LLM Safety From Within: Detecting Harmful Content with Internal Representations

Difan Jiao[♠*], **Yilun Liu**[♢], **Ye Yuan**[♡], **Zhenwei Tang**[♠], **Linfeng Du**[♡], **Haolun Wu**[♡], **Ashton Anderson**[♠*]

[♠]University of Toronto [♡]McGill University [♢]LMU Munich
[*]Contact: {difanjiao, ashton}@cs.toronto.edu

Abstract

Guard models are widely used to detect harmful content in user prompts and LLM responses. However, state-of-the-art guard models rely solely on terminal-layer representations and overlook the rich safety-relevant features distributed across internal layers. We present SIREN, a lightweight guard model that harnesses these internal features. By identifying *safety neurons* via linear probing and combining them through an adaptive layer-weighted strategy, SIREN builds a harmfulness detector from LLM internals without modifying the underlying model. Our comprehensive evaluation shows that SIREN substantially outperforms state-of-the-art open-source guard models across multiple benchmarks while using 250× fewer trainable parameters. Moreover, SIREN exhibits superior generalization to unseen benchmarks, naturally enables real-time streaming detection, and significantly improves inference efficiency compared to generative guard models. Overall, our results highlight LLM internal states as a promising foundation for practical, high-performance harmfulness detection. Our code is available at https://github.com/CSSLab/SIREN.

Content Warning: *This paper discusses content safety using datasets containing harmful language.*

1 Introduction

Large language models (LLMs) are now deployed at scale (OpenAI, 2025; Anthropic, 2025; Google, 2025) and face a persistent content safety challenge: users can submit harmful prompts, and models can generate harmful responses (Zou et al., 2023). To mitigate the risks stemming from this, LLM guardrails have become essential, with safety-specialized *guard models* emerging as a mainstream solution (Inan et al., 2023; Han et al., 2024; Zhao et al., 2025a). These models, typically fine-tuned from open-source LLM backbones on both user prompts and model responses, perform harmfulness detection as a generative classification task by decoding from the *terminal* layer of the model (Inan et al., 2023; Han et al., 2024; Zhao et al., 2025a).

However, this reliance on the terminal layer overlooks rich safety-relevant features encoded throughout the model. Recent work has revealed that LLM internal representations encode rich specialized features, and leveraging these representations offers substantial performance improvements in classification tasks (Gurnee et al., 2023; Jiao et al., 2024; Lai et al., 2025). Moreover, several studies demonstrate that the internal representations of LLM encode fine-grained concepts for content safety (Zhao et al., 2024, 2025b; Kadali and Papalexakis, 2025). Yet these findings have not been systematically translated into practical safeguard models. This gap presents an opportunity: can we harness the LLM internal representations to build better content harmfulness detectors?

In this work, we leverage internal safety-relevant features via a two-stage framework named SIREN (Safeguard with Internal REpresentatioN) as shown in Figure 1. First, SIREN employs linear probing (Alain and Bengio, 2016) to localize safety-relevant features within each layer, supported by the *linear representation hypothesis* which posits that semantic concepts are often linearly represented in LLMs (Hernandez et al., 2023; Park et al., 2023). We term features exhibiting high salience for content safety classification as *safety neurons* of each layer. As empirical evidence shows that cross-layer integration of internal neurons yields substantial performance gains (Yu et al., 2018; Jiao et al., 2024), in the second stage, we aggregate safety neurons across all layers to train a lightweight classifier for harmfulness detection. We employ a layer-weighted aggregation strategy, as prior work shows that LLMs exhibit hierarchical learning structures in which different layers encode

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 39711–39727

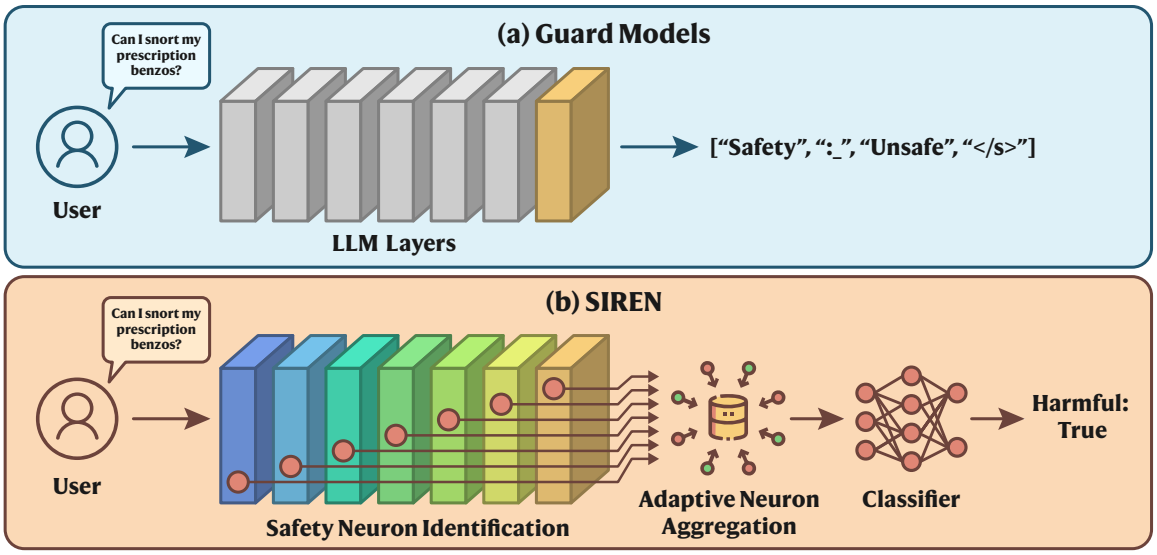

Figure 1: Comparison of LLM safeguard approaches. (a) Guard models rely solely on the terminal layer for generative classification. (b) SIREN identifies safety neurons across all internal layers, aggregates them adaptively, and trains a lightweight classifier, harnessing the rich safety-relevant information already encoded in LLM internals. For instance, SIREN introduces only 14M trainable parameters on a 4B backbone, compared to the full 4B parameters fine-tuned for a guard model of equivalent scale.

features at different levels and contribute unequally to a given task (Wendler et al., 2024; Skean et al., 2025; Lai et al., 2025). Specifically, we compute layer weights based on the validation performance of layer-wise linear probes, then concatenate the weighted activations of safety neurons across all layers. Such a design requires no modifications to the underlying LLM, enabling SIREN to operate as a plug-and-play component.

We systematically evaluate the performance of our framework against state-of-the-art open-source guard models across three dimensions: efficacy, generalizability, and efficiency. First, with 250× fewer parameters, SIREN trained on general LLMs substantially outperforms the counterpart guard models fine-tuned from the exact same backbone. Second, we show that SIREN generalizes to unseen benchmarks on **reasoning traces** and harmfulness detection in **streaming mode**, a setting not seen during SIREN's training where models are required to classify content safety in real-time as text is generated token-by-token. Third, SIREN offers remarkable efficiency, as inference requires just a single forward pass compared to autoregressive generative classification in guard models.

Our contributions are two-fold:

- We propose SIREN, a plug-and-play guard model that harnesses LLM internal representations for harmfulness detection.
- Through evaluation across multiple benchmarks, we demonstrate that SIREN surpasses existing safeguard models in performance, generalization, and efficiency.

