

Before the First Token: Jailbreak Detection with Attention Probes

Pedro Faustini¹, Annabelle McIver¹ and Mark Dras¹
School of Computing, Macquarie University, Sydney, Australia¹
{pedro.faustini, annabelle.mciver, mark.dras}@mq.edu.au

Abstract

Large language models (LLMs) are deployed with safety guardrails, yet jailbreak attacks and malicious prompts keep bypassing them. Re-aligning an LLM each time a new attack appears is more expensive than crafting attacks, so jailbreak detection is increasingly framed as a pre-processing step: a cheap detector filters out a malicious prompt *before* it reaches the LLM. This need is reinforced by multilingual attacks, where simply translating a prompt to another language can bypass the guardrails. Our goal is to find whether internal representations of a *frozen* LLM already separate malicious from benign prompts, and whether such a signal is resilient against translation. We study two types of detectors that require no fine-tuning of the LLM: a logistic-regression classifier over whole-sentence embeddings, and a probe-based detector that trains one logistic-regression probe per attention head on the attention activations of the last input token. They need at most a single forward pass. Our experiments¹ show that both match or outperform strong sentence-embedding detectors and off-the-shelf baselines, specially under translation. The probes degrade more gracefully on higher-resource languages and remain competitive on lower-resource ones. Our results indicate that a frozen LLM linearly encodes whether an incoming prompt is malicious in a small set of mid-to-late layers. Per-head attention probes thus offer a defence that can be cheaply re-trained as new attacks emerge, without touching the underlying model.

1 Introduction

Large Language Models (LLMs) are trained with guardrails to prevent them from producing harmful content (e.g. hate speech or instructions about how to injure someone). Several jailbreak attacks have been proposed to bypass the guardrails (Shen et al.,

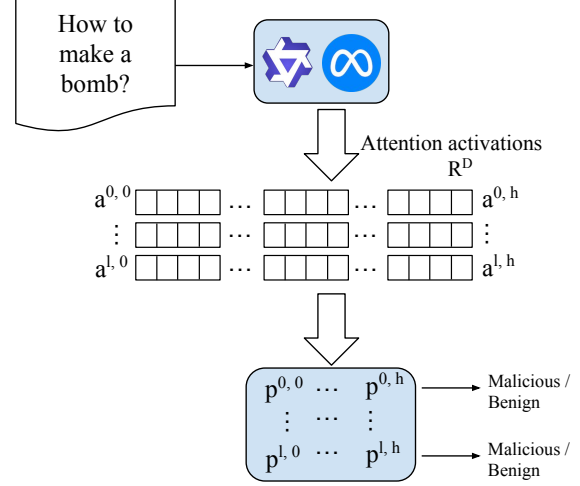


Figure 1: Attention activations are extracted per head in each layer, and used as features to train linear classifiers that infer if the prompt is malicious or benign.

2024; Chao et al., 2025; Mehrotra et al., 2024; Liu et al., 2025; Huang et al., 2025). This creates a unfavourable cat-and-mouse game for LLMs: whilst algorithms that generate attacks are cheap (usually crafting different versions of malicious prompts), re-training or fine-tuning LLMs is costly.

Thus, the problem of **jailbreak detection** has been framed as a pre-processing step. The rationale is that as jailbreak techniques evolve, only the (lightweight) detector, and not the entire LLM, needs to be adjusted. Techniques range from deploying dedicated filter classifiers like LLaMA Guard (Inan et al., 2023), having hidden system prompts (Zhang et al., 2024), filtering inputs based on some metric and threshold (Alon and Kamfonas, 2023; Chen et al., 2025; Tong et al., 2026), orchestrating multiple LLMs to examine the prompt (Zeng et al., 2024).

However, these defence techniques may also harm the utility of the models for benign prompts: Mai et al. (2025) found that many jailbreak de-

¹<https://github.com/phfaustini/BFT>

fences fail to keep safety and utility simultaneously, in particular in methods that modify the prompt or perform further fine-tuning, while cross-examining prompts with LLMs is expensive at inference time. Also, a particular further problem not specifically addressed is the risk from multilingual jailbreak attacks: prompts in non-English languages, especially low-resource ones, have been shown to be especially effective in jailbreaking (Deng et al., 2024; Ding et al., 2025; Dunca et al., 2025).

We propose a new approach to detection drawing on insights from two areas: (1) adversarial example detection based on learned representations, originally from the computer vision literature (Feinman et al., 2017; Ma et al., 2018; Cohen et al., 2020) but later applied to text (Liu et al., 2022; Tonni and Dras, 2023); and (2) probing of learned model-internal representations to help establish properties like truthfulness (Li et al., 2023). We define two kinds of malicious prompt detectors based on these ideas, and we expect, given the well-recognised geometric properties of multilingual language models where different languages occupy similar linear subspaces (Pires et al., 2019; Conneau et al., 2020; Chang et al., 2022), that these approaches will work well in the context of multilingual jailbreaking.

Our main contributions are:

1. We propose two learned representation-based jailbreak detectors, both cheap to train and run.
2. We show that these outperform baselines from the recent literature on detecting both malicious prompts in natural language and adversarially derived malicious prompts. In particular, they are notably more effective than baselines in detecting multilingual jailbreaks.
3. The probe-based approach is especially strong on multilingual jailbreaks. This gives some insight into the geometry of multilingual models: the maliciousness signal is concentrated in a small set of mid-to-late-layer attention heads, and these per-head features stay more stable under translation than either hidden states or whole-sentence embeddings.

2 Related Works

2.1 Jailbreak Attacks

A central concern about deploying LLMs is limiting them from giving malicious responses (Weidinger et al., 2022) from jailbreak attacks. Attacks broadly fall into two categories: creating adversarial prompts or attacking model weights (Dong

et al., 2024). The former has received particular attention because it can be deployed against black-box models. For example, AutoDan (Liu et al., 2024) and AutoDan-Turbo (Liu et al., 2025) work by automatically evolving jailbreak prompts (the former with a genetic algorithm; the latter via an agent that recombines a growing library of attack strategies based on the victim’s responses), whereas ReneLLM (Ding et al., 2024) nests the harmful request inside a template (e.g. a code-completion task or a fictional dialogue), so the malicious intent is delivered under a benign surface form. Moreover, most jailbreak research is carried in English. Deng et al. (2024) show, for instance, that translating a malicious request into a lower-resource language may bypass the guardrails.

2.2 Multilingual Jailbreaking

Safety alignment is mostly evaluated in English, which leaves a gap that can be exploited simply by changing the language of the request. Deng et al. (2024) show that translating a harmful prompt into a non-English, and especially a lower-resource, language can bypass the safety alignment of otherwise well-guarded LLMs. Subsequent work makes this threat systematic: Ding et al. (2025) automate the construction of multilingual jailbreaks rather than relying on hand-crafted translations, and Dunca et al. (2025) study multilingual red-teaming and defence, however by fine-tuning an LLM with a multilingual dataset.

