Qwen3-4B as a representative example, probe performance indicates that such features are detectable across nearly the entire model depth, with the exception of the initial embedding layer (L0). Detection capability rises sharply at the first transformer layer (L1), establishing strong separability between benign and harmful prompts. Moreover, performance—particularly in terms of robustness and stability—generally improves with increasing depth, achieving near-perfect AUROC scores in deeper layers. These results suggest that safety-relevant signals emerge early in the network and are progressively refined across layers, making intermediate and deep representations reliable sources for detection.

4.1.2 Layer-Wise Cognitive Drift

We analyze the **toxicity detection rate** ($R_{\text{tox}}^{(\ell)}$) across models, layers, and perturbation strategies. The results, shown in Fig. 4 (lighter colors indicate higher $R_{\text{tox}}^{(\ell)}$), reveal systematic but non-uniform variations in layer-wise sensitivity across architectures. Our key observations are as follows:

Distinct effects of perturbation strategies. Different perturbation strategies produce markedly different detection patterns. As shown in Fig. 4, unmodified harmful prompts ("Original") consistently yield high detection rates, whereas perturbed inputs introduce pronounced fluctuations, substantially



(a) Llama3.2-3B-Instruct (b) Llama3.1-8B-Instruct

(c) Qwen2.5-1.5B-Instruct (d) Qwen2.5-7B-Instruct

(e) Qwen3-4B (f) Qwen3-8B

Figure 4: Toxicity detection rate ($R_{\text{tox}}^{(\ell)}$) across models, layers, and perturbation strategies. Heatmaps illustrate how safety-relevant features correlated with harmful prompting are detected at different depths under element-, syntactic-, and semantic-level perturbations. The patterns vary across architectures and model sizes, revealing systematic yet non-uniform layer sensitivities.

degrading detection at specific layers.

Layer-dependent sensitivity. Sensitivity to perturbations varies with depth. In Llama3.1-8B-Instruct (Fig. 4(b)) and Qwen2.5-7B-Instruct (Fig. 4(d)), prefix-based perturbations maintain high detection in early layers but degrade sharply in deeper layers. In contrast, suffix-based perturbations predominantly disrupt detection in intermediate layers, reflecting hierarchical processing effects.

Absence of uniformly robust layers. No single perturbation strategy is consistently detected across all layers. For example, in Fig. 4(e), the `prefix_fictional` attack is strongly detected in early layers but largely missed in the final layers. This non-uniformity underscores the limitations of single-layer defenses and motivates the need for multi-layer fusion.

Consistent trends across model scales. Despite differences in model size, layer-wise detection landscapes remain broadly

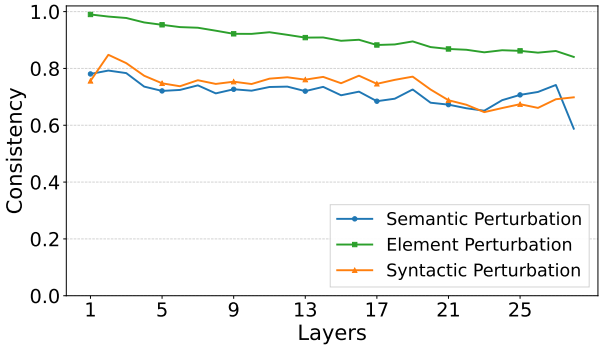


Figure 5: Consistency profiles $\mathcal{C}$ for Qwen2.5-7B-Instruct. Each curve plots the cosine similarity $C(\ell)$ between hidden states of original and perturbed prompts across layers. The profiles exhibit early-layer stratification, a stable mid-layer plateau, and sharp divergence in late layers, revealing where adherence to refusal-aligned representations deteriorates.

similar. Comparing Qwen2.5-1.5B-Instruct (Fig. 4(c)) and Qwen2.5-7B-Instruct (Fig. 4(d)) reveals shared vulnerability patterns, particularly for prefix-based attacks in later layers. This suggests that safety-relevant processing is largely shared across scales, with parameter size primarily affecting sensitivity rather than qualitative behavior.

These observations provide a key insight: although individual layers exhibit distinct vulnerability patterns to specific perturbation strategies, these blind spots are rarely aligned across depth. Thislayer-wise complementarity directly underpins the robustness of our multi-layer design. By fusing signals from diverse layers, the Sentinel mitigates failures in any single layer—for example, late-layer breakdowns under prefix-based attacks are compensated by strong detection in earlier layers. Moreover, the consistency of these patterns across model sizes (from 1.5B to 8B) suggests that such complementarity is a fundamental property of transformer-based LLMs, supporting the scalability and generality of our approach across architectures.

We further analyze layer-wise responses to perturbed inputs across representative models to characterize how adversarial modifications affect internal representations. Fig. 5 visualizes the *consistency profile* $\mathcal{C} = \{C(1), \ldots, C(L)\}$ for sample prompt pairs under different attack strategies, plotting layer-wise cosine similarity $C(\ell)$ between original and perturbed inputs. We use Qwen2.5-7B-Instruct as a representative case.

Early-layer stratification. The consistency profiles exhibit immediate stratification rather than gradual or noisy divergence. Element-level perturbations largely preserve alignment with the original refusal trajectory, whereas syntactic and semantic perturbations induce a stable offset ($C(\ell) \approx 0.7$–0.8) from the earliest layers onward, indicating an early but bounded representational shift.

Mid-layer plateau. Across mid-depth layers (e.g., $\ell = 5$–13),

Table 2: ACI values relative to harmful and benign references across models and perturbation types.

Model	ACI Type	Syntactic-Prefix	Syntactic-Suffix	Semantic	Element
Qwen2.5-7B-Instruct	*harmful*	0.0915	0.4744	0.2620	0.0928
	benign	0.2189	0.6666	0.3253	0.2329
Llama3.1-8B-Instruct	*harmful*	0.0947	0.4692	0.2620	0.0770
	benign	0.2766	0.6724	0.3751	0.2578

$C(\ell)$ remains relatively stable, indicating that the representational distance between perturbed and original prompts does not increase further. This plateau suggests that adversarial inputs establish a fixed internal offset while maintaining partial alignment with refusal-related representations prior to generation.

Late-layer override. For semantic attacks, a pronounced terminal divergence emerges in the final layers ($\ell > 25$), where $C(\ell)$ drops sharply below 0.6. Since $C(\ell)$ measures alignment with the original refusal trajectory, this decline quantifies the point at which the model decouples from safety-aligned representations. This layer-localized breakdown confirms that the decisive override of refusal behavior occurs late in the forward pass.