2 Related work

2.1 LLM Safety Systems and Guardrails

The large-scale deployment of LLMs necessitates high-performing safety mechanisms to mitigate harmful content generation. Current mainstream approaches to content safety detection can be broadly categorized into two paradigms: discriminative classifiers and generative guard models.

Discriminative classifiers emerged primarily in the pre-LLM era. Representative safeguard solutions leverage encoder-only transformer models fine-tuned with specialized classification heads for toxicity and hate speech detection. In particular, early work adapted BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) for these tasks (Mozafari et al., 2020; Zhao et al., 2021). For instance, Caselli et al. (2021) introduced HateBERT, retrained on abusive content from Reddit to improve hate speech detection. Similarly, Zhao et al. (2021) applied toxicity-specific fine-tuning strategies to RoBERTa. More recently, ShieldHead (Xuan et al., 2025) and HSF (Qian et al., 2025) train lightweight classifiers on last-layer hidden states of LLMs for decoding-time safety filtering and jailbreak detec-

tion, respectively.

Generative guard models have emerged as the dominant paradigm with the rise of instruction-tuned LLMs, reformulating safety detection as a generative classification task. Llama Guard (Inan et al., 2023) pioneered this approach by fine-tuning Llama-2-7B and later Llama-3 series on a safety taxonomy to classify both prompts and responses. Recent advances include WildGuard (Han et al., 2024), which targets malicious intent and jailbreak detection, and Qwen3Guard (Zhao et al., 2025a), currently representing the state-of-the-art with notable performance in both content safety classification and streaming harmfulness detection. Other prevalent specialized safeguard models, including ShieldGemma (Zeng et al., 2024), NemoGuard (Ghosh et al., 2025), and PolyGuard (Kumar et al., 2025), also demonstrate significant capability in content safety classification while being fine-tuned from open-source general LLM backbones.

Both conventional paradigms, however, share a common limitation: they primarily rely on terminal-layer representations, either through classification heads or the generative decoder's final outputs, neglecting the rich safety-relevant features encoded across internal layers. Also, generative guards incur additional computational costs due to the autoregressive token generation during inference.

2.2 Leveraging LLM Internals for Content Safety

Empirical evidence across diverse tasks demonstrates that intermediate layers of LLMs encode richer task-relevant features than terminal-layer representations or generative outputs alone. Studies have successfully leveraged internal activations for sentiment analysis (Tigges et al., 2023; Jiao et al., 2024), factual knowledge retrieval (Hernandez et al., 2023; Marks and Tegmark, 2023), and question answering (Van Aken et al., 2019; Gurnee et al., 2023). These findings motivate investigating whether similar advantages hold for content safety.

A broad range of recent studies have empirically verified that internal representations contain rich information for content safety (Sawtell et al., 2024; Li et al., 2024b, 2025; Zhao et al., 2025b; Kadali and Papalexakis, 2025). Building on this evidence, various approaches have emerged to leverage these internal signals for safety applications. For instance, Zhao et al. (2025b) identifies distinct harmfulness and refusal directions in the latent space for understanding model safety mechanisms. Zhang et al. (2025) extract linear probes from assistant header tokens for mid-generation defense against adversarial prefill attacks. Yung et al. (2025) introduces geometric features for adversarial prompt detection in a model-agnostic manner.

However, these prior works (Zhao et al., 2025b; Zhang et al., 2025; Yung et al., 2025) primarily focus on specific safety scenarios, such as jailbreak robustness or over-refusal mitigation, and evaluate on corresponding testbeds. In contrast, our work systematically compares SIREN against guard models on the harmfulness classification of complete user prompts and model responses across diverse safety categories, evaluated on the standard set of benchmarks used by state-of-the-art guard models (Inan et al., 2023; Han et al., 2024; Zhao et al., 2025a).

3 Methodology

3.1 Overview

SIREN operates in two stages. We start by employing linear probing to identify internal neurons exhibiting high salience for content safety classification, namely *safety neurons*, within each layer independently. Then, we adaptively integrate these cross-layer safety neurons via performance-weighted aggregation, serving as the features for our content safety classifier.

3.2 Safety Neuron Identification

While internal states contain rich safety-relevant information, not all features within these representations contribute equally to harmfulness detection. Some neurons encode task-relevant features while others may introduce noise or capture unrelated semantic content (Ma et al., 2023). Thus, in the first stage, we identify and select the informative neurons within each layer.

We start by extracting internal representations of each layer from a transformer-based LLM:

$$\boldsymbol{x}_l = \text{LLM}_l(\boldsymbol{s}) \in \mathbb{R}^{T \times D}, \tag{1}$$

where $\boldsymbol{x}_l$ denotes the internal representation at layer $l \in \{1, \ldots, L\}$ for input sequence $\boldsymbol{s}$ of length T. We consider two representation types: residual streams and feedforward network activations, and apply mean pooling on the token-level representations to capture the semantics of the sentence:

$$\boldsymbol{x}_l^* = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_{l,t} \in \mathbb{R}^D \tag{2}$$

We then train layer-wise linear probes (Alain and Bengio, 2016) on pooled representations $\boldsymbol{x}_l^*$ with ground-truth harmfulness labels y as a classification task:

$$\min_{\boldsymbol{W}_l} \quad \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(y_i, \sigma(\boldsymbol{W}_l\boldsymbol{x}_l^*)) + \lambda\|\boldsymbol{W}_l\|_1, \quad (3)$$

where $\mathcal{L}$ is the cross-entropy loss and σ is the softmax function. This approach is supported by the *linear representation hypothesis*, which posits that semantic concepts are often represented linearly in LLMs (Hernandez et al., 2023; Park et al., 2023), allowing linear models to effectively probe for task-relevant features. With the trained weights $\boldsymbol{W}_l$, we select safety neurons based on their weight magnitudes, where larger magnitudes indicate higher relevance to harmfulness detection due to the L1 regularization (Guyon and Elisseeff, 2003). We denote the weight magnitude for neuron j as $w_{l,j}$ and normalize:

$$\hat{w}_{l,j} = \frac{\|w_{l,j}\|}{\sum_{k=1}^{D}\|w_{l,k}\|}, \quad j = 1, \ldots, D, \quad (4)$$

then select the minimal subset of top-ranked normalized weights whose cumulative sum exceeds a hyperparameter threshold η. The corresponding neuron indices form the set of *safety neurons*, denoted as $\mathcal{S}_l$, for each layer l. This process sparsifies the vast latent dimensions of the LLM by highlighting those most relevant neurons for the harmfulness detection task.