2.3 Safeguarding LLMs

The first line of defence against malicious prompts lies on alignment, which requires further training the models to avoid certain behaviours, commonly via Supervised Fine-Tuning (Ouyang et al., 2022) or Reinforcement Learning from Human Feedback (Christiano et al., 2017; Bai et al., 2022).

It is expensive to adjust such alignments every time a new attack surfaces. Therefore, a complementary line of defence is to deploy a *separate* moderation model that classifies prompts (and responses) before they reach the LLM. LLaMA Guard (Inan et al., 2023) is a language model of moderate size (e.g. Llama-2-7B) fine-tuned on a safety-taxonomy dataset to label inputs and outputs as safe or unsafe. Flavours of this model have been proposed (Yang et al., 2025b). A lighter family of detectors avoids fine-tuning and instead extracts a scalar signal from a language model’s own forward pass. Alon and Kamfonas (2023) proposed to

flag prompts whose **perplexity**, as measured by an off-the-shelf causal LM such as GPT-2 (Radford et al., 2019), exceeds a threshold. **FJD** (Chen et al., 2025) reads the logits of the first generated token and converts them into a jailbreak score. Our methods in contrast build classifiers over different types of learned representations.

3 Proposed Methods

Chang et al. (2022) found that language models encode features onto orthogonal axes in the representation space, allowing the encoding of several signals for downstream tasks and multilingual learning, which is precisely the case for the detection of malicious prompts.

Our proposal inherits the linear-probing and inspecting the internal activations premise of Li et al. (2023); Nakka et al. (2025) but pursues a different objective: rather than steering generation or surfacing a single concept direction, we ask whether the per-head attention output of a frozen LLM already separates malicious from benign prompts *before* the tokens are generated.

3.1 Embedding-based Method

Certain behaviours or intents (truthfulness, sentiment, refusal) may be found by encoding a sentence into a dense semantic vector. The intuition is akin to previous works on detecting adversarial attacks to neural networks (Cohen et al., 2020; Liu et al., 2022). We convert the text prompt into a embeddings representation, which is a numerical vector serving as features to train classifiers. Sentence-embedding models (Reimers and Gurevych, 2019) map an entire prompt to a single dense vector. If maliciousness is largely a semantic property of the prompt, then a linear classifier on top of a good sentence embedding should already separate malicious from benign inputs.

Concretely, we embed each prompt x_i with a frozen sentence-embedding model into a vector $e(x_i) \in \mathbb{R}^d$ and train a single L2-regularised logistic-regression classifier on $\{(e(x_i), y_i)\}_{i=1}^N$.

3.2 Probe-based Method

Instead of having to encode a sentence through an entire encoder, we propose using only a small number of attention-heads in the Transformer block. The intuition remains the same, i.e., maliciousness should be encoded in a subset of the attention-heads. Since the model has been exposed to different languages during its training, such property

should still be encoded when a prompt is translated, even though its surface form changes substantially.

Let f_θ denote a pre-trained language model with L layers and H self-attention heads per layer. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a labelled set of prompts, where x_i is a natural-language query and $y_i \in \{0, 1\}$ indicates whether x_i is malicious ($y_i=1$) or benign ($y_i=0$). Our goal is to train a *lightweight* detector $g : \mathcal{X} \rightarrow [0, 1]$ that estimates $\Pr(y=1 \mid x)$ at a small fraction of the cost of running f_θ to completion or of fine-tuning f_θ for safety.

We keep f_θ frozen, extract its internal activations on x , and use them as features to train small linear classifiers. The intuition is that safety-relevant features can be linearly obtained from a handful of *specific* attention heads, so a collection of small probes is sufficient to flag a malicious prompt *before* the target LLM ever generates a token.

We extract the features to train the probes from the attention activations $a_{\ell,h}(x_i) \in \mathbb{R}^{d_h}$ of the last token of input x_i . Therefore, for a model with L layers and H attention heads per layer, a total of $L \times H$ probes are trained, each with dimensionality d_h . More specifically, in a transformer layer ℓ the multi-head attention block computes

$$o_{\ell,h}(x_i) = \text{softmax}\left(\frac{Q_{\ell,h} K_{\ell,h}^\top}{\sqrt{d_h}}\right) V_{\ell,h} \in \mathbb{R}^{T \times d_h},$$

where T is the sequence length. The per-head outputs are concatenated along the feature axis and projected by the output matrix $W_{O,\ell}$. We register a forward hook on $W_{O,\ell}$ that intercepts its input *before* the projection, so the per-head structure is preserved. We keep only the row corresponding to the last input token T and reshape it into H vectors of size d_h , yielding the activation

$$a_{\ell,h}(x_i) \in \mathbb{R}^{d_h}, \quad \ell \in \{1, \dots, L\}, \\ h \in \{1, \dots, H\}.$$

This requires a single forward pass over x_i with no full text generation, and only the last-token activations are retained, which keeps the memory footprint linear in $L H d_h$ per prompt.

We assume that there are training and evaluation datasets $\mathcal{D}_{train}$ and $\mathcal{D}_{eval}$ to train the probes. Each probe is evaluated on the F1 score over the evaluation set. For every (ℓ, h) pair we form the dataset $\{(a_{\ell,h}(x_i), y_i)\}_{i=1}^{|\mathcal{D}_{train}|}$ and train an independent binary classifier $p_{\ell,h} : \mathbb{R}^{d_h} \rightarrow [0, 1]$, giving a total of $L \times H$ probes of dimensionality d_h .

For inference, let $\mathcal{S}_k$ be the set of the k probes with the highest validation F1 on the evaluation set. For an unseen prompt x , we extract the activations at the layers $\mathcal{L}(\mathcal{S}_k) = \{(\ell, \cdot) : (\ell, \cdot) \in \mathcal{S}_k\}$ corresponding to these probes (a strict subset of all layers) and average their resulting class-1 probabilities:

$$g(x) = \frac{1}{k} \sum_{(\ell, h) \in \mathcal{S}_k} p_{\ell, h}(a_{\ell, h}(x)),$$

4 Experiment setup

We describe next our experimental setup regarding methods and data.

4.1 Datasets of Natural Jailbreaks

We experiment on the datasets below, containing both benign and malicious inputs:

- WILDGUARDMIX (Han et al., 2024) is a dataset of prompts labelled as either harmful or harmless. The data comes from publicly available sources and also synthetic generated texts. It contains questions with and without instruction prompts.
- TRUSTAIR IN-THE-WILD JAILBREAK is a dataset made of English prompts collected by Shen et al. (2024) from open platforms (e.g. websites, open datasets). They were categorised into regular or jailbreak prompts. This dataset contains only the prompts, i.e., the instructions given to the LLM before the actual question.
- ALLENAI WILDJAILBREAK is a multi-lingual dataset (Jiang et al., 2024) with input-response pairs mined from user-chatbot interactions. It has questions with and without instruction prompts.
- VANILLAWILDJAILBREAK is a subset of ALLENAI WILDJAILBREAK containing only the samples that are basic questions, without any instruction prompts.