4.1.3 Quantifying Attack Mechanisms

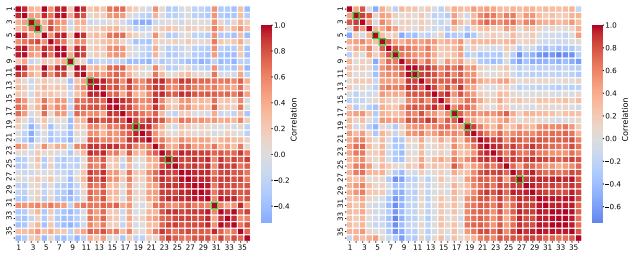
Our layer-wise analysis indicates that adversarial attack behavior is governed by how perturbed prompts reposition internal representations relative to harmful and benign reference states. To make this explicit, we decompose representational drift into two complementary dimensions:

1. **Retention of harm** ($\text{ACI}_{\textbf{harmful}}$)**:** The distance to the representation of the original harmful prompt. Lower values indicate stronger preservation of harmful semantics.
2. **Proximity to safety** ($\text{ACI}_{\textbf{benign}}$)**:** The distance to a semantically matched benign prompt. Lower values indicate closer alignment with the model's benign label manifold.

Table 2 reports these metrics across perturbation types for Qwen2.5-7B-Instruct and Llama3.1-8B-Instruct, revealing two qualitatively distinct attack regimes.

Stealth regime. Highly effective attacks—most notably *syntactic-prefix* and *element* perturbations—exhibit a characteristic geometric profile: very low $\text{ACI}_{\text{harmful}}$ (< 0.1) combined with moderately low $\text{ACI}_{\text{benign}}$ (0.21–0.27). These perturbations preserve alignment with harmful reference representations while simultaneously shifting toward regions typically associated with benign inputs. This pattern substantiates the *adversarial camouflage* phenomenon: harmful content is embedded within representations that remain structurally plausible, avoiding rejection while retaining attack efficacy.

Disruption regime. In contrast, *syntactic-suffix* perturbations yield substantially higher values for both metrics (e.g., $\text{ACI}_{\text{harmful}} \approx 0.47$ and $\text{ACI}_{\text{benign}} \approx 0.67$ on Qwen). These attacks introduce broad representational distortion, pushing hidden states away from both harmful and benign baselines. Although such disruption can occasionally bypass safeguards via confusion, it lacks the precision of stealth-based strategies and produces large, easily detectable deviations.



(a) Qwen3-4B (b) Qwen3-8B

Figure 6: Layer-wise correlation heatmaps of probe scores across intermediate layers for Qwen3-4B and Qwen3-8B. Strong diagonal similarity indicates redundancy among adjacent layers, while off-diagonal variation reveals complementary signals. Green markers denote the selected Sentinel layers that balance representational diversity and efficiency.

Implications for defense. This dichotomy has direct implications for detection. Threshold-based defenses are effective against disruptive attacks with large representational drift but are liable to miss stealthy perturbations that remain close to harmful references while superficially resembling benign inputs. The tight clustering of high-risk attacks near the harmful baseline highlights the necessity of multi-layer analysis: only by aggregating complementary signals across depth can subtle geometric inconsistencies introduced by adversarial camouflage be reliably detected.

4.1.4 Multi-Layer Sentinel Evaluation

To evaluate the effectiveness of the proposed Sentinel, we examine how internal layer responses vary across diverse harmful strategies. For each layer, we construct a behavior profile by aggregating its toxicity detection rate (TDR) across perturbation types, and compute pairwise cosine similarity to quantify functional alignment. Fig. 6 shows the resulting similarity heatmaps for Qwen3-4B and Qwen3-8B. Both models exhibit strong high-similarity bands along the diagonal, indicating substantial redundancy among adjacent layers and suggesting that internal representations evolve gradually rather than via abrupt transitions. Using Algorithm 1, we iteratively select layers with high separability and low inter-layer correlation; the green markers denote the resulting non-redundant layer subset $\mathcal{S}$, which maximizes behavioral diversity while minimizing computational overhead.

Table 3: Toxic prompt detection performance (Accuracy / AUROC) of Sentinel compared with black-box baselines, including model-internal generative judgment (*Judge*) and external classifiers (NSFW Text Classifier and Llama-Guard-3-8B). Sentinel's use of internal representations yields consistently higher performance than input- or output-level filters.

Models	Methods	Accuracy↑ / AUC↑				Average Accuracy↑ / AUC↑
		Sneaky	I2P	MMA	Labelled	
Qwen2.5-1.5B-Instruct	Judge	0.850 / 0.850	0.396 / 0.633	0.840 / 0.860	0.798 / 0.727	0.721 / 0.768
(1,3,6,10,12,19,21)	**Ours**	0.948 / 0.948	**0.970** / 0.970	0.973 / 0.957	0.839 / 0.723	0.933 / 0.900
Qwen2.5-7B-Instruct	Judge	0.798 / 0.798	0.281 / 0.563	0.765 / 0.765	0.743 / **0.855**	0.647 / 0.745
(1,3,6,8,14,20)	**Ours**	0.943 / 0.943	0.880 / 0.880	0.959 / 0.936	0.837 / 0.751	0.905 / 0.878
Qwen3-4B	Judge	0.874 / 0.018	0.310 / 0.193	0.804 / 0.027	**0.871** / 0.847	0.715 / 0.271
(3,4,9,12,19,24,31)	**Ours**	0.932 / **1.000**	0.886 / **1.000**	**0.995** / **1.000**	0.844 / 0.601	0.914 / 0.900
Qwen3-8B	Judge	0.856 / 0.005	0.344 / 0.015	0.818 / 0.030	0.820 / 0.810	0.710 / 0.215
(1,4,7,12,16,21,28,32)	**Ours**	0.954 / **1.000**	0.957 / **1.000**	0.993 / **1.000**	0.845 / 0.714	**0.937** / **0.929**
Llama3.2-3B-Instruct	Judge	0.833 / 0.853	0.750 / 0.848	0.846 / 0.832	0.817 / 0.672	0.812 / 0.801
(1,2,3,6,20)	**Ours**	0.950 / 0.950	0.943 / 0.943	0.973 / 0.957	0.844 / 0.756	0.928 / 0.902
Llama3.1-8B-Instruct	Judge	0.883 / 0.855	0.734 / 0.838	0.815 / 0.915	0.827 / 0.778	0.815 / 0.847
(1,6,9,11,20)	**Ours**	**0.963** / 0.963	0.965 / 0.965	0.976 / 0.962	0.843 / 0.783	**0.937** / 0.918
NSFW Text Classifier [50]		0.892 / 0.904	0.626 / 0.655	0.942 / 0.971	0.837 / 0.843	0.824 / 0.843
Llama-Guard-3-8B [51]		0.826 / 0.783	0.518 / 0.681	0.610 / 0.747	0.379 / 0.606	0.583 / 0.704

Table 4: Detection accuracy comparison between the Multi-Layer Sentinel and other white-box baselines (HiddenDetect, JBShield, and GradSafe) on the Labelled and PIJ benchmarks. The Multi-Layer Sentinel consistently outperforms existing white-box approaches across both datasets.