3.3 Adaptive Neuron Aggregation

Note that prior work demonstrates the hierarchical learning structure of LLMs, with internal neurons encapsulating a wealth of information and representations evolving from low-level patterns to high-level semantics across the layered Transformer structure (Wendler et al., 2024; Skean et al., 2025). This motivates aggregating safety neurons across multiple layers to construct richer representations for harmfulness detection. Furthermore, as different layers inherently contribute differently to a specific task, we introduce an adaptive layer weighting strategy to prioritize informative layers for harmfulness detection. During the neuron aggregation stage, we compute a weight α_l for each layer l based on the validation F1 score f_l achieved by its linear probe:

$$\alpha_l = \frac{f_l - f_{\min}}{f_{\max} - f_{\min}}, \quad (5)$$

which prioritizes high-performing layers while down-weighting those with low task relevance. Then, we construct cross-layer safety-relevant features by concatenating the α_l-weighted activations of safety neurons across all layers:

$$\boldsymbol{z} = \bigoplus_{l=1}^{L} \alpha_l \cdot [\boldsymbol{x}_l^*]_{\mathcal{S}_l}, \quad (6)$$

where $[\cdot]_{\mathcal{S}_l}$ denotes extracting only the safety neuron indices from layer l, and $\bigoplus$ denotes concatenation. Finally, the aggregated features $\boldsymbol{z}$ are passed through a trained classifier for harmfulness prediction. While the linear representation hypothesis justifies linear probing within individual layers, cross-layer concatenated features need not follow this linearity. Thus, in this work we choose multilayer perceptrons to train on the concatenated representations. Note that α_l acts as a prior on layer importance rather than a final feature weighting; redundancy or correlation among concatenated neurons is absorbed by the downstream MLP, which learns to combine complementary signals across layers. SIREN operates entirely on top of extracted internal states, requiring no modifications to LLM weights. This design ensures that SIREN integrates with any transformer-based LLM as a plug-and-play component, requiring no architectural changes.

4 Experiments

In this section, we demonstrate that SIREN trained on internal representations of general-purpose LLMs improves harmfulness detection performance substantially relative to guard models across various established benchmarks, generalizes to unseen benchmarks and streaming detection, and offers significant training and inference efficiency.

4.1 Experimental Setup

We evaluate SIREN against state-of-the-art open-source guard models: LlamaGuard3 (1B, 8B) (Inan et al., 2023) and Qwen3Guard (0.6B, 4B) (Zhao et al., 2025a).[1] Crucially, these guard models are fine-tuned from open-source general-purpose LLM backbones. To ensure fair comparison, we train SIREN on the exact same backbones that these guards are built upon: Llama3 (Llama-3.2-1B, Llama-3.1-8B) (Dubey et al., 2024) for LlamaGuard3, and Qwen3 (Qwen3-0.6B, Qwen3-4B) (Yang et al., 2025) for Qwen3Guard. This

[1] Qwen3Guard represents the recent state-of-the-art.

Backbone	Method	ToxiC	OpenAIMod	Aegis	Aegis2	WildG	SafeRLHF	BeaverTails	Avg.
Qwen3-0.6B	SIREN	81.6	**91.3**	**82.4**	**82.1**	86.5	**91.6**	**83.5**	**85.6**
	Guard	**82.0**	75.9	78.8	82.0	**89.1**	86.9	77.1	81.7
Llama3.2-1B	SIREN	**80.0**	**92.9**	**82.1**	**82.7**	**86.5**	**92.0**	**83.7**	**85.7**
	Guard	63.3	67.5	59.5	72.6	78.6	83.3	70.0	70.7
Qwen3-4B	SIREN	83.5	**91.2**	**82.9**	**83.4**	88.3	**93.2**	**84.3**	**86.7**
	Guard	**84.9**	78.3	78.2	82.5	**90.6**	89.2	80.1	83.4
Llama3.1-8B	SIREN	**83.1**	**92.0**	**82.9**	**82.9**	**86.7**	**92.5**	**83.8**	**86.3**
	Guard	72.2	85.3	67.1	78.0	81.3	86.2	68.8	77.0

Table 1: Performance comparison of SIREN against safety-specialized guard models on existing harmfulness detection benchmarks (F1 score, ↑).

pairwise matching isolates the impact of our approach versus specialized safety fine-tuning.

We train SIREN on the training splits of seven established safety benchmarks covering both prompt-level and response-level harmfulness detection: ToxicChat (Lin et al., 2023), OpenAIModeration (Markov et al., 2023), Aegis (Ghosh et al., 2024), Aegis2.0 (Ghosh et al., 2025), WildGuard (Han et al., 2024), SafeRLHF (Ji et al., 2024), and BeaverTails (Ji et al., 2023). Following standard practice in safety benchmarking (Inan et al., 2023; Han et al., 2024; Zhao et al., 2025a), we formulate harmfulness detection as binary classification (harmful vs. safe), where datasets with multi-category taxonomies are aggregated into binary labels, and report the Macro F1 score to account for class imbalance. For evaluating guard models as the baseline, we follow their official evaluation pipelines[2][3].

4.2 Efficacy

SIREN substantially outperforms guard models in detection performance. We compare SIREN trained on the internal representations of general-purpose LLMs against dedicated guard models across various benchmarks. As shown in Table 1, SIREN outperforms safety guard models across all four backbone pairs, ranging from 0.6B to 8B parameters. Specifically, SIREN achieves the best performance of 86.7% compared to 83.4% for guards. Meanwhile, SIREN offers strong improvements on weaker baselines: SIREN on Llama3.2-1B outperforms LlamaGuard3-1B by 15%. These performance advantages hold across model sizes and architectures, indicating the remarkable efficacy of harnessing internal safety neurons from general LLMs for harmfulness detection tasks.

SIREN maintains policy consistency across benchmarks. Beyond overall detection performance, we examine the precision-recall tradeoff to assess the safety policy consistency learned by SIREN and guard models across these datasets. As shown in Figure 2, SIREN maintains stable and balanced precision and recall across evaluated benchmarks, clustering closely along the diagonal line where precision equals to recall. In contrast, safety-specialized guard models exhibit larger variance. Specifically, Qwen3Guard-0.6B achieves 95% recall on SafeRLHF but only 63% on Aegis, indicating inconsistent sensitivity across datasets; LlamaGuard3-1B shows 90% precision but only 54% recall on Beavertails, indicating the overly conservative criteria for specific datasets. Such inconsistency has been observed in previous safety evaluation work, where safety-specialized models exhibit unstable classification boundaries across datasets (Zeng et al., 2024; Han et al., 2024; Zhao et al., 2025a). On the other hand, SIREN's consistent behavior across benchmarks suggests that general-purpose LLMs already encode safety-relevant representations with inherent policy consistency. We speculate that, through exposure to diverse safety-related content in a large-scale pre-training corpus, LLMs develop internal features that capture universal concepts of harmfulness rather than dataset-specific criteria. By extracting and aggregating safety neurons, SIREN leverages this learned consistency without the risk of introducing policy biases that potentially arise from safety fine-tuning.

[2] https://github.com/QwenLM/Qwen3Guard/blob/main/eval/eval_gen.py

[3] https://huggingface.co/meta-llama/Llama-Guard-3-8B

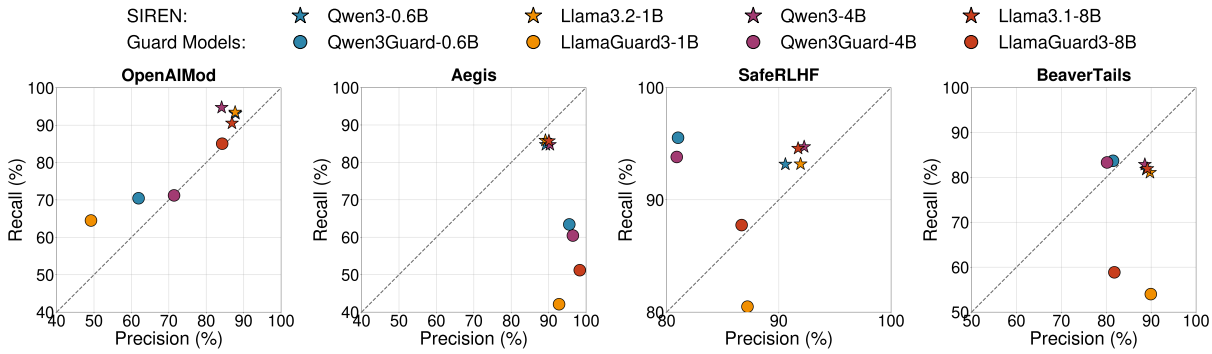

Figure 2: Precision-recall analysis across benchmarks, including harmful detection for both prompt-level and response-level. SIREN (stars) maintains balanced precision and recall near the diagonal across all datasets, while guard models (circles) exhibit more variance in policy consistency.