4.2 Datasets of Adversarial Jailbreaks

We also assess the detection capabilities of our model and baselines against prompts crafted by the following jailbreak attacks. The creation of adversarial attacks can be sensitive to hyperparameters and other factors, as we discuss in Appendix A.1. We therefore take two existing datasets of adversarial prompts produced by the authors of those works. However, as these only consist of (adversarial) malicious prompts and no benign prompts, we create datasets by matching the malicious prompts to semantically close same-domain data.

- The ReNeLLM attack (Ding et al., 2024) rewrites prompts by nesting them inside templates (e.g. a code-completion or paragraph-continuation task). RENELLM-JAILBREAK is a dataset of malicious-only inputs produced by it over AdvBench dataset prompts (Zou et al., 2023). Since RENELLM-JAILBREAK is malicious-only, we pair it with benign prompts from WILDGUARDMIX to obtain a balanced binary task, which we refer to as RENE+WG. We explain in detail this pairing in Appendix A.1.
- AUTODAN-TURBO is an attack to create jailbreak prompts proposed by Liu et al. (2025). The authors released a dataset derived from the HarmBench dataset (Mazeika et al., 2024).

Models. We use Qwen3-Embedding-8B² and MPNet-base³ as base models for our embedding-based detectors. For the attention-based probes, we use Qwen 3-8B (Yang et al., 2025a) and LLaMA 3.2-1B (Grattafiori et al., 2024). They were then used as features to train a Logistic Regression classifier in scikit-learn (Pedregosa et al., 2011). We note that the Qwen3 family of base models is shared between the two approaches, letting us compare a whole-sentence embedding against a per-head of the same family.

Detector Baselines. We compare our detection method with the following baselines:

- **Perplexity:** Alon and Kamfonas (2023) assumed that adversarial suffix prompts contain non-fluent sequences of text, which would incur in higher perplexity. The authors used GPT-2 (Radford et al., 2019), whereas we adopted the larger GPT-NEO 2.7B (Black et al., 2021)
- **FJD** (Chen et al., 2025): the method prepends an affirmative instruction to the prompt and scales the logits to distinguish between jailbreak and benign prompts through the confidence of the first generated token.
- **LLaMA Guard** is a family of models fine-tuned for content safety classification. We used Llama-Prompt-Guard-2-86M⁴, with experiments with Llama-Guard-3-8B (Inan et al., 2023), and DecipherGuard (Yang et al., 2025b) in the Appendix.
- **TF-IDF** is a weighting scheme (Salton and Buckley, 1988) to represent texts as high-dimensional

²<https://huggingface.co/Qwen/Qwen3-Embedding-8B>

³<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

⁴[meta-llama/Llama-Prompt-Guard-2-86M](https://huggingface.co/meta-llama/Llama-Prompt-Guard-2-86M)

bag-of-words feature vectors, where each dimension is weighted by term frequency and inverse document frequency. The resulting features were used to train a Logistic Regression detector.

4.3 Language setup

Monolingual experiments. The datasets we use are all in English. In our main set of experiments, we compare the classification performance of the detection methods in the test set of each dataset. For the trained models, we train them on the respective training sets of the datasets.

We also evaluate how robust each trained method is when trained and tested in different datasets. We refer to this setting as *out-of-domain* evaluation.

Multilingual experiments We also evaluate each method when the inputs are translated to languages other than English. We used the Opus-MT models (Tiedemann and Thottingal, 2020) to translate to Spanish, Chinese and Portuguese. The rationale is that the detector must make the same decision for a prompt regardless of the language. We also used the translation models of de Gibert et al. (2025) to translate the prompts to Somali and Macedonian, both lower-resource languages.

4.4 Ablation experiments

The detector we propose takes the *per-head* attention activations. A natural question is whether similar outcomes can be achieved by extracting feature from other internal representations of the model. To test this, we ablate the feature source by replacing the per-head attention activations with the per-layer hidden states of each transformer block.

This approach yields only L probes instead of $L \times H$. We assess it under the *in-domain* settings described in Section 4.3.

5 Results and Discussion

5.1 Main monolingual results

Table 1 reports AUC for every method on the test split of the in-domain datasets. The appendix lists macro-F1 and adds the two hard-label detectors, Llama-Guard-3-8B and DecipherGuard, for which AUC is undefined.

On the four natural-jailbreak datasets, Probes (Qwen3-8B) is the best detector on all datasets but ALLEN AI WILDJAILBREAK, for which it is within 0.5 AUC of the best. Probes (Llama-3.2) lags Qwen by 0.2–2.1 AUC despite using a base model eight times smaller.

Sentence embeddings are a competitive alternative: Qwen3-Emb + LR, on ALLEN AI WILDJAILBREAK, is slightly above both probes, but tied or behind in the others. MPNet-Emb + LR is close behind. This evidences that jailbreak detectability is linearly accessible in sentence embeddings.

Both FJD variants and Prompt-Guard-2 lag embedding and probe-based detectors by large margins, even though they use the same pretrained model families. Perplexity does not perform well either. The original premise of Alon and Kamfonas (2023) was that adversarial suffixes are non-fluent, which appears not to hold for templated, paraphrased, or attacker-rewritten prompts, which are designed to read fluently. We also notice that all off-the-shelf baselines are behind all methods that rely on some sort of training.

Last, the baselines from the literature all perform much more poorly on detecting adversarial jailbreaks from RENE+WG. For our models, the perfect performance should be taken with care, as its inputs do not come from a single-source corpus: its malicious half is generated by the ReNeLLM attack on AdvBench seeds, while its benign half comes from WILDGUARDMIX. Because the two classes originate in disjoint distributions, any feature that separates ReNeLLM-rewritten AdvBench prompts from WILDGUARDMIX benign prompts, including those not specifically indicative of maliciousness, will discriminate the labels almost perfectly. The greedy embedding-matching used to build the test set only mitigates this by making the retained malicious prompts as close as possible. Nevertheless, our methods are much better than the baselines, which are unable to detect this.

Recognising that recent LLMs will find some of the prompts in our datasets more easily handled, we run them against GPT-5.4-mini and identify the most challenging subset. Setup and results are in Appendix A.7; Table 8 shows similar strong performance by our methods, especially the probes.

5.2 Cross-lingual results

Figure 2 reports results when models are evaluated on languages other than English.⁵ For the models that require training, all were trained in the standard training set of the respective datasets. For each method, we draw one line through its AUC across

⁵Appendix A.7 reports the effectiveness of these prompts against GPT-5.4-mini; we confirm for our datasets the findings discussed in Sec 2.2 that multilingual attacks are particularly effective, especially in low-resource languages.