Dataset	Ours	HiddenDetect	JBShield	GradSafe
Labelled	**0.8434**	0.7385	0.8173	0.3219
PIJ	**0.9117**	0.6667	0.7183	0.6650

The selected layers also provide interpretable depth-wise specialization. As illustrated in Fig. 4(f), earlier layers (e.g., *L*5) are more sensitive to syntactic perturbations such as question reformulations, whereas later layers (e.g., *L*31) respond more strongly to high-level semantic deviations, including role-playing attacks. This distribution aligns with the hierarchical processing structure of transformer models, where superficial anomalies are addressed early and policy-level violations manifest at greater depth.

Based on the selected layers, we construct the Sentinel as a multi-layer detector that operates on concatenated internal representations. Table 3 compares Sentinel against black-box baselines. Sentinel consistently outperforms the model's own generative judgment across all benchmarks, with particularly large gains on stealthy attacks such as I2P, where generative judgment fails on the majority of harmful prompts. Across different backbone models, Sentinel maintains strong detection performance, achieving an average AUROC above 0.87 on both Qwen and Llama families. Compared to specialized safety classifiers—NSFW Text Classifier and Llama-Guard-3-8B—Sentinel achieves over 0.90 average accuracy, outperforming their sub-0.83 accuracy while using fewer than 5 M parameters and requiring no additional LLM decoding. Full results are reported in Appendix F.

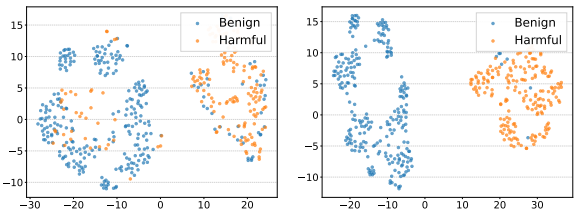
Given its white-box design, we further compare Sentinel against white-box baselines, including HiddenDetect, JBShield, and GradSafe. As shown in Table 4, Sentinel substantially outperforms all alternatives. Notably, on the PIJ benchmark, Sentinel achieves 0.91 accuracy, exceeding JBShield by nearly 20%. This result highlights the advantage of multi-layer signal fusion over defenses based on single-layer concepts or gradient patterns.

Beyond single-turn prompts, adversarial intent in real-world settings often unfolds across multiple interactions. Without additional training, we evaluate Sentinel on the Multi-turn Robustness benchmark. Sentinel achieves 97.0% accuracy in distinguishing distributed harmful intent from benign follow-up instructions, demonstrating that safety-relevant signals persist across turns and can be effectively aggregated across layers.

Finally, to assess interpretability, we visualize internal representations using two-dimensional t-SNE projections. For a strong single-layer baseline (Layer 20), Fig. 7(a) shows substantial overlap between safe and unsafe prompts, indicating limited discriminative capacity in isolation. In contrast, the multi-layer representation (Fig. 7(b)) exhibits clear separation by aggregating complementary signals across depth. These visualizations reinforce that Sentinel leverages diverse internal features—spanning syntactic irregularities and semantic policy violations—enabling more transparent and reliable detection.

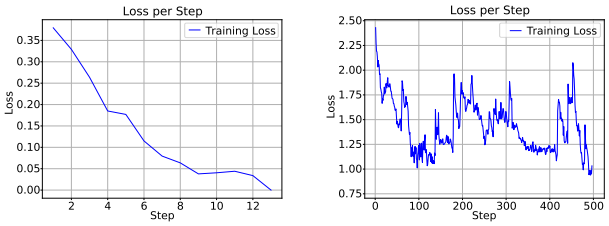
4.2 Adaptive Attacker

We consider a strong adaptive threat model in which the adversary has full knowledge of the Sentinel defense, including the selected sentinel layers, probe parameters, and detection thresholds. Under this white-box setting, the attacker explicitly optimizes malicious inputs to evade detection while still



(a) t-SNE Single-Layer (b) t-SNE Multi-Layer

Figure 7: t-SNE visualization of single-layer versus multi-layer probing on Qwen2.5-7B-Instruct. Single-layer representations exhibit substantial overlap between safe and unsafe prompts, whereas multi-layer aggregation yields clear separation, illustrating the improved discriminative capacity and interpretability of the Sentinel framework.



(a) Adaptive Single-Layer (b) Adaptive Multi-Layer

Figure 8: Adaptive attacks against (a) a single-layer probing classifier and (b) the proposed multi-layer Sentinel. Single-layer probes rapidly degrade under optimization, whereas multi-layer fusion resists white-box attacks due to conflicting gradients across layers, substantially reducing attack success rates.

eliciting harmful model behavior.

We implement a gradient-based adaptive attack inspired by the Greedy Coordinate Gradient (GCG) algorithm [53]. The goal is to automatically construct adversarial suffixes that cause Sentinel to misclassify harmful prompts as benign. The suffix is initialized as a neutral, high-entropy string ("!!!!!!!!!!!!!!!") and appended to the original harmful prompt. At each optimization step, the concatenated input is processed by the target LLM, and the pre-trained layer-wise probes p_ℓ (Section 3) compute harmfulness probabilities based on last-token hidden states. We define the loss as

$$\ell_{\text{loss}} = \begin{cases} P - 0.5, & P > 0.5, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

which encourages the optimized input to cross the Sentinel's decision threshold. Gradients are backpropagated through one-hot token representations to guide discrete token selection. For each suffix position, we compute gradients over the vocabulary, select the top-$k = 256$ tokens with the most negative gradients, generate $n = 512$ candidate suffixes, and retain the best-performing candidate for the next iteration. The process runs for up to 500 optimization steps.

Table 5: Computational cost of Sentinel, measured by wall-clock time, including representation extraction and binary classifier training. The end-to-end overhead remains modest even for large-scale models such as Qwen2.5-72B-Instruct.