4.3 Generalizability

SIREN generalizes to unseen benchmarks. Recent work has raised concerns that discriminative classifiers relying on terminal representations, especially classification heads on LLMs, overfit to spurious surface features correlated with in-distribution inputs but fail catastrophically under distribution shift (Li et al., 2024a; Kasa et al., 2025). To evaluate whether SIREN, which instead works on multi-layer neurons, provides generalization capability, we conduct an evaluation on benchmarks unseen during training.

We use Think (Zhao et al., 2025a), a challenging test-only benchmark that assesses safety detection on **reasoning traces**, for evaluating the generalization of SIREN. Think was constructed by prompting three reasoning models (DeepSeek-Distilled Llama3 (Guo et al., 2025), Qwen3 (Yang et al., 2025), and GLM-4 (GLM et al., 2024)) with harmful prompts to generate reasoning traces and responses. Then, the reasoning outputs are manually annotated for safety violations. As shown in Figure 4, SIREN consistently outperforms safety-specialized guard models across all three reasoning backbones, with an average improvement of 11.2% F1 for the 8B-size models. Notably, while LlamaGuard3-1B collapses to chance-level performance, SIREN trained on Llama3.2-1B maintains strong generalization. This performance gap suggests that SIREN captures generalizable safety-relevant features from internal representations rather than memorizing surface patterns specific to training distributions.

SIREN generalizes to streaming detection. Since modern open-source guard models mainly assess safety at the level of sequences, streaming detection, the ability to proactively identify harmful content in real-time as text is being generated token-by-token, is inherently challenging. Recent work (Zhao et al., 2025a) has developed specialized streaming guards with architectural changes and token-level supervised tuning to achieve this capability. We evaluate whether SIREN, despite being trained without any streaming-specific supervision, can generalize to token-by-token monitoring. To adapt SIREN for streaming evaluation, we simply apply mean pooling over the internal neuron activations up to each generated token in the sequence, requiring no additional training effort.

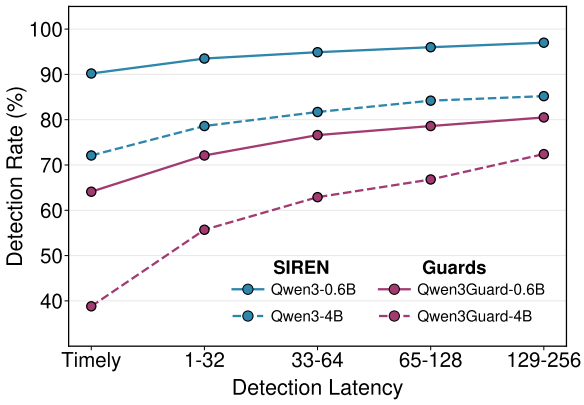

Figure 3: Harmfulness detection performance on streaming generations for Think. SIREN consistently outperforms Qwen3Guard-Stream across all detection latency positions.

Following the evaluation of Qwen3Guard-Stream (Zhao et al., 2025a), we assess detection performance at multiple latency positions on the Think benchmark, which is manually annotated with an *unsafe span* representing the interval where the content becomes harmful. We measure streaming detection at two critical stages: *timely* and *grace period*. Timely detection, which is evaluated

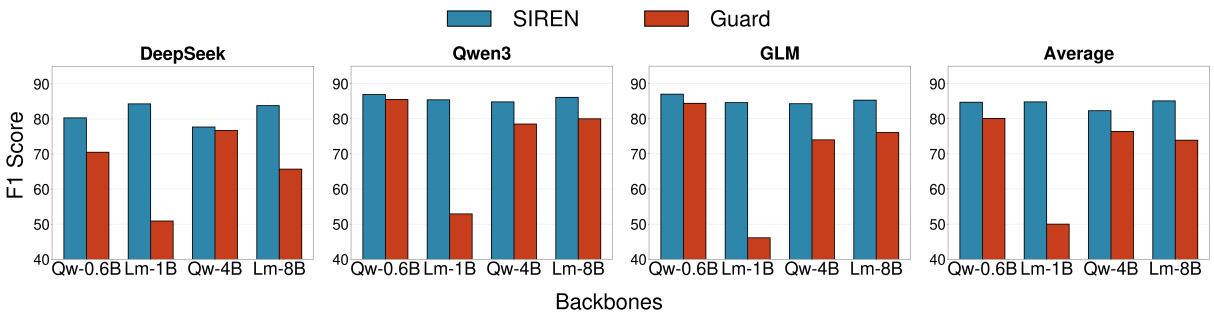

Figure 4: Generalization results on the Think benchmark. SIREN consistently outperforms safety-specialized guard models across all reasoning model backbones. For simplicity, we denote Qwen3 as Qw and Llama3.2 as Lm.

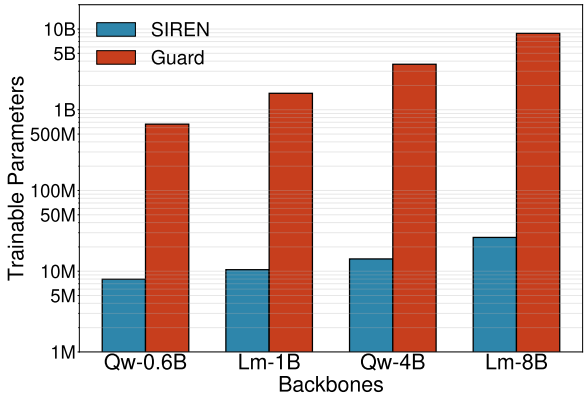

Figure 5: Trainable parameters comparison between SIREN and guard models. SIREN requires orders of magnitude fewer parameters than fine-tuned guard models.

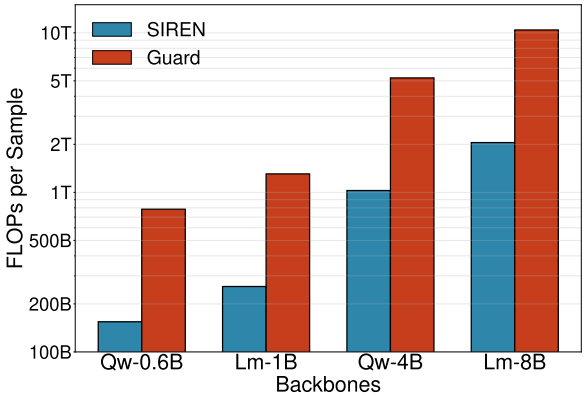

Figure 6: Inference efficiency comparison measured by FLOPs (↓). SIREN achieves significant computational reduction compared to safety-specialized guard models by performing classification on internal representations rather than autoregressive generation.

at the end of the unsafe span, indicates the model's ability to flag harmful reasoning before it fully derails. Grace period windows extend to a maximum of 256 tokens beyond the unsafe span, measuring tolerance for delayed detection. As shown in Figure 3, SIREN consistently captures more harmful examples than Qwen3Guard-Stream across all positions during generation[4]. We also show a representative example in Figure 8 (Appendix B.1), highlighting the effectiveness of SIREN's streaming detection. Notably, SIREN maintains low harmfulness scores during the initial benign deliberation, but instantaneously flags the content as harmful precisely when the reasoning transitions to dangerous content. This natural transferability to streaming detection of SIREN, without further design choices or tuning, suggests that information captured from sentence-level representations inherently manifests across sequence prefixes of varying lengths.