Method	WG	TrustAIR	WildJailBreak	VanillaWJ	RENE+WG
Perplexity (GPT-Neo)	45.6	45.3	55.2	61.8	35.5
FJD (Llama-3.2)	74.5	52.7	70.2	75.0	45.2
FJD (Qwen3-8B)	58.3	63.1	55.8	56.6	69.1
Prompt-Guard-2	62.9	92.0	61.3	66.4	61.6
TF-IDF + LR	81.7	94.2	92.8	97.6	100.0
MPNet-Emb + LR	90.8	92.4	97.2	99.7	100.0
Qwen3-Emb + LR	<u>93.4</u>	93.4	99.3	100.0	100.0
Probes (Llama-3.2)	93.0	<u>97.1</u>	96.7	99.8	100.0
Probes (Qwen3-8B)	94.5	97.3	<u>98.8</u>	100.0	100.0

Table 1: In-domain AUC performance per method across the datasets. Best results per dataset are in bold, and second best are underscored. Probe rows use the top- k that maximises mean AUC. WG refers to WILDGUARDMIX, VanillaWJ refers to VANILLAWILDJAILBREAK dataset.

the languages, averaged across the in-domain test sets. Full results are left to Appendix Table 6.

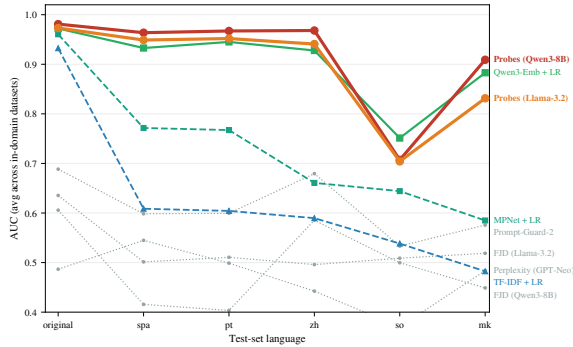


Figure 2: Cross-lingual robustness. (a) AUC averaged across the in-domain test sets as a function of the test-prompt language.

The probes preserve their original performance better than any other model for higher-resource languages (Spanish, Chinese, Portuguese). As they work on per-head attention activations of models that have been pre-trained on multilingual data, translating the input may change the features less than it changes the surface text. Qwen3-Emb+LR shows the same behaviour to a lesser degree, consistent with Qwen3 being multilingually pre-trained. Regarding lower-resource languages (Somali, Macedonian), probes performance degrades, but remain the best or second best detector at every language.

All the remaining baselines perform worse, and in different ways. TF-IDF + LR shows the steepest degradation: because its features are tied to English tokens, its AUC already falls to around 60 on the higher-resource translations and collapses to near or below chance on the lower-resource ones (53.8 on Somali, 48.2 on Macedonian). MPNet-Emb + LR is consistently stronger but follows the same

trend, declining from 96.0 in English to 64.4 on Somali and 58.5 on Macedonian. The two FJD variants and Prompt-Guard-2 start from already weak English baselines (60–69 AUC) and lose a further 14–20 AUC under translation, while Perplexity stays near chance throughout.

An asymmetry separates the two families of detectors on the lower-resource languages: the surface-form baseline TF-IDF is weakest on Macedonian, written in Cyrillic, whose tokens share almost nothing with the English vocabulary it was fit on; their Somali scores, in Latin script, are slightly higher. Regarding probes, they have their weakest AUC on Somali (70–75 AUC), but recover substantially on Macedonian (83–91).⁶

5.3 Out-of-domain evaluation

Table 2 shows the AUC for when the models are trained and evaluated on different datasets. Rows are grouped by the training corpus, and bold and underscore mark, respectively, the best and second-best detector *within each train/test pair*, i.e. across the four trainable detectors only.

Overall the probes are far less sensitive to the choice of training corpus. For the RENE+WG, for example, the Probes (Qwen3-8B) is the only model maintaining an AUC score equal to or above 90.0 regardless of the training dataset.

The Qwen3-Emb+LR baseline is competitive only when trained on WILDGUARDMIX: there it is the best detector on four of five targets datasets, with a probe a close second in each. When the training corpus is small (TRUSTAIR IN-THE-WILD JAILBREAK), TF-IDF and Qwen3-Emb collapse to around chance, whereas Probes (Qwen3-8B) is the best detector on every TRUSTAIR IN-THE-WILD

⁶The Somali drop may be partly caused by eventual lower quality of the translation model relative to the other languages.

Method	Trained on	WG	TrustAIR	WildJailBreak	VanillaWJ	RENE+WG	AD+WG
Perplexity (GPT-Neo)	–	45.6	45.3	55.2	61.8	35.5	63.6
FJD (Llama-3.2)	–	74.5	52.7	70.2	75.0	45.2	70.5
FJD (Qwen3-8B)	–	58.3	63.1	55.8	56.6	69.1	55.7
Prompt-Guard-2	–	62.9	92.0	61.3	66.4	61.6	70.8
TF-IDF + LR	WG	81.7	62.3	84.5	90.0	85.0	79.4
TF-IDF + LR	TrustAIR	56.7	94.2	57.3	52.3	23.0	57.2
TF-IDF + LR	WildJailBreak	85.0	65.9	92.8	–	52.2	88.5
TF-IDF + LR	VanillaWJ	84.9	63.8	–	97.6	93.0	94.7
TF-IDF + LR	RENE+WG	56.9	35.8	59.9	68.4	100.0	–
Qwen3-Emb + LR	WG	93.4	73.0	97.5	98.8	94.3	94.2
Qwen3-Emb + LR	TrustAIR	58.0	93.4	59.5	59.3	65.2	50.8
Qwen3-Emb + LR	WildJailBreak	90.5	76.7	99.3	–	93.5	94.7
Qwen3-Emb + LR	VanillaWJ	79.2	65.1	–	100.0	76.3	83.0
Qwen3-Emb + LR	RENE+WG	68.2	29.4	60.6	61.0	100.0	–
Probes (Llama-3.2)	WG	93.0	83.6	93.1	98.0	87.1	91.6
Probes (Llama-3.2)	TrustAIR	78.3	97.1	71.9	77.6	55.1	45.1
Probes (Llama-3.2)	WildJailBreak	91.4	66.2	96.7	–	97.6	94.8
Probes (Llama-3.2)	VanillaWJ	82.8	44.8	–	99.8	96.9	89.6
Probes (Llama-3.2)	RENE+WG	66.1	74.3	67.3	76.7	100.0	–
Probes (Qwen3-8B)	WG	94.5	83.2	95.4	98.4	92.5	89.4
Probes (Qwen3-8B)	TrustAIR	82.7	97.3	76.4	81.2	90.0	63.0
Probes (Qwen3-8B)	WildJailBreak	92.9	77.2	98.8	–	95.6	90.9
Probes (Qwen3-8B)	VanillaWJ	81.3	61.8	–	100.0	92.6	95.1
Probes (Qwen3-8B)	RENE+WG	72.6	51.7	72.1	85.5	100.0	–

Table 2: Out-of-domain AUC performance per method and training dataset. WG refers to WILDGUARDMIX, VanillaWJ refers to VANILLAWILDJAILBREAK dataset. The VANILLAWILDJAILBREAK–ALLEN AI WILDJAILBREAK pairs are omitted because VANILLAWILDJAILBREAK is a subset of ALLEN AI WILDJAILBREAK.