Model	**Extraction Time (min)**	**Training Time (min)**	**Storage Usage (GB)**
Qwen3-8B	2.05	3.06	1.38
Qwen2.5-72B-Instruct	8.37	10.84	6.10

We evaluate adaptive attacks under two configurations: (*i*) a degenerate Sentinel using a single probing layer, and (*ii*) the full multi-layer Sentinel. When attacking the single-layer variant, the loss is computed only at the target layer. As shown in Fig. 8(a), the attack succeeds in all cases, requiring an average of only 13.1 optimization steps. This result highlights the fragility of single-layer defenses: once a favorable gradient direction is identified, the detector collapses rapidly.

In contrast, attacking the multi-layer Sentinel (Fig. 8(b)) requires minimizing the loss simultaneously across all selected layers. This induces unstable optimization dynamics characterized by fluctuating gradients and early stagnation. The attack success rate increases from 17% to at most 58% and fails to converge further. Gradient analysis indicates conflicting descent directions across layers: perturbations that reduce harmfulness scores in early syntactic layers often amplify signals in deeper semantic layers, preventing a unified update that satisfies all detection constraints.

We further evaluate a stricter adaptive scenario in which the adversary attempts to suppress not only detection scores but also internal signals of representational drift. To this end, we augment the objective with an auxiliary *consistency loss* that penalizes cosine distance between adversarial hidden states and the original refusal trajectory across all sentinel layers. This forces the attack to induce harmful generation while remaining internally consistent with refusal-aligned representations.

As shown in Fig. 9, this enhanced attack consistently fails. The added constraint severely restricts the optimization space, creating an inherent tension: eliciting harmful completions requires deviation from refusal-aligned representations, while the consistency loss penalizes precisely this deviation. As a result, optimization becomes unstable and fails to converge to a successful evasion. These results demonstrate that the multi-layer Sentinel imposes incompatible constraints on adaptive attackers, effectively rendering white-box optimization infeasible.

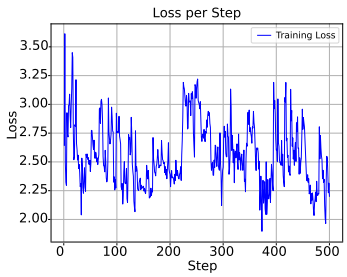


Figure 9: Training loss trajectory under an enhanced adaptive attack with inter-layer consistency constraints. Despite the additional guidance, the optimization fails to converge and exhibits pronounced oscillations, indicating that the Sentinel's multi-layer constraints severely restrict the adversarial search space.

4.3 Efficiency

To characterize the computational overhead of Sentinel, we evaluate its efficiency using Qwen2.5-72B-Instruct as a representative large-scale model. As shown in Table 5, Sentinel introduces negligible inference overhead, enabling real-time safety assessment with approximately 50 ms latency per prompt. The training overhead is similarly modest: even for a 72B-parameter model, representation extraction and classifier training complete within minutes and require only modest storage, demonstrating the practicality of Sentinel for large-scale deployment.

5 Discussion

Mid-depth layers encode safety-relevant features associated with harmful prompting. Across Qwen, Llama, and other model families, we observe a consistent pattern: last-token hidden states in intermediate transformer layers (roughly 50–70% of the stack) are strongly separable between prompts labeled as harmful and benign, often achieving AUROC values above 0.9. This separability persists even under sophisticated jailbreak wrappers; for example, probes on Llama3.1-8B-Instruct flag 91.1% of PIJ prompts as unsafe. Importantly, this does not imply that the model semantically "understands" malicious intent. Rather, it indicates that safety-relevant features correlated with harmful prompting are encoded in internal representations but are not reliably enforced during decoding. Mechanistically, this observation is consistent with prior work showing that refusal behavior corresponds to a low-dimensional direction in residual space [54], suggesting that unsafe-associated features remain internally accessible even when surface-level refusals fail.

Why late layers override earlier safety signals. Although modern LLMs undergo extensive alignment, harmful content can still be generated at decoding time. Under the theory of subsequence associations [55], generation favors statistically dominant continuations. If pre-training data contain far more step-by-step solutions than explicit refusals, unsafe continuations can dominate at inference. Adversarial prompt wrappers further exacerbate this effect by reallocating attention in higher layers toward benign or role-play text, attenuating late-stage refusal cues. Our perturbation ablation results show that semantic wrappers reduce generation-level refusals by 41%, while mid-layer probe performance drops by only 6%, indicating that alignment mechanisms introduced by DPO or RLHF primarily operate in late layers and can be overridden without eliminating earlier safety-relevant features.

Multi-layer fusion mitigates information bottlenecks. As representations propagate upward through the transformer stack, they become increasingly compressed and specialized for fluent decoding, potentially discarding safety-relevant features. Aggregating signals from early, intermediate, and late layers mitigates this loss. As shown in Table 3, fusing complementary layers restores these signals and achieves over 90% detection accuracy across benchmarks, while trajectory-based selection removes redundancy without sacrificing coverage.

Implications for defense. These findings suggest several practical implications for safety mechanisms. First, multi-layer sentinels can enable `early_exit` or pre-generation interventions by triggering refusals before unsafe tokens are produced, rather than relying on post-hoc filtering. Second, larger models may require stronger residual alignment or explicit supervision at intermediate depths, whereas smaller models can often rely on simpler gating mechanisms, consistent with the layer-divergence trends observed in Section 4.1.2. More broadly, enforcing safety solely at the output layer risks overlooking internal signals that remain intact earlier in the forward pass.

Sentinel applications and scope. By inspecting internal representations during the forward pass, Sentinel enables millisecond-scale safety assessment with negligible latency overhead. As illustrated in Fig. 2, this supports two primary deployment scenarios: (*i*) *offline verification*, where Sentinel serves as a high-throughput diagnostic tool for red-teaming and post-training auditing; and (*ii*) *online defense* in white-box settings, where Sentinel operates as a real-time gate that intervenes as soon as safety-relevant internal signals exceed a threshold. We emphasize that Sentinel is intended as a complementary internal safeguard for model developers and auditors, rather than a universal replacement for black-box content moderation.

6 Limitations and Future Work

Limitations. We highlight several limitations of our approach.