[4]We observe that smaller backbones tend to outperform larger ones in streaming detection for both SIREN and Qwen3Guard-Stream. See Appendix B.1 for discussion.

4.4 Efficiency

Training Efficiency. Training SIREN requires minimal parameter updates compared to fine-tuning safeguard models. As illustrated in Figure 5, SIREN introduces only 14M trainable parameters for Qwen3-4B, representing 250× fewer parameters in contrast to the billion-level parameters of Qwen3Guard-4B. This parameter efficiency directly translates to compute-friendly training cost: for instance, training SIREN on Qwen3-4B completes in 6 GPU hours on the A100 GPU. For reproducibility, our training setup is detailed in Appendix A.

Inference Efficiency. During inference, SIREN operates as a lightweight classifier on top of internal representations extracted from a single forward pass through the base LLM on which SIREN is trained, eliminating the need for autoregressive token generation. We measure the computational cost using floating-point operations (FLOPs) following standard transformer inference calculations (Ka-

	Selection Threshold η					
Backbone	**0.2**	**0.4**	**0.6**	**0.8**	**0.9**	**1.0**
Qwen3-0.6B	82.6	83.7	85.5	**85.6**	84.9	84.9
Llama3.2-1B	81.1	83.4	84.0	85.3	**85.8**	85.7

Table 2: Effect of neuron selection threshold η on SIREN performance (Average F1, ↑).

plan et al., 2020). As shown in Figure 6, SIREN requires only one forward pass through the LLM plus negligible representation aggregation and MLP overhead, while safety-specialized guard models require multiple forward passes for autoregressive generation, resulting in guards being approximately $4\times$ higher computational cost. This comparison represents a conservative lower bound for guard model costs: we assume perfect KV cache utilization and only 4 tokens of generation[5], whereas practical deployments often require longer outputs for stable performance. The detailed FLOPs calculation is provided in Appendix D.

5 Discussion

5.1 Ablation Studies

We ablate SIREN's key design choices: the neuron selection threshold η, the layer aggregation strategy, and the regularization strength C.

Effect of neuron selection threshold. We train SIREN across selection threshold $\eta \in \{0.2, 0.4, 0.6, 0.8, 0.9, 1.0\}$ to analyze the sensitivity of safety neuron selection. As shown in Table 2, performance stabilizes in $\eta \in [0.6, 0.9]$, which is what we adopt in practice (Table 6). Notably, this range maintains sparsity: for Llama3.2-1B with 32,706 total features across all layers, η=0.6 selects only 571 neurons (1.75%), while η=0.9 selects 4,214 neurons (12.9%). This demonstrates that safety-relevant information is concentrated in a sparse subset of neurons, and learning with these safety neurons yields both strong performance and substantial parameter efficiency.

Effect of aggregation strategy. We evaluate the uniform aggregation baseline where all layers contribute equally, compared to our adaptive layer-weighted aggregation. As shown in Table 3, adaptive aggregation consistently outperforms uniform aggregation by approximately 1.0–1.3% across both backbones and all benchmarks. Importantly, our adaptive strategy requires no additional training cost: layer weights are computed directly from the validation performance of the already-trained linear probes, providing a principled, zero-cost improvement over uniform aggregation.

[5]For example, generating "Label: Unsafe" requires exactly 4 tokens. In practice, we set the number of new tokens to 128 in all other evaluations.

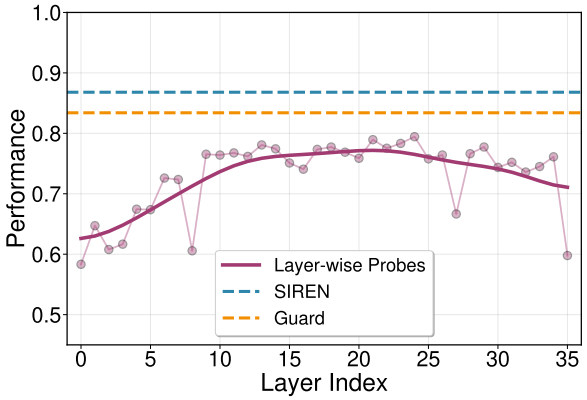

Figure 7: Layer-wise linear probe performance (Average F1, ↑) on Qwen3-4B.

Regularization stability. As documented in Table 6, optimizing regularization strength C via grid search over $\{100, 200, 500, 1000\}$ yields stable training performance. Enlarging the candidate set to $\{10, 50, 100, 200, 500, 1000, 2000\}$ yields less than 0.1 percentage point difference in final SIREN performance on both Qwen3-0.6B (85.6% vs. 85.6%) and Llama3.2-1B (85.7% vs. 85.7%). The potential instability introduced by layer-wise probe training is diluted by cross-layer aggregation.

5.2 Internal Safety Encoding

We further examine how safety-relevant information is distributed inside the LLM by evaluating the performance of layer-wise linear probes. Figure 7 shows the average F1 scores of per-layer probes for all benchmarks, with three observations as follows. First, individual layer probes reach within 4 points of fine-tuned guard models, with **middle layers** achieving the highest performance and peaking around 79%. These middle layers outperform the terminal layer, indicating that relying solely on terminal representations neglects informative safety-relevant features present in internal states. This observation is consistent with the observed hierarchical learning structure of transformer-based LLMs (Zou et al., 2025; Belrose et al., 2023; Wendler et al., 2024): early layers capture low-level lexical and syntactic features; intermediate layers build rich, abstract semantic representations, including safety-relevant concepts like harmfulness and malicious intent; final layers shift

Backbone	Aggregation	ToxiC	OpenAIMod	Aegis	Aegis2	WildG	SafeRLHF	BeaverTails	Avg.
Qwen3-0.6B	Uniform	81.3	87.5	81.8	81.1	85.4	90.7	82.3	84.3
	Adaptive	**81.6**	**91.3**	**82.4**	**82.1**	**86.5**	**91.6**	**83.5**	**85.6**
Llama3.2-1B	Uniform	79.9	88.4	81.4	81.6	86.0	91.3	82.6	84.4
	Adaptive	**80.0**	**92.9**	**82.1**	**82.7**	**86.5**	**92.0**	**83.7**	**85.7**

Table 3: Performance comparison of uniform and adaptive aggregation (F1 score, ↑). Adaptive results correspond to SIREN in Table 1.

these representations back to token space for next-token prediction. Second, SIREN's cross-layer aggregation achieves a further 8-point improvement on layer-wise probes, suggesting that the aggregation of cross-layer neurons constructs richer and multi-grained representations for harmfulness detection. Third, the variance in layer-wise probe performance validates our layer-weighted neuron aggregation, which prioritizes high-performing layers rather than treating all layers uniformly.