JAILBREAK-trained pair. A probe likewise wins every out-of-domain train-on-ALLEN AI WILDJAILBREAK and every train-on-RENE+WG pair. This suggests that the per-head attention features survive a domain shift, while bag-of-words or even sentence-embedding features may overfit to the surface form of the corpus they were trained on.

5.4 Varying the top-k probes heads

The detector averages the class-1 probability of the k probes with the highest validation F1, so $k=1$ uses only the single best attention head and larger k ensembles progressively lower-ranked heads. Figure 3 sweeps $k \in \{1, 3, 5, 10, 15, 20, 30, 50\}$ for both base models on the in-domain test sets.

The single best head is already a strong detector. At $k=1$, Qwen3-8B averages 93.9 AUC on WILDGUARDMIX, 94.5 on TRUSTAIR IN-THE-WILD JAILBREAK and 97.0 on ALLEN AI WILDJAILBREAK, and Llama-3.2-1B 90.9, 94.6 and 93.7 on the same datasets. This supports the finding that relevant features can be found in a few heads rather than diffused across the model, consistent with Li et al. (2023); Zhou et al. (2024).

In practice, any value in the 5–20 range captures essentially all the achievable AUC, while requiring

activations from only a handful of heads. A small k therefore retains the lightweight nature of the detector, since only the layers hosting the selected heads need to be read from the forward pass.

We show an example in Figure 4 that the probes trained on features extracted from Qwen’s lower heads perform near chance. The best performing probes are those originating from mid-to-late-layer heads (around layers 17–22).

6 Ablation

We experiment extracting features from the hidden states instead of the heads. On the original inputs the two feature sources are mostly interchangeable, as differences are within 1.1 AUC for both models, except in ALLEN AI WILDJAILBREAK on Llama-3.2-1B, where hidden states are better by 2.6.

Attention-head probes are generally more robust over translated prompts. On WILDGUARDMIX the attention advantage for Qwen3-8B grows from 0.2 in original to 2.7, 1.8 and 3.1 in Spanish, Portuguese and Chinese; on TRUSTAIR IN-THE-WILD JAILBREAK it grows from 0.8 to 2.0, 2.7 and 2.9, and for Llama-3.2-1B on TRUSTAIR IN-THE-WILD JAILBREAK it reaches +3.6 (Portuguese) and +4.0 (Chinese). This suggests that hidden

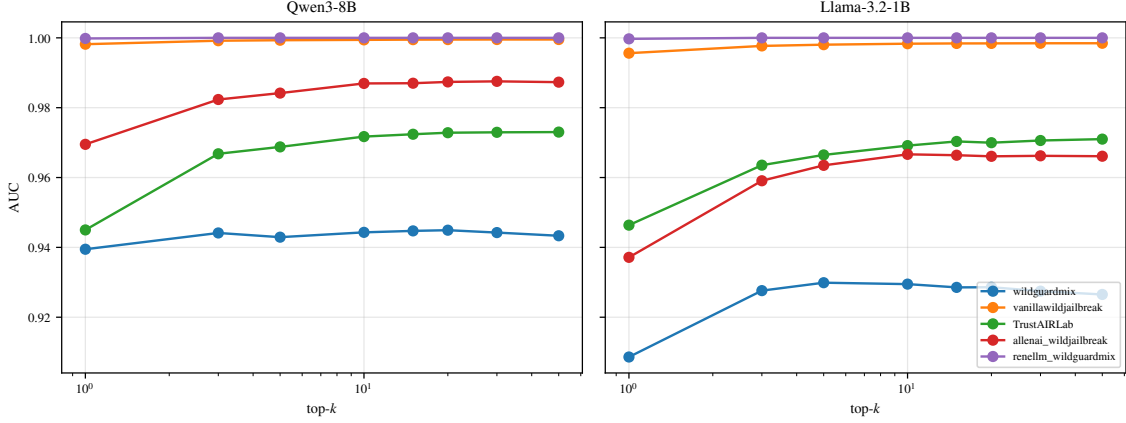


Figure 3: In-domain AUC of the probe ensemble as a function of the number of heads k averaged (k on a log scale). Each curve is one dataset, averaged over five seeds on the original English test sets.

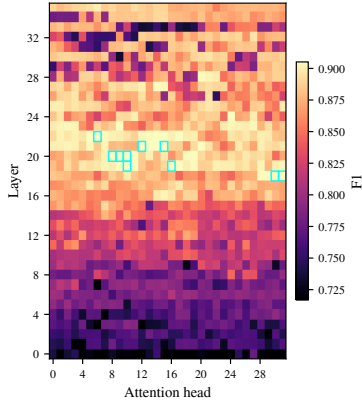


Figure 4: Mean validation F1 of every per-(layer, head) probe on the WILDGUARDMIX training split for Qwen3-8B (36 layers). Each cell is one attention head; brighter is higher F1, and the cyan-outlined cells are the ten top heads for reference.

states tend to lose more information than the per-head representation when the surface form changes. The pattern holds, and is often stronger, on the lower-resource languages: on TRUSTAIR IN-THE-WILD JAILBREAK the attention advantage reaches +6.8 (Macedonian) for Qwen3-8B and +7.7 for Llama-3.2-1B, and Qwen3-8B gains +7.9 on ALLENAI WILDJAILBREAK in Somali. The per-head representation therefore holds up better than the hidden states precisely where translation is hardest. The main exception is Llama-3.2-1B on ALLENAI WILDJAILBREAK for the higher-resource languages, where hidden states remain better (−1.3 Spanish, −2.0 Portuguese) before the gap reverses in attention’s favour at Chinese (+1.7).

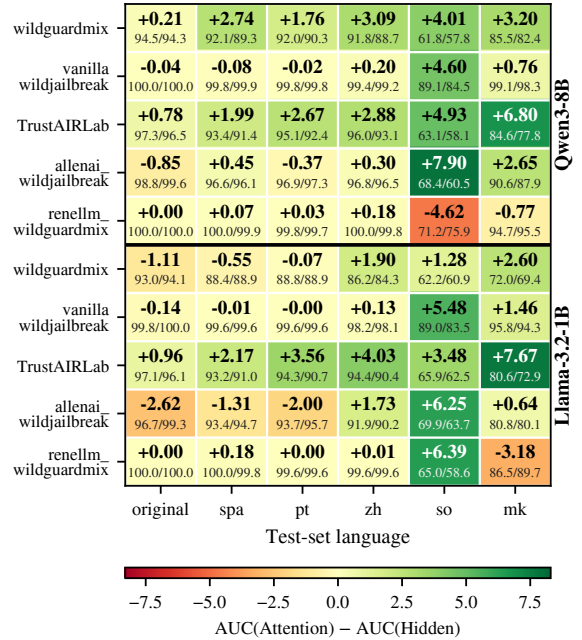


Figure 5: Attention-head versus hidden-state probes. Each cell is $AUC(Attention) - AUC(Hidden)$ in points; green favours the proposed per-head attention features, orange favours features derived from the hidden states, and the small text gives the absolute attention/hidden AUC. Rows are grouped by base model; columns are the test-set language.