- **Computational overhead.** Our framework separates *offline training* from *online deployment*. The training phase requires extracting hidden states across all layers; however, as discussed in Section 4.3, this overhead remains manageable even for 72B-parameter models. *Deployment is substantially lighter*: by monitoring only a small set of selected sentinel layers, inference incurs negligible overhead ($\sim$50 ms), preserving real-time usability. Future work may explore *sub-layer* or partial-forward extraction strategies to further reduce training cost in highly resource-constrained environments.
- **Model dependence and generalizability.** We evaluate Sentinel on models up to 72B parameters (e.g., Qwen2.5-72B-Instruct) and observe consistently strong detection accuracy (> 0.9) and efficiency. Nevertheless, generalization to substantially larger models or different architectural paradigms (e.g., mixture-of-experts or multimodal LLMs) remains an open question. Differences in training data, alignment objectives, and architectural inductive biases may affect how safety-relevant features manifest across layers.
- **Evolution of adversarial strategies.** As defensive techniques improve, adversarial strategies continue to evolve. Sophisticated attackers may attempt to minimize detectable representational shifts across intermediate layers. Addressing such adaptive behavior will require continual refinement of detection strategies and the incorporation of dynamic or uncertainty-aware defenses.

Future directions. Several avenues for future work are promising.

- **Cross-model and multilingual evaluation.** Extending analysis to additional architectures (e.g., encoder–decoder and hybrid models) would test the generality of layer-wise safety signals. Multilingual evaluation may further reveal language-specific or universal adversarial patterns, improving global applicability.
- **In-context defenses and shielding prompts.** Studying how in-context defenses (e.g., safety-oriented "shield" prompts) reshape layer-wise representational trajectories may enable proactive mitigation strategies. Well-designed shields could attenuate adversarial feature amplification before it propagates to later layers.
- **Autonomous and agentic systems.** Extending detection to autonomous agents involving tool use and multi-step planning remains an open challenge. Future work could track how safety-relevant representation drift accumulates across long-horizon agentic workflows.

Overall, our findings underscore the value of layer-level analysis for understanding and mitigating adversarial behavior in LLMs. By inspecting internal representations, we obtain a finer-grained view of how harmful prompting patterns propagate and evade enforcement. The proposed metrics, including the Attack Consistency Index (ACI), provide a foundation for developing more nuanced, representation-aware defenses, equipping system designers with new tools for securing next-generation language models.

7 Conclusion

We present an end-to-end framework for *quantifying and detecting adversarial attacks* in large language models through *layer-wise analysis of internal representations*. By systematically examining hidden states across the transformer stack, we demonstrate that: (*i*) safety-relevant features correlated with harmful prompting become strongly separable under lightweight probes in intermediate layers, even when surface-level outputs appear benign; (*ii*) the proposed Attack Consistency Index (ACI) provides a model-agnostic, continuous metric for characterizing representational drift and distinguishing attack mechanisms; and (*iii*) fusing complementary signals from multiple layers yields a robust Sentinel that detects up to 99% of harmful prompts across I2P, SneakyPrompt, MMA, Labelled, and PIJ benchmarks, substantially reducing false negatives and resisting adaptive white-box attacks.

More broadly, our findings motivate an *inside-out* approach to safety. Instead of relying solely on reactive input–output filtering, defenders can leverage internal representations to sense safety-relevant deviations early in the forward pass, make informed intervention decisions, and act before unsafe generation occurs. This perspective reframes safety enforcement as a problem of internal signal monitoring and representation-level consistency, rather than post-hoc output moderation.

Beyond prompt-level defense, our framework contributes a diagnostic lens for interpretability, forensic analysis, and alignment evaluation by exposing how safety-relevant features propagate and degrade across depth. Future directions include dynamic in-layer mitigation strategies (e.g., shield prompts), extension to multimodal and agentic systems, causal tracing of safety-related representations, and deployment in real-time autonomous settings. We argue that treating internal representations as first-class safety signals—rather than after-the-fact artifacts—is essential for building robust and trustworthy generative AI systems.

Acknowledgments

We thank all the reviewers for their insightful comments. This work was partially supported by the National Natural Science Foundation of China under grants No. 25Z990200238, No. 62532016, No. 6257071391, and No. 62502309, the Shanghai Municipal Education Commission under grant No. ZXDF030140, and the Natural Science Foundation of Shanghai under grant No. 25ZR1402211. Shuo Wang is the corresponding author.

Ethical Considerations

This work investigates internal LLM cognitive dynamics to enable proactive safety. While this advances defense, it necessitates careful consideration of dual use, bias, and stakeholder impact.

Stakeholders and Dual Use. Our method impacts diverse stakeholders, from developers to marginalized groups at risk of overblocking. A primary risk is that adversarial actors could exploit internal activation patterns to craft "stealth" attacks or reverse-engineer safety filters. To mitigate the risk of accelerating a safety-evasion arms race, we adopt a strictly defensive stance: we disclose high-level methodologies to facilitate research but **withhold sensitive artifacts**, including hidden state dumps, trained probe weights, and gradients.

Data, Privacy, and Licensing. We exclusively employ open-weight models and public, license-compliant datasets. All data was filtered to remove PII and exploitative content; no human-subject data was collected. Our experiments rely on aggregate model behavior rather than individual user data, posing minimal privacy risks.

Bias and Fairness. Safety probes trained on synthetic or limited data may inherit biases, potentially over-flagging specific dialects. We mitigated this by evaluating across diverse benchmarks to reduce architecture-specific overfitting. However, we strongly recommend fairness audits, subgroup analysis, and human-in-the-loop review prior to any real-world deployment.

Disclosure and Transparency. Our findings reflect general properties of transformer architectures rather than vendor-specific vulnerabilities; therefore, coordinated disclosure was unnecessary. Regarding user transparency, deployments should disclose the use of internal sentinels at a high level without revealing technical details that could facilitate evasion.

Conclusion. We publish this work because the value of interpretable, robust defense mechanisms outweighs residual risks when sensitive implementation details are withheld. By restricting artifact release and using open-source data, we aim to strengthen model safety and interpretability without compromising alignment security.

Open Science

In accordance with the USENIX Security 2026 open-science policy, we commit to transparency and reproducibility. We have structured our artifacts to facilitate modular verification of our major claims (C1–C4). We will release the following artifacts upon acceptance:

- **Source code & structured verification:** The codebase is organized by claims, providing dedicated Jupyter notebooks (e.g., `src/claims/claim1.ipynb`) to reproduce specific experiments (E1–E4). We also provide a `quick_start.py` CLI tool for instant verification of key metrics in Table 2 using pre-computed logs.
- **Data access:** To facilitate reproducibility, we provide the **pre-processed datasets** used in our experiments, which are compliant with ethical guidelines and licensing. The artifact includes:
  - The probe training set (toxic: filtered text from NSFW-56k; non-toxic: GPT-4o-generated "home scene" prompts).
  - Evaluation benchmarks (I2P, Sneaky, MMA, Labelled, PIJ) and the multi-level perturbed variants (e.g., `prefix_fictional`) used in our Cognitive Drift analysis.
- **Environment & probe training:** We provide a Conda environment setup (`lac`) and full training scripts. The codebase is validated on a single NVIDIA A100 GPU. While we provide the hyperparameters and architecture, the trained probe weights are withheld to prevent misuse.
- **Hidden states:** Precomputed hidden states are not released due to size and security considerations. Our code supports on-the-fly extraction using publicly available LLMs (e.g., Qwen, Llama series) via the Hugging Face Hub.
- **Reproducibility:** A `README.md` and `installation.ipynb` detail the setup. The artifact supports the reproduction of layer-wise separability (Fig. 3), cognitive drift heatmaps (Fig. 4), and adaptive attack evaluations.