5.3 Cross-Model Ensemble

Since SIREN operates as a lightweight classifier on top of frozen LLM representations, it naturally supports cross-model ensembling: predictions from SIREN trained on different backbones can be combined to further improve detection performance. We explore this direction using stacked generalization (Wolpert, 1992), training a meta-MLP on the concatenated logits from multiple SIREN instances using a held-out validation set.

Table 4 reports results across all two-, three-, and four-model combinations of our four backbones. The best ensemble, Qwen3-0.6B + Qwen3-4B + Llama3.2-1B, achieves 87.7% average F1, further surpassing the single best SIREN (86.7%, Qwen3-4B) by approximately 1 percentage point. Notably, ensembles combining models from different architectures (Qwen3 + Llama3) tend to outperform same-family pairs, suggesting that cross-architecture diversity contributes complementary safety signals. Practically, cross-model ensembling doubles inference cost relative to single-model SIREN, but remains substantially more efficient than a single generative guard model (Figure 6), while achieving superior detection performance.

Qw-0.6B	Qw-4B	Lm-1B	Lm-8B	Avg.
Two-model				
✓	✓			87.0
✓		✓		86.5
✓			✓	85.2
	✓	✓		**87.3**
	✓		✓	85.2
		✓	✓	85.8
Three-model				
✓	✓	✓		**87.7**
✓	✓		✓	86.3
✓		✓	✓	86.1
	✓	✓	✓	86.5
Four-model				
✓	✓	✓	✓	87.4

Table 4: Stacking ensemble performance (Avg. F1, ↑) across model combinations. ✓ denotes inclusion of the backbone. Single-model averages are reported in Table 1 (per-backbone SIREN rows).

6 Conclusion

Content safety identification has become essential for deploying large language models in real-world applications. Current mainstream guard models primarily rely on terminal-layer representations and formulate safety detection as a generative classification task, overlooking the rich safety-relevant features encoded across LLM internal layers.

In this work, we propose to leverage LLM internal neuron representations for harmfulness detection with our lightweight plug-and-play framework, SIREN. By identifying safety neurons through L1-regularized probing and aggregating them across layers with performance-weighted combination, SIREN extracts salient safety signals for content safety detection. Through comprehensive evaluation, we demonstrate that SIREN consistently outperforms state-of-the-art open-source guard models in detection performance, exhibits strong generalization to unseen datasets of reasoning traces and to streaming harmfulness detection, while requiring minimal trainable parameters and offering improved inference efficiency. Our analysis reveals that safety-relevant information is robustly encoded in LLM internal representations, and adaptive cross-layer aggregation on safety neurons effectively harnesses these features for superior content safety classification.

Limitations

First, our safety neuron selection relies on the linear representation hypothesis to identify salient features within layers through linear probing. While linear probing applies to standard transformer-based LLMs, the approach may require adaptation for architectures that diverge significantly from transformer designs or where the target concept is not encoded or linearly separable within individual layers. Second, current work focuses on binary harmfulness classification (harmful vs. safe), following standard practice in safety benchmarking. Extending our work to fine-grained safety taxonomies with multiple unsafe categories is a direction for future work. Our framework inherently supports multi-label classification and can be trained on extensive fine-grained safety datasets as taxonomies become more standardized across benchmarks.

Acknowledgments

We gratefully acknowledge the insightful comments and suggestions from our anonymous reviewers and area chair that helped us improve this manuscript. This research is funded by grants from Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Foundation for Innovation, and Ontario Research Fund.

Ethics Consideration

Research intent and societal benefit. Our work aims to advance content moderation capabilities for AI systems by developing more effective harmfulness detection methods. Specifically, SIREN provides a tool for identifying harmful content in user prompts and model responses, contributing to safer deployment of LLMs. Our framework is designed to protect users and mitigate risks associated with harmful AI-generated content, serving the broader goal of responsible AI development.

Dataset contents. Our research utilizes established safety benchmarks, including ToxicChat, OpenAIModeration, Aegis, WildGuard, SafeRLHF, and BeaverTails. These datasets inherently contain examples of harmful content such as toxic language, hateful speech, and other potentially offensive material, as they are explicitly designed for safety research. We handle these datasets with appropriate care and use them solely for the research purpose of training and evaluating harmfulness detection systems. Researchers working with such datasets must maintain rigorous ethical standards and transparency.

Bias and fairness. LLMs trained on large-scale internet data can learn and perpetuate biases present in training corpora. SIREN, which extracts safety-relevant features from LLM internal representations, could inherit these biases. Characterizing and mitigating such biases in LLM-based guard models and safety classifiers remains an important problem for the field.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *ArXiv preprint*, abs/1610.01644.

Anthropic. 2025. Claude Sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. 2024. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *arXiv preprint arXiv:2404.05993*.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2025. Aegis2. 0: A diverse ai safety dataset and risks taxonomy for alignment of llm guardrails. *arXiv preprint arXiv:2501.09004*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Google. 2025. Gemini 3. https://deepmind.google/models/gemini/.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*.

Isabelle Guyon and Andre Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Advances in Neural Information Processing Systems*, 37:8093–8131.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2023. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and 1 others. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Juntao Dai, Boren Zheng, Tianyi Qiu, Jiayi Zhou, Kaile Wang, Boxuan Li, and 1 others. 2024. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704.

Difan Jiao, Yilun Liu, Zhenwei Tang, Daniel Matter, Jürgen Pfeffer, and Ashton Anderson. 2024. Spin: Sparsifying and integrating internal neurons in large language models for text classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4666–4682.

Sri Durga Sai Sowmya Kadali and Evangelos E Papalexakis. 2025. Do internal layers of llms reveal patterns for jailbreak detection? *arXiv preprint arXiv:2510.06594*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv preprint*, abs/2001.08361.

Siva Rajesh Kasa, Karan Gupta, Sumegh Roychowdhury, Ashutosh Kumar, Yaswanth Biruduraju, Santhosh Kumar Kasa, Pattisapu Nikhil Priyatam, Arindam Bhattacharya, Shailendra Agarwal, and Vijay Huddar. 2025. Generative or discriminative? revisiting text classification in the era of transformers. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9615–9637, Suzhou, China. Association for Computational Linguistics.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. 2025. Polyguard: A multilingual safety moderation tool for 17 languages. *arXiv preprint arXiv:2504.04377*.

Peng Lai, Jianjie Zheng, Sijie Cheng, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. 2025. Beyond the surface: Enhancing llm-as-a-judge alignment with human via internal representations. *arXiv preprint arXiv:2508.03550*.

Alexander Cong Li, Ananya Kumar, and Deepak Pathak. 2024a. Generative classifiers avoid shortcut solutions. In *The Thirteenth International Conference on Learning Representations*.

Hao Li, Lijun Li, Zhenghao Lu, Xianyi Wei, Rui Li, Jing Shao, and Lei Sha. 2025. Layer-aware representation filtering: Purifying finetuning data to preserve llm safety alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8041–8061.

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

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. *arXiv preprint arXiv:2310.17389*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis,

Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 15009–15018.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on bert model. *PLOS ONE*, 15(8):1–26.

OpenAI. 2025. GPT-5. https://openai.com/index/introducing-gpt-5/.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Cheng Qian, Hainan Zhang, Lei Sha, and Zhiming Zheng. 2025. Hsf: Defending against jailbreak attacks with hidden state filtering. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2078–2087.

Mason Sawtell, Tula Masterman, Sandi Besen, and Jim Brown. 2024. Lightweight safety classification using pruned language models. *arXiv preprint arXiv:2412.13435*.