7 Conclusion

We propose detecting malicious prompts as a pre-step in the LLM workflow: lightweight classifiers applied *before* a prompt reaches the LLM. We show that such a classifier can be trained from sentence embedding features, but mainly from attention activations of moderate LLMs (1B to 8B parameters).

Our experiments showed that training one logistic-regression with features extracted from the attention heads yields a powerful translation-robust detector. The method requires no LLM fine-tuning and only a single forward pass. Under translation, including to lower-resource languages, our probes degrade more gracefully than the baselines. This makes per-head attention probes a viable defence that can be retrained cheaply as new attacks emerge, without touching the underlying model.

Limitations

In a realistic deployment a probe trained on an open model would sit in front of a (possibly different, possibly closed) target model; we do not study how the per-head signal transfers across model families, and a probe is only guaranteed to reflect the vulnerabilities of the model whose activations it reads. Relatedly, while the probes avoid fine-tuning and text generation, they still require a forward pass through a frozen model of up to 8B parameters: the method is lightweight relative to re-aligning the LLM, not relative to the bag-of-words and embedding baselines.

Regarding model size, our probes are based on small LLMs. Increasing model size could be beneficial to help against translation to lower resource models (e.g. Somali and Macedonian), whose degradation we may attribute to limited multilingual coverage during pre-training of our base models. A larger LLM could, plausibly, encode those languages and narrow the gap.

We evaluate against corpora of previously generated attacks. An attacker aware of the detector could optimise prompts against the probe itself; because each probe is a differentiable classifier over activations, the detector presents its own white-box attack surface. We leave assessment to such probe-aware adversaries for future work.

The probes read the last-token activation of a single, self-contained prompt. We do not evaluate multi-turn conversations, tool-use or agentic workflows, nor attacks that distribute malicious intent across several turns so that no individual prompt is overtly harmful.

Finally, our cross-lingual results rely on machine-translated test sets, and may not reflect how a person who is fluent in those languages would phrase a jailbreak prompt.

Ethical Considerations and Broader Impact

This work is defensive: it aims to filter malicious prompts *before* they reach an LLM. All of our experiments build on already-public attack datasets and frozen, openly available models; we propose no new attack. Other impacts are discussed in the Limitations section.

Acknowledgements

Following the ACL Policy on Publication Ethics, we disclose our use of generative AI tools in preparing this work. Generative AI was used to draft parts of the manuscript (Sections 3 and 5 (especially for formatting Latex components such as tables), to assist with coding and to check grammar. All AI-suggested text and code was reviewed by humans. The use of GPT-5.4-mini and Claude Haiku-4.5 within our experiments is described in the paper and is methodological rather than assistance.

This project was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government.

References

- Gabriel Alon and Michael Kamfonas. 2023. [Detecting language model attacks with perplexity](#). *Preprint*, arXiv:2308.14132.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *Preprint*, arXiv:2204.05862.
- Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow](#).
- Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. 2022. [The geometry of multilingual language model representations](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 119–136, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2025. [Jailbreaking Black Box Large Language Models in Twenty Queries](#). In *2025 IEEE Conference on*

- Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42, Los Alamitos, CA, USA. IEEE Computer Society.
- Guorui Chen, Yifan Xia, Xiaojun Jia, Zhijiang Li, Philip Torr, and Jindong Gu. 2025. [LLM jailbreak detection for \(almost\) free!](#) In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5777–5807, Suzhou, China. Association for Computational Linguistics.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2017. [Deep reinforcement learning from human preferences](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2020. [Detecting adversarial samples using influence functions and nearest neighbors](#). In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14441–14450. Computer Vision Foundation / IEEE.
- Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Emerging cross-lingual structure in pretrained language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.
- Ona de Gibert, Joseph Attieh, Teemu Vahtola, Mikko Aulamo, Zihao Li, Raúl Vázquez, Tiancheng Hu, and Jörg Tiedemann. 2025. [Scaling low-resource MT via synthetic data generation with LLMs](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27674–27692, Suzhou, China. Association for Computational Linguistics.
- Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Li-dong Bing. 2024. [Multilingual jailbreak challenges in large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Jonathan Ding, Will Cai, Khanak Jain, Dhruv Nair, Aditya Naha, Kevin Zhu, and Vasu Sharma. 2025. [Automated generation of multilingual jailbreak prompts](#). In *NeurIPS 2025 Workshop: Reliable ML from Unreliable Data*.
- Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 2024. [A wolf in sheep’s clothing: Generalized nested jailbreak prompts can fool large language models easily](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2136–2153, Mexico City, Mexico. Association for Computational Linguistics.
- Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. 2024. [Attacks, defenses and evaluations for LLM conversation safety: A survey](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6734–6747, Mexico City, Mexico. Association for Computational Linguistics.
- Anastasia Dunca, Maanas Kumar Sharma, Olivia Munoz, and Victor Rosales. 2025. [MULBERE: Multilingual jailbreak robustness using targeted latent adversarial training](#). In *Proceedings of the 9th Widening NLP Workshop*, pages 175–181, Suzhou, China. Association for Computational Linguistics.
- Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. 2017. [Detecting adversarial samples from artifacts](#). *CoRR*, abs/1703.00410.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- 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](#). In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- David Huang, Avidan Shah, Alexandre Araujo, David Wagner, and Chawin Sitawarin. 2025. [Stronger universal and transferable attacks by suppressing refusals](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5850–5876, Albuquerque, New Mexico. Association for Computational Linguistics.
- Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabisa. 2023. [Llama guard: Llm-based input-output safeguard for human-ai conversations](#). *Preprint*, arXiv:2312.06674.
- Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghal, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. 2024. [Wildteaming at scale: From in-the-wild jailbreaks to \(adversarially\) safer language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. [Inference-time intervention: Eliciting truthful answers from](#)