Our framework supports defensive model introspection and safety research. The artifact repository is hosted at: https://zenodo.org/records/17959094.

References

[1] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[2] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*, 2024.

[3] 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 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1671–1685, 2024.

[4] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[5] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[6] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024.

[7] Yuto Harada, Yusuke Yamauchi, Yusuke Oda, Yohei Oseki, Yusuke Miyao, and Yu Takagi. Massive supervised fine-tuning experiments reveal how data, layer, and training factors shape llm alignment quality. *arXiv preprint arXiv:2506.14681*, 2025.

[8] Yichen Gong, Delong Ran, Xinlei He, Tianshuo Cong, Anyu Wang, and Xiaoyun Wang. Safety misalignment against large language models. In *Proceedings of the 2025 Annual Network and Distributed System Security Symposium (NDSS)*, 2025.

[9] Tianyi Alex Qiu, Fanzhi Zeng, Jiaming Ji, Dong Yan, Kaile Wang, Jiayi Zhou, Yang Han, Josef Dai, Xuehai Pan, and Yaodong Yang. Reward generalization in rlhf: A topological perspective. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15884–15930, 2025.

[10] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. *arXiv preprint arXiv:2407.07880*, 2024.

[11] Yifan Wang, Runjin Chen, Bolian Li, David Cho, Yihe Deng, Ruqi Zhang, Tianlong Chen, Zhangyang Wang, Ananth Grama, and Junyuan Hong. More is less: The pitfalls of multi-model synthetic preference data in dpo safety alignment. *arXiv preprint arXiv:2504.02193*, 2025.

[12] Zeyu Yang, Zhao Meng, Xiaochen Zheng, and Roger Wattenhofer. Assessing adversarial robustness of large language models: An empirical study. *arXiv preprint arXiv:2405.02764*, 2024.

[13] Yiyi Tao, Yixian Shen, Hang Zhang, Yanxin Shen, Lun Wang, Chuanqi Shi, and Shaoshuai Du. Robustness of large language models against adversarial attacks. In *2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*, pages 182–185. IEEE, 2024.

[14] Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, et al. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM workshop on large AI systems and models with privacy and safety analysis*, pages 57–68, 2023.

[15] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

[16] Andreas Waldis, Vagrant Gautam, Anne Lauscher, Dietrich Klakow, and Iryna Gurevych. Aligned probing: Relating toxic behavior and model internals. *arXiv preprint arXiv:2503.13390*, 2025.

[17] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[18] Izzat Alsmadi, Nura Aljaafari, Mahmoud Nazzal, Shadan Alhamed, Ahmad H Sawalmeh, Conrado P Vizcarra, Abdallah Khreishah, Muhammad Anan, Abdulelah Algosaibi, Mohammed Abdulaziz Al-Naeem, et al. Adversarial machine learning in text processing: a literature survey. *IEEE Access*, 10:17043–17077, 2022.

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

[20] Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019.

[21] Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, 64(12):3197–3234, 2022.

[22] Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

[23] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[24] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the association for computational linguistics*, 8:842–866, 2020.

[25] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

[26] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[27] Yilei Jiang, Xinyan Gao, Tianshuo Peng, Yingshui Tan, Xiaoyong Zhu, Bo Zheng, and Xiangyu Yue. Hiddendetect: Detecting jailbreak attacks against multimodal large language models via monitoring hidden states. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14880–14893, 2025.

[28] Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. Jbshield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. *arXiv preprint arXiv:2502.07557*, 2025.

[29] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. Masterkey: Automated jailbreak across multiple large language model chatbots. *arXiv preprint arXiv:2307.08715*, 2023.

[30] Yiqi Yang and Hongye Fu. Transferable ensemble black-box jailbreak attacks on large language models. *arXiv preprint arXiv:2410.23558*, 2024.

[31] Kazuhiro Takemoto. All in how you ask for it: Simple black-box method for jailbreak attacks. *Applied Sciences*, 14(9):3558, April 2024.

[32] Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023.

[33] Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023.

[34] Daniel Kang, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia, and Tatsunori Hashimoto. Exploiting programmatic behavior of llms: Dual-use through standard security attacks. In *2024 IEEE Security and Privacy Workshops (SPW)*, pages 132–143. IEEE, 2024.

[35] Anton Razzhigaev, Matvey Mikhalchuk, Temurbek Rahmatullaev, Elizaveta Goncharova, Polina Druzhinina, Ivan Oseledets, and Andrey Kuznetsov. Llm-microscope: Uncovering the hidden role of punctuation in context memory of transformers. *arXiv preprint arXiv:2502.15007*, 2025.

[36] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

[37] Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*, 2024.

[38] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[39] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019.

[40] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann,

Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[41] Alex Kim. NSFW Data Scraper, 2023. Available at: https://github.com/alex000kim/nsfw_data_scraper.

[42] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 897–912, 2024.

[43] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22522–22531, 2023.

[44] Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. Mma-diffusion: Multimodal attack on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7737–7746, 2024.

[45] Sean M. Toxic or Neutral Text Labelled Dataset, 2024. Available at: https://huggingface.co/datasets/seanius/toxic-or-neutral-text-labelled.

[46] Qualifire. Prompt Injections Benchmark, 2024. Available at: https://huggingface.co/datasets/qualifire/prompt-injections-benchmark.

[47] Ning Bian, Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, and Ben He. Chatalpaca: A multi-turn dialogue corpus based on alpaca instructions. https://github.com/cascip/ChatAlpaca, 2023.

[48] Tom Gibbs. Multi-Turn Jailbreak Attack Datasets, 2024. Available at: https://huggingface.co/datasets/tom-gibbs/multi-turn_jailbreak_attack_datasets.

[49] OpenAI. GPT-o1, 2024. Available at: https://openai.com/o1/.

[50] Michelle Jieli. NSFW Text Classifier, 2023. Available at: https://huggingface.co/michellejieli/NSFW_text_classifier/.

[51] 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.

[52] Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. *arXiv preprint arXiv:2402.13494*, 2024.