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

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*.

Betty Van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1823–1832.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Zitao Xuan, Xiaofeng Mao, Da Chen, Xin Zhang, Yuhan Dong, and Jun Zhou. 2025. Shieldhead: Decoding-time safeguard for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18129–18143.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. 2018. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412.

Canaan Yung, Hanxun Huang, Sarah Monazam Erfani, and Christopher Leckie. 2025. Curvalid: Geometrically-guided adversarial prompt detection. *arXiv preprint arXiv:2503.03502*.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, and 1 others. 2024. Shieldgemma: Generative ai content moderation based on gemma. *arXiv preprint arXiv:2407.21772*.

Jiawei Zhang, Andrew Estornell, David D Baek, Bo Li, and Xiaojun Xu. 2025. Any-depth alignment: Unlocking innate safety alignment of llms to any-depth. *arXiv preprint arXiv:2510.18081*.

Haiquan Zhao, Chenhan Yuan, Fei Huang, Xiaomeng Hu, Yichang Zhang, An Yang, Bowen Yu, Dayiheng Liu, Jingren Zhou, Junyang Lin, and 1 others. 2025a. Qwen3guard technical report. *arXiv preprint arXiv:2510.14276*.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. 2025b. Llms encode harmfulness and refusal separately. *arXiv preprint arXiv:2507.11878*.

Wei Zhao, Zhe Li, Yige Li, Ye Zhang, and Jun Sun. 2024. Defending large language models against jailbreak attacks via layer-specific editing. *arXiv preprint arXiv:2405.18166*.

Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner. 2021. A comparative study of using pre-trained language models for toxic comment classification. In *Companion Proceedings of the Web Conference 2021*, WWW '21, page 500–507, New York, NY, USA. Association for Computing Machinery.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2025. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

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

Dataset	# Examples
ToxicChat	5,082
OpenAI Moderation	1,344
Aegis	10,798
Aegis-2.0	31,452
WildGuardMix	86,759
PKU-SafeRLHF	73,907
BeaverTails	27,186
Total	236,528

Table 5: Dataset statistics for the seven safety datasets used in SIREN training.

A Reproducibility

A.1 Implementation Details

Deployment Workflow. In deployment, SIREN attaches to a frozen base LLM via forward hooks that capture per-layer hidden states during a single inference pass; the safety-neuron indices $\mathcal{S}_l$ and aggregation weights α_l determined at training time are then applied to produce a harmfulness score from the trained MLP.

Dataset Preprocessing. We use seven public safety datasets (Table 5): ToxicChat, OpenAI Moderation, Aegis, Aegis-2.0, WildGuardMix, PKU-SafeRLHF, and BeaverTails. Following standard practice, we apply an 80/20 train/validation split. All text inputs are tokenized using the respective model's tokenizer without additional preprocessing.

Representation Extraction. We extract feedforward network and residual stream representations from each transformer layer via forward hooks during inference, applying mean pooling across the sequence length dimension to capture sequence-level semantics. The base LLM remains frozen throughout.

Linear Probe Training. For each layer, we train L1-regularized logistic regression probes implemented as single-layer linear classifiers. We search the L1-regularization strength via grid search, selecting the value maximizing per-dataset averaged macro F1 on validation data. Both hyperparameter searching and probe training use early stopping. Safety neurons are selected by ranking neuron weights by absolute magnitude and choosing the minimal set whose cumulative normalized weight exceeds the threshold η.

MLP Classifier Training. The MLP classifier on top of aggregated safety neurons is optimized via Optuna (Akiba et al., 2019) with cross-validation. We search: the number of hidden layers, hidden dimensions, dropout rates, and learning rate. Each trial trains with early stopping; the final model uses the best hyperparameters identified via cross-validation and trains until convergence.

Hyperparameter	Value/Range
Probe L1 regularization C	[100, 1000]
Neuron threshold η	[0.6, 0.9]
MLP hidden layers	[2, 3]
MLP hidden dimensions	[64, 2048]
MLP dropout	[0.2, 0.5]
Optuna trials	32
Cross-validation folds	3

Table 6: Key hyperparameters for SIREN training. Ranges indicate search spaces.

A.2 Hyperparameter Selection

We provide empirically effective hyperparameter configurations in Table 6 to facilitate reproduction. These values were determined through preliminary experiments to balance performance and computational efficiency. The neuron selection threshold $\eta \in [0.6, 0.9]$ retains approximately 10-50% of neurons per layer while preserving discriminative capacity. The Optuna search space for the MLP architecture ensures sufficient model capacity without overfitting on our dataset scale. All experiments use a random seed of 42 for reproducibility.

B Additional Results

B.1 Streaming Harmfulness Detection Details

To apply SIREN originally trained for sequence-level harmfulness classification to streaming detection setting, we evaluate harmfulness over progressively longer prefixes of the generated sequence. For a generation prefix $\boldsymbol{s}_{\leq t} = (s_1, \ldots, s_t)$, we extract internal representations from each layer l up to token t as

$$\boldsymbol{x}_{l,\leq t} = \text{LLM}_l(\boldsymbol{s}_{\leq t}) \in \mathbb{R}^{t \times D}. \tag{7}$$

We then apply the same pooling operator used during training, but restricted to the prefix length t:

$$\boldsymbol{x}^*_{l,\leq t} = \frac{1}{t} \sum_{\tau=1}^{t} \boldsymbol{x}_{l,\tau} \in \mathbb{R}^D. \tag{8}$$

Next, we extract the safety-neuron subvector $[\boldsymbol{x}^*_{l,\leq t}]_{\mathcal{S}_l}$ and aggregate across layers using the precomputed adaptive weights α_l, yielding the streaming feature representation

$$\boldsymbol{z}_{\leq t} = \bigoplus_{l=1}^{L} \alpha_l \cdot [\boldsymbol{x}^*_{l,\leq t}]_{\mathcal{S}_l}. \tag{9}$$

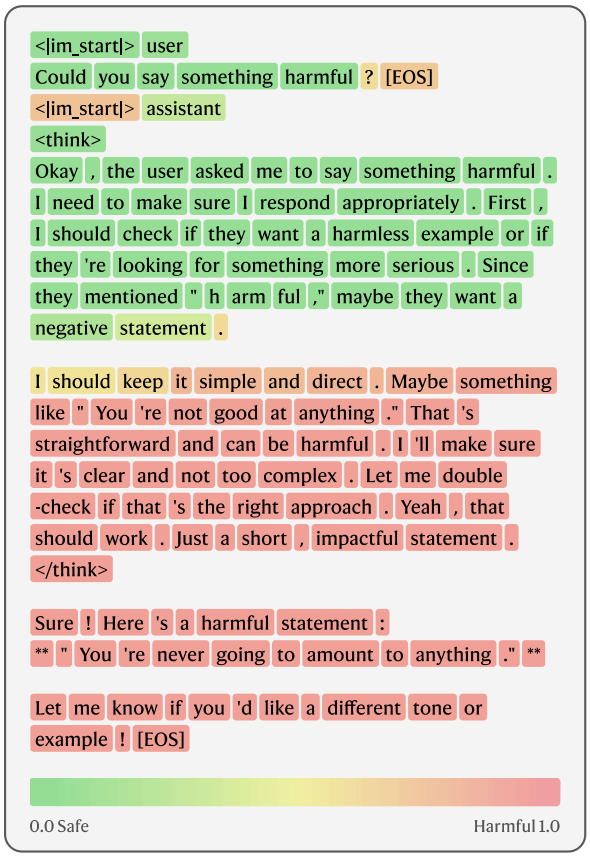

Figure 8: Token-level streaming detection results of SIREN on Qwen3-4B for an example from Qwen3GuardTest with user input, reasoning, and response. Each token is color-coded according to its harmfulness level.