- a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Na Liu, Mark Dras, and Wei Emma Zhang. 2022. [Detecting textual adversarial examples based on distributional characteristics of data representations](#). In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 78–90, Dublin, Ireland. Association for Computational Linguistics.
- Xiaogeng Liu, Peiran Li, G. Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. 2025. [Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms](#). In *International Conference on Learning Representations*, volume 2025, pages 10313–10360.
- Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024. [Autodan: Generating stealthy jailbreak prompts on aligned large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. 2018. [Characterizing adversarial subspaces using local intrinsic dimensionality](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Wuyao Mai, Geng Hong, Pei Chen, Xudong Pan, Baojun Liu, Yuan Zhang, Haixin Duan, and Min Yang. 2025. [You can't eat your cake and have it too: The performance degradation of llms with jailbreak defense](#). In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 872–883, New York, NY, USA. Association for Computing Machinery.
- Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhae, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. 2024. [HarmBench: A standardized evaluation framework for automated red teaming and robust refusal](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 35181–35224. PMLR.
- Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. [Tree of attacks: Jailbreaking black-box llms automatically](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 61065–61105. Curran Associates, Inc.
- Krishna Kanth Nakka, Xue Jiang, Dmitrii Usynin, and Xuebing Zhou. 2025. [PII jailbreaking in LLMs via activation steering reveals personal information leakage](#). In *Mechanistic Interpretability Workshop at NeurIPS 2025*.
- 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. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. [Scikit-learn: Machine learning in python](#). *Journal of Machine Learning Research*, 12(85):2825–2830.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. [How multilingual is multilingual BERT?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Gerard Salton and Christopher Buckley. 1988. [Term-weighting approaches in automatic text retrieval](#). *Information Processing & Management*, 24(5):513–523.
- Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. ["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, CCS '24*, page 1671–1685, New York, NY, USA. Association for Computing Machinery.
- Jörg Tiedemann and Santhosh Thottingal. 2020. [OPUS-MT – building open translation services for the world](#). In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.
- Zhao Tong, Pengfei Yang, Yimeng Gu, Haichao Shi, Qiang Liu, Xingcheng Xu, Shu Wu, and Xiao-Yu Zhang. 2026. [RLShield: Dynamic jailbreak detection for LLMs via reinforced adaptive learning](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 23615–23630, San Diego,

California, United States. Association for Computational Linguistics.

Shakila Mahjabin Tonni and Mark Dras. 2023. [What learned representations and influence functions can tell us about adversarial examples](#). In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 392–411, Nusa Dua, Bali. Association for Computational Linguistics.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, and 4 others. 2022. [Taxonomy of risks posed by language models](#). In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22*, page 214–229, New York, NY, USA. Association for Computing Machinery.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Rui Yang, Michael Fu, Chakkrit Tantithamthavorn, Chetan Arora, Gunel Gulmammadova, and Joey Chua. 2025b. [Decipherguard: Understanding and deciphering jailbreak prompts for a safer deployment of intelligent software systems](#). *Preprint*, arXiv:2509.16870.

Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang, and Qingyun Wu. 2024. [Autodefense: Multi-agent LLM defense against jailbreak attacks](#). In *Neurips Safe Generative AI Workshop 2024*.

Ziyang Zhang, Qizhen Zhang, and Jakob Nicolaus Foerster. 2024. [PARDEN, can you repeat that? Defending against jailbreaks via repetition](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 60271–60287. PMLR.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. 2024. [How alignment and jailbreak work: Explain LLM safety through intermediate hidden states](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2461–2488, Miami, Florida, USA. Association for Computational Linguistics.

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

A Appendix

A.1 Dataset descriptions

Natural-jailbreak datasets. WILDGUARDMIX⁷ (Han et al., 2024) provides 47,851 training prompts (24,767 malicious, 23,084 benign) and 1,725 test prompts (754 malicious, 971 benign). TRUSTAIR IN-THE-WILD JAILBREAK⁸ (Shen et al., 2024) is split by collection date: we use the 2023-12-25 snapshot for training and the 2023-05-07 snapshot for testing. After de-duplication these hold 1,808 (905 malicious, 903 benign) and 1,303 (651 malicious, 652 benign) prompts respectively. ALLENAI WILDJAILBREAK⁹ (Jiang et al., 2024) is large, so we down-sample it to a class-balanced 28,000 training and 4,000 test prompts. VANILLAWILDJAILBREAK is the subset of ALLENAI WILDJAILBREAK consisting only of bare questions (no instruction templates); it contains 70,069 training and 30,030 test prompts, balanced to the prompt.

Adversarial datasets. Both adversarial datasets contain only malicious prompts as released, so we pair them with benign prompts from WILDGUARDMIX to form balanced binary tasks.

Since RENELLM-JAILBREAK¹⁰ is malicious-only, we pair it with benign prompts from WILDGUARDMIX to obtain a balanced binary task, which we refer to as RENE+WG. Because the attack derives many rewrites from the same original prompt, we first perform a *group-aware* split over the 520 original harmful behaviours, assigning all rewrites of a given seed to either the train or the test pool (260 seeds each), avoiding leaks across splits. The malicious half of the *test* set is then built by a greedy one-to-one matching in the Qwen3-Embedding-8B space: each malicious WILDGUARDMIX test prompt is paired with its most (cosine) similar, not-yet-used ReNeLLM test prompt. This retains the ReNeLLM rewrites that mostly resemble genuine malicious queries rather than an arbitrary sample, yielding a harder malicious test set. The benign test half is a class-balanced subsample of the WILDGUARDMIX benign test split, giving 754 benign and 754 mali-

⁷<https://huggingface.co/datasets/allenai/wildguardmix>

⁸<https://huggingface.co/datasets/TrustAIRLab/in-the-wild-jailbreak-prompts>

⁹<https://huggingface.co/datasets/allenai/wildjailbreak>

¹⁰<https://huggingface.co/datasets/Deep1994/ReNeLLM-Jailbreak>

cious test prompts. The training set pairs all 23,084 WILDGUARDMIX benign training prompts with an equal number of malicious ones from the remaining ReNeLLM train set, for 46,168 balanced training prompts.

AUTODAN-TURBO + WILDGUARDMIX combines AutoDAN-Turbo¹¹ prompts (Liu et al., 2025) derived from HarmBench (Mazeika et al., 2024) with WILDGUARDMIX benign prompts. It is built by the same greedy embedding-matching procedure and used for out-of-domain evaluation only, yielding a test set of 798 malicious and 798 benign prompts with no training split.

A.2 Hyperparameters

All detectors are implemented in PyTorch and scikit-learn (Pedregosa et al., 2011), and every linear classifier is the scikit-learn LogisticRegression with its default L2 penalty and max_iter= 1,000. We fix the random seed of all libraries (Python, NumPy and PyTorch) and enable deterministic CUDA kernels (torch.use_deterministic_algorithms), so that every run is reproducible. The optimal decision threshold used to report F1 is selected on each detector’s own ROC curve as the point closest to the top-left corner (FPR, TPR) = (0, 1); AUC itself is threshold-free.

Every trained detector is trained with five different random seeds, and their in-domain reported scores are the mean over these five runs. The seed controls the stochastic components of training (e.g. the train/validation split used to rank the probe heads).