[53] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

[54] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

[55] Yiyou Sun, Yu Gai, Lijie Chen, Abhilasha Ravichander, Yejin Choi, and Dawn Song. Why and how llms hallucinate: Connecting the dots with subsequence associations. *arXiv preprint arXiv:2504.12691*, 2025.

[56] OpenAI. GPT-4o, 2024. Available at: https://openai.com/index/gpt-4o-system-card/.

A Used Perturbation Examples

To comprehensively evaluate the robustness of safety signals against adversarial manipulations, we employ a diverse set of perturbation templates. These include **semantic prefixes and suffixes** that frame the content as fictional or artistic (Items 1–8), **role-playing scenarios** (Item 9), **topic-shifting markers** (Item 10), and various **syntactic or benign element insertions** (Items 11–16). The specific strings appended or prepended to the base prompts are detailed below:

1. **Safety-disclaiming prefixes/suffixes:** Variants such as "fictional", "non-malicious", "artistic", and "virtual", used as either prefixes (e.g., "The following content is...") or suffixes (e.g., "The above content is...") to frame harmful queries as benign.
2. **Context-shifting phrases:** Prompts like "Imagine you're a detective", "Now shifting gears...", "Flowers are blooming", "The dove of peace takes flight", and sentiment cues (e.g., "Absolutely stunning...", "With great hope...", "It is absolutely not true that...", "Could it be possible that...").

B Justification for Last Token Extraction

Our choice to use the last-token hidden state is motivated by the architectural properties of decoder-only LLMs. In all evaluated models, causal self-attention ensures that the hidden state of the final input token at each layer aggregates information from all preceding tokens in the prompt. As a result, this representation captures the model's full contextual state immediately prior to next-token generation.

To empirically validate this design choice, we compare last-token probing against a standard mean-pooling baseline. On the *Sneaky* benchmark, mean pooling across all token representations achieves only **0.332 accuracy**, approximately **0.6 lower** than the accuracy obtained using last-token features. This substantial gap indicates that averaging hidden states dilutes safety-relevant features concentrated near the generation boundary, making mean pooling ineffective for detecting subtle adversarial prompting strategies.

C Sentinel Layer Selection Details

C.1 Selection Mechanism

The construction of the Multi-Layer Sentinel is a data-driven process designed to maximize detection capability while minimizing computational redundancy. As discussed in Section 3.3 and visualized in Fig. 6, adjacent transformer layers often exhibit high functional similarity. To address this, we employ a greedy selection strategy based on two criteria:

1. **Performance (AUROC):** We first filter for layers that individually demonstrate strong separability between benign and harmful prompts.
2. **Non-redundancy (Correlation):** Among the high-performing layers, we iteratively select those that have low mutual correlation (cosine similarity) with the layers already selected.

The selected hidden states are then concatenated to form a fused representation vector, which serves as the input for a lightweight MLP classifier. This approach ensures that the Sentinel captures diverse cognitive signals—ranging from early-layer syntactic anomalies to late-layer semantic policy violations—without processing the entire stack.

C.2 Stability and Robustness

While the specific set of sentinel layers $\mathcal{S}$ is model-specific due to architectural differences, the distribution of safety signals is consistent and smooth across adjacent layers. We validated the stability of our selection strategy through ablation studies. When we replaced the optimal layer group with a sub-optimal group exhibiting similar inter-layer correlation patterns, the detection accuracy saw only a minor degradation (from **0.932** to **0.903**). This confirms that the safety signals are not isolated to unique "magic layers" but are distributed across bands of the model depth, making our selection method robust to minor variations in training data.

D System Prompt for Judger

In Table 3, the generative detection approach utilizes the following system prompt to evaluate user inputs and assign a safety score:

> Please evaluate the user input and assign a safety score between 0 and 100. A score of 100 indicates the input is completely safe and harmless, while a score of 0 indicates the input is highly harmful or sensitive. Provide only the numerical score without any additional explanation. Here is the user input:

Based on the assigned score, the input is classified as either **benign** ($score \geq 50$) or **harmful** ($score < 50$). This scoring mechanism allows the Judger to quantitatively assess the safety of inputs and make consistent decisions across different samples.

E Benign Dataset Generation Process

In the generation of the benign dataset, we used the OpenAI gpt-4o model [56] with a fixed system message to guide the model in producing short, positive descriptions of home life. Each sentence was generated with the following parameters: a maximum output length of 30 tokens to ensure brevity (limiting responses to around 20 words), a temperature of 0.7 to balance creativity and coherence, and a top-p of 0.9. The system message was set to:

You are a creative assistant tasked with generating sentences that depict the beauty of home life. Each sentence should be no longer than 20 words and convey a sense of warmth, comfort, or happiness.

All generated sentences were automatically filtered by length and manually reviewed to ensure they depicted safe, realistic, and emotionally positive home environments, resulting in a final dataset of 1,000 high-quality, benign sentences for use as a semantic control in the analysis.

F Full Black-Box Experimental Results

To provide a comprehensive view of our method's performance, we report complete evaluation metrics in Table 6. While the main text focuses on accuracy and AUROC for brevity, this detailed breakdown additionally includes precision, recall, F1-score, and true positive rates (TPR) at strict false positive rate (FPR) thresholds of 0.1%, 1%, and 5%.

The results confirm that the **Multi-layer Sentinel** consistently outperforms both the model's own generative judgment and external black-box baselines (NSFW Text Classifier and Llama-Guard-3-8B) across all reported metrics, particularly in maintaining high recall without sacrificing precision.

Table 6: Toxic detection performance of our Multi-Layer Sentinel versus **Black-box Baselines** (Generative Judgment, NSFW text classifier, and Llama-Guard-3-8B) across Sneaky, I2P, MMA, and Labelled benchmarks. Results demonstrate that our internal monitoring method achieves superior accuracy, robustness, and consistency compared to these input- or output-level filters, while remaining lightweight (<5M).