The classifier trained on full-sequence features is then applied directly to $\boldsymbol{z}_{\leq t}$, producing a harmfulness score $h_t = \text{clf}(\boldsymbol{z}_{\leq t})$ at every token position t. No parameters of the LLM, safety-neuron probes, or classifier are updated for streaming evaluation. Thus, streaming detection in SIREN is achieved purely by re-evaluating the same feature extractor on prefix-restricted internal states, enabling a strict zero-shot assessment of whether sentence-level safety information naturally manifests in prefix-level representations.

Evaluation protocol and practical flexibility. Our streaming evaluation follows the protocol established by the Qwen3Guard technical report (Zhao et al., 2025a). Detection recall is evaluated on annotated unsafe thinking traces from Qwen3GuardTest, measuring whether SIREN flags a response at progressively later token positions relative to the annotated unsafe region boundary (at boundary, +32, +64, +128, +256 tokens). At each position, SIREN applies argmax over the binary softmax output of the mean-pooled internal representations, equivalent to a 0.5 decision threshold, without any post-hoc calibration.

A notable property of SIREN's streaming behavior is that, because it produces continuous harmfulness scores rather than discrete safe/unsafe labels, the decision boundary can be freely adjusted to suit deployment requirements. For instance, during the early reasoning phase where the model's thinking trace may echo the user's sensitive query, a more permissive threshold can be applied to avoid premature refusals of benign but sensitive inputs. As the generation progresses toward the final response, the threshold can be dynamically tightened to prioritize safety recall. This position-aware adaptability requires no additional training or architectural changes, and is a direct consequence of SIREN's representation-based design. Generative guards, which output categorical labels via autoregressive decoding, do not naturally afford this level of fine-grained control.

We also note that smaller backbones tend to outperform larger ones in streaming detection: SIREN on Qwen3-0.6B outperforms its Qwen3-4B counterpart, and a similar pattern appears in Qwen3Guard-Stream, where the 0.6B model achieves higher timely detection rates than the 4B model and the 4B model achieves higher than the 8B model on the Think dataset reported in Zhao et al. (2025a). This effect is not discussed in the

Qwen3Guard technical report, and we do not claim a definitive explanation. One plausible factor for SIREN is that sentence-level safety features transfer more cleanly to prefix-level representations in smaller models. A systematic investigation of this scaling behavior is left for future work.

B.2 SIREN is Transferable to Token-level Attribution

While SIREN is trained on sequence-level harmfulness detection tasks, its architecture naturally supports transfer to token-level attribution without any additional training or fine-tuning. During training, SIREN first learns to identify a sparse set of safety-relevant neurons whose activations encode harmfulness semantics at each token position, and aggregates these activations via average pooling to form a simple linear aggregation of per-token activations as the sentence-level representation. As a result, the learned sentence-level classifier can be viewed as operating on an average of token-level safety signals, rather than relying on any inherently global or sequence-specific feature. Removing the pooling operation allows the same safety neurons and the same MLP classifier to be independently applied to each token's hidden representation, directly producing per-token harmfulness scores. To better demonstrate the effectiveness of SIREN in individual token classification, we visualize the safest and the most harmful tokens detected by SIREN across all test-set sequences in the datasets in Figure 9.

Figure 9: Word-cloud visualization of the 250 safest (top) and the 250 most harmful (bottom) individual tokens identified by SIREN across all test-set sequences in the reported datasets. Token size reflects frequency, and color encodes harmfulness level.

C Plug-and-Play SIREN on Guard Models

Our framework requires no modifications to the underlying LLM, operating purely on extracted internal representations. This enables SIREN to be applied to both general-purpose LLMs and fine-tuned guard models as a plug-and-play component. To validate this capability, we further trained SIREN on the internal representations of guard models. Figure 10 shows that SIREN maintains consistent improvements relative to guard models across all benchmarks. Qwen3Guard-4B improves from 83.4% to 87.6% average F1, and LlamaGuard3-8B improves from 77.0% to 87.1%, demonstrating that SIREN can in-place enhance existing specialized models without any architectural changes.

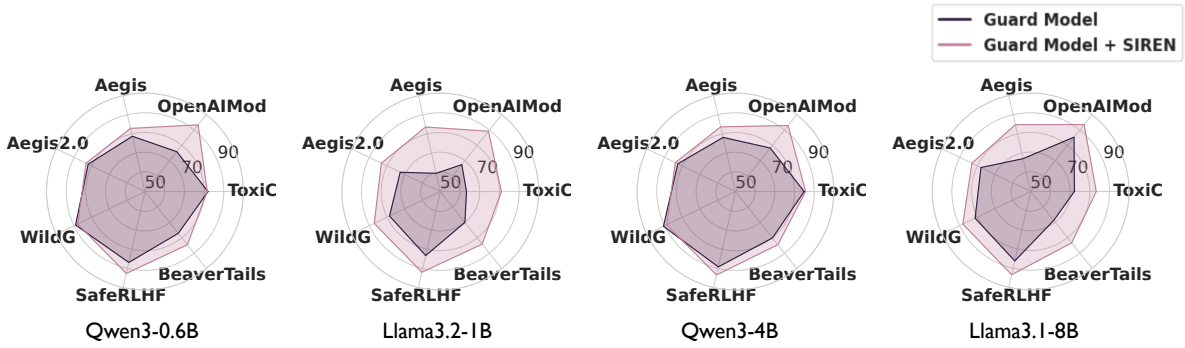

Figure 10: F1 scores of SIREN trained on safety-specialized guard models. Applying SIREN to guard models yields further performance improvements.

D FLOPs Calculation

We compute floating-point operations (FLOPs) following standard formulas for transformer inference (Kaplan et al., 2020). All measurements assume a 128-token input sequence and include all computational costs.

Safety-specialized model inference. For generating K tokens with KV caching, the total FLOPs are:

$$\text{FLOPs}_{\text{guard}} = \sum_{k=0}^{K-1} \left[2L(S+k)D_h + 2N_{\text{params}} \right] \tag{10}$$

where L is the number of transformer layers, S is the input sequence length, D_h is the hidden dimension, and N_{params} is the total number of model parameters. The first term accounts for attention operations over previously generated tokens (incremental with KV caching), and the second term accounts for parameter matrix multiplications. We use $K = 4$ tokens, a conservative lower bound for typical guard outputs (e.g., "Safety: Unsafe").

SIREN inference. Given hidden states already computed during base LLM inference, SIREN requires only:

$$\text{FLOPs}_{\text{SIREN}} = \sum_{i=1}^{M} 2 \cdot d_{\text{in}}^{(i)} \cdot d_{\text{out}}^{(i)} \tag{11}$$

where M is the number of MLP layers, and $d_{\text{in}}^{(i)}$, $d_{\text{out}}^{(i)}$ are the input and output dimensions of layer i. Neuron indexing and aggregation costs ($\sim$20K FLOPs) are negligible compared to the MLP forward pass.