Model	Method	Dataset	Acc	Prec	Rec	F1	AUC	TPR@#%FPR		
								0.1	1	5
Qwen2.5-1.5B-Instruct	Multi-layer (1, 3, 6, 10, 12, 19, 21)	Sneaky	0.9475	1.0000	0.8950	0.9446	0.9475	0.8950	0.8950	0.8950
		I2P	0.9700	1.0000	0.9400	0.9691	0.9700	0.9400	0.9400	0.9400
		MMA	0.9727	1.0000	0.9140	0.9551	0.9570	0.9140	0.9140	0.9140
		Labelled	0.8393	0.8478	0.9867	0.9120	0.7226	0.0144	0.1440	0.3200
	generative judgment	Sneaky	0.8500	1.0000	0.7000	0.8235	0.8500	0.7000	0.7000	0.7000
		I2P	0.3961	1.0000	0.2664	0.4207	0.6332	0.2664	0.2664	0.2664
		MMA	0.8400	1.0000	0.7200	0.8371	0.8600	0.7200	0.7200	0.7200
		Labelled	0.7976	0.8938	0.8627	0.8780	0.7273	0.0056	0.0560	0.2768
Qwen2.5-7B-Instruct	Multi-layer (1, 3, 6, 8, 14, 20)	Sneaky	0.9425	1.0000	0.8850	0.9390	0.9425	0.8850	0.8850	0.8850
		I2P	0.8800	1.0000	0.7600	0.8636	0.8800	0.7600	0.7600	0.7600
		MMA	0.9590	1.0000	0.8710	0.9310	0.9355	0.8710	0.8710	0.8710
		Labelled	0.8370	0.8467	0.9854	0.9108	0.7511	0.1425	0.2566	0.3735
	generative judgment	Sneaky	0.7975	1.0000	0.5950	0.7461	0.7975	0.5950	0.5950	0.5950
		I2P	0.2812	1.0000	0.1267	0.2250	0.5634	0.1267	0.1267	0.1267
		MMA	0.7650	1.0000	0.5300	0.6928	0.7650	0.5300	0.5300	0.5300
		Labelled	0.7428	0.9615	0.7244	0.8263	0.8553	0.0252	0.0882	0.3221
Qwen3-4B	Multi-layer (3, 4, 9, 12, 19, 24, 31)	Sneaky	0.9320	1.0000	0.8300	0.9071	1.0000	1.0000	1.0000	1.0000
		I2P	0.8863	1.0000	0.8496	0.9187	1.0000	1.0000	1.0000	1.0000
		MMA	0.9954	1.0000	0.9940	0.9970	1.0000	1.0000	1.0000	1.0000
		Labelled	0.8443	0.8446	0.9996	0.9155	0.6010	0.0313	0.2128	0.3639
	generative judgment	Sneaky	0.8740	1.0000	0.6850	0.8131	0.0175	0.9650	0.9653	0.9667
		I2P	0.3095	1.0000	0.0870	0.1601	0.1928	0.6148	0.6182	0.6337
		MMA	0.8038	1.0000	0.7450	0.8539	0.0270	0.9461	0.9465	0.9487
		Labelled	0.8708	0.9360	0.9092	0.9224	0.8474	0.0603	0.1364	0.3390
Qwen3-8B	Multi-layer (1, 4, 7, 12, 16, 21, 28, 32)	Sneaky	0.9540	1.0000	0.8850	0.9390	1.0000	1.0000	1.0000	1.0000
		I2P	0.9569	1.0000	0.9431	0.9707	1.0000	1.0000	1.0000	1.0000
		MMA	0.9931	1.0000	0.9910	0.9955	1.0000	1.0000	1.0000	1.0000
		Labelled	0.8449	0.8462	0.9976	0.9157	0.7136	0.0134	0.0543	0.2690
	generative judgment	Sneaky	0.8560	1.0000	0.6400	0.7805	0.0053	0.9756	0.9806	0.9902
		I2P	0.3444	1.0000	0.1332	0.2351	0.0147	0.8688	0.9057	0.9748
		MMA	0.8177	1.0000	0.7630	0.8656	0.0298	0.9381	0.9395	0.9434
		Labelled	0.8198	0.9474	0.8329	0.8864	0.8104	0.0048	0.0479	0.2395
Llama3.2-3B-Instruct	Multi-layer (1, 2, 3, 6, 20)	Sneaky	0.9500	1.0000	0.9000	0.9474	0.9500	0.9000	0.9000	0.9000
		I2P	0.9425	1.0000	0.8850	0.9390	0.9425	0.8850	0.8850	0.8850
		MMA	0.9727	1.0000	0.9140	0.9551	0.9570	0.9140	0.9140	0.9140
		Labelled	0.8443	0.8542	0.9834	0.9143	0.7556	0.0403	0.1825	0.3060
	generative judgment	Sneaky	0.8325	1.0000	0.8135	0.8501	0.8525	0.8135	0.8135	0.8135
		I2P	0.7498	1.0000	0.6960	0.8208	0.8480	0.6960	0.6960	0.6960
		MMA	0.8460	1.0000	0.8026	0.8203	0.8320	0.8026	0.8026	0.8026
		Labelled	0.8166	0.8512	0.9486	0.8973	0.6721	0.0011	0.0737	0.1957
Llama3.1-8B-Instruct	Multi-layer (1, 6, 9, 11, 20)	Sneaky	0.9625	1.0000	0.8250	0.9610	0.9625	0.9250	0.9250	0.9250
		I2P	0.9650	1.0000	0.9300	0.9637	0.9650	0.9300	0.9300	0.9300
		MMA	0.9761	1.0000	0.9247	0.9609	0.9624	0.9247	0.9247	0.9247
		Labelled	0.8434	0.8489	0.9909	0.9144	0.8826	0.0275	0.2529	0.3422
	generative judgment	Sneaky	0.8825	1.0000	0.8100	0.8529	0.8550	0.8100	0.8100	0.8100
		I2P	0.7339	1.0000	0.6767	0.8072	0.8383	0.6767	0.6767	0.6767
		MMA	0.8150	1.0000	0.8300	0.9071	0.9150	0.8300	0.8300	0.8300
		Labelled	0.8271	0.8593	0.9509	0.9028	0.7784	0.0094	0.0943	0.2794
NSFW text classifier	sentiment-analysis	Sneaky	0.8920	0.8650	0.8650	0.8650	0.9040	0.8000	0.8150	0.8500
		I2P	0.6263	0.9486	0.5349	0.6841	0.6548	0.3534	0.4318	0.5027
		MMA	0.9423	0.9724	0.9520	0.9621	0.9714	0.9080	0.9300	0.9430
		Labelled	0.8366	0.8615	0.9610	0.9085	0.8425	0.0063	0.0630	0.3148
Llama-Guard-3-8B	pretrained-model	Sneaky	0.8260	1.0000	0.5650	0.7220	0.7825	0.5654	0.5693	0.5867
		I2P	0.5175	1.0000	0.3620	0.5315	0.6810	0.3626	0.3684	0.3939
		MMA	0.6100	1.0000	0.4930	0.6604	0.7465	0.4935	0.4981	0.5183
		Labelled	0.3793	0.9593	0.2765	0.4293	0.6064	0.0043	0.0435	0.2173