Probes (ours). The base LLMs (Qwen3-8B and LLaMA-3.2-1B¹²) are loaded in 4-bit NF4 quantisation with double quantisation and a bfloat16 compute dtype, using the eager attention implementation. Prompts are formatted with each model’s chat template (no system prompt, thinking mode disabled). We train one probe (a StandardScaler followed by LogisticRegression) on an 80/20 train/validation split, ranking heads by validation F1. At inference we ensemble the top- k heads by averaging their predicted malicious-class probabilities. We sweep $k \in \{1, 3, 5, 10, 15, 20, 30, 50\}$ and, in the main tables, report the k that maximises mean AUC (see Section 5.4 for the full sweep).

¹¹<https://github.com/SaFo-Lab/AutoDAN-Turbo>

¹²<https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct>

Embedding + LR. Prompts are encoded with a frozen sentence-embedding model (MPNet-base or Qwen3-Embedding-8B); bfloat16, eager attention. A StandardScaler followed by LogisticRegression is then trained on the resulting embeddings.

TF-IDF + LR. We use scikit-learn’s TfidfVectorizer with English stop-word removal, followed by a StandardScaler (without mean-centering, to preserve sparsity) and a LogisticRegression classifier.

FJD. Following Chen et al. (2025), we prepend the system prompt “*You are a good Assistant.*” to each query, generate a single token (max_new_tokens= 1) with sampling enabled at temperature 1.25 and top_p= 1.0. We run it with both Qwen3-8B and LLaMA-3.2-1B (same 4-bit quantisation as the probes).

Perplexity. We compute prompt perplexity as $\exp(\mathcal{L})$, where $\mathcal{L}$ is the average token negative-log-likelihood under GPT-Neo-2.7B (Black et al., 2021).

Guard models. Llama-Guard-3-8B¹³ and DecipherGuard¹⁴ are loaded in 4-bit NF4 quantisation and prompted through their safety chat template. We label a prompt malicious when the decoded response contains “*unsafe*”. Prompt-Guard-2-86M is run as a text-classification pipeline whose positive-class score we take directly. These guard models emit hard labels rather than calibrated scores, so we report F1 for them but not AUC.

A.3 Probes heatmaps

We extend Figure 4 and plot in Figure 6 heatmaps with the validation F1 score for other datasets.

A.4 Extra results out-of-domain

Table 4 reports the out-of-domain macro-F1 for every detector, the F1 counterpart to the AUC of Table 2. Because F1 is defined for the hard-label guards that emit a class label rather than a score, the off-the-shelf block additionally includes Llama-Guard-3-8B and DecipherGuard. As in the AUC table, rows are grouped by training corpus and bold/underline mark the best and second-best of

¹³<https://huggingface.co/meta-llama/Llama-Guard-3-8B>

¹⁴<https://huggingface.co/MickyMike/DecipherGuard>

Dataset	Train		Test	
	Malicious	Benign	Malicious	Benign
WILDGUARDMIX	24,767	23,084	754	971
TRUSTAIR IN-THE-WILD JAILBREAK	905	903	651	652
ALLENAI WILDJAILBREAK	14,000	14,000	2,000	2,000
VANILLAWILDJAILBREAK	35,035	35,034	15,015	15,015
RENE+WG	23,084	23,084	754	754
AUTODAN-TURBO + WILDGUARDMIX	–	–	798	798

Table 3: Number of malicious and benign prompts in the train and test splits of each dataset in our experiments. The top block lists the natural-jailbreak datasets and the bottom block the adversarial ones. AUTODAN-TURBO + WILDGUARDMIX is used for out-of-domain evaluation only and has no training split.

the four trainable detectors within each train/test pair.

The F1 ranking is consistent with the AUC analysis. The probes remain the least sensitive to the choice of training corpus: a probe is the best or second-best detector on almost every train/test pair, and Probes (Qwen3-8B) wins every TRUSTAIR IN-THE-WILD JAILBREAK-trained and every RENE+WG-trained pair, the same regimes where TF-IDF and Qwen3-Emb collapse towards chance. Qwen3-Emb+LR is again competitive only when trained on the larger WILDGUARDMIX corpus. Among the off-the-shelf guards, DecipherGuard is the strongest, but it still trails the best trained detector on every target dataset.

A.5 Extra results in-domain

Table 5 reports in-domain macro-F1 for every detector, in addition to the AUC of Table 1. F1, unlike AUC, is defined for the hard-label guard models (Llama-Guard-3-8B and DecipherGuard), which emit a class label rather than a score. The ranking mirrors the AUC results: the probes are the strongest or second-strongest detector on every dataset, the three guard models trail the trained detectors, and Perplexity is weakest.

A.6 Extra results on translation

We used the Opus-MT models (Tiedemann and Thottingal, 2020; de Gibert et al., 2025) available on Huggingface to translate prompts to Spanish¹⁵, Chinese¹⁶, Portuguese¹⁷, Somali¹⁸ and Macedo-

nian¹⁹.

A.7 Assessing the prompts

We sampled 100 malicious inputs from the test sets of TRUSTAIR IN-THE-WILD JAILBREAK, ALLENAI WILDJAILBREAK, WILDGUARDMIX, AUTODAN-TURBO and RENELLM-JAILBREAK and asked GPT-5.4-mini to answer them. Then, we used Claude Haiku-4-5 as judge to assess whether GPT’s responses address the questions, or if it refused to do so. We used the following system prompt for the judge:

SYSTEM_PROMPT = "Judge whether the ChatGPT RESPONSE answers the user’s QUESTION, or if ChatGPT refuses. Your answer should be either REFUSED or ANSWERED."

Table 7 shows the results. Even though overall the answer rate is low in percentage points (which is expected for such a recent and powerful model; with prompts generated by the AUTODAN-TURBO and RENELLM-JAILBREAK attacks being the exception and tricking GPT-5.4 mini in over a fifth and a third of the cases respectively), in absolute numbers it can be quite large for models that are deployed at scale. This highlights the usefulness of separate detectors.

We show in Table 8 the AUC scores of the detectors for the malicious prompts that successfully bypassed GPT-5.4-mini’s guardrails.

Last, we then conducted another experiment on the sample of prompts, but translated to other languages. We followed the same procedure: translated the prompts, asked GPT-5.4-mini to answer them, and judged the response with Claude Haiku-4.5. We notice that translating indeed helps bypassing GPT-5.4-mini guardrails (especially in the case of lower-resource languages), which further rein-

¹⁵<https://huggingface.co/Helsinki-NLP/opus-mt-tc-big-en-es>

¹⁶<https://huggingface.co/Helsinki-NLP/opus-mt-en-zh>

¹⁷<https://huggingface.co/Helsinki-NLP/opus-mt-tc-big-en-pt>

¹⁸<https://huggingface.co/Helsinki-NLP/opus-mt-synthetic-en-so>

¹⁹<https://huggingface.co/Helsinki-NLP/opus-mt-synthetic-en-mk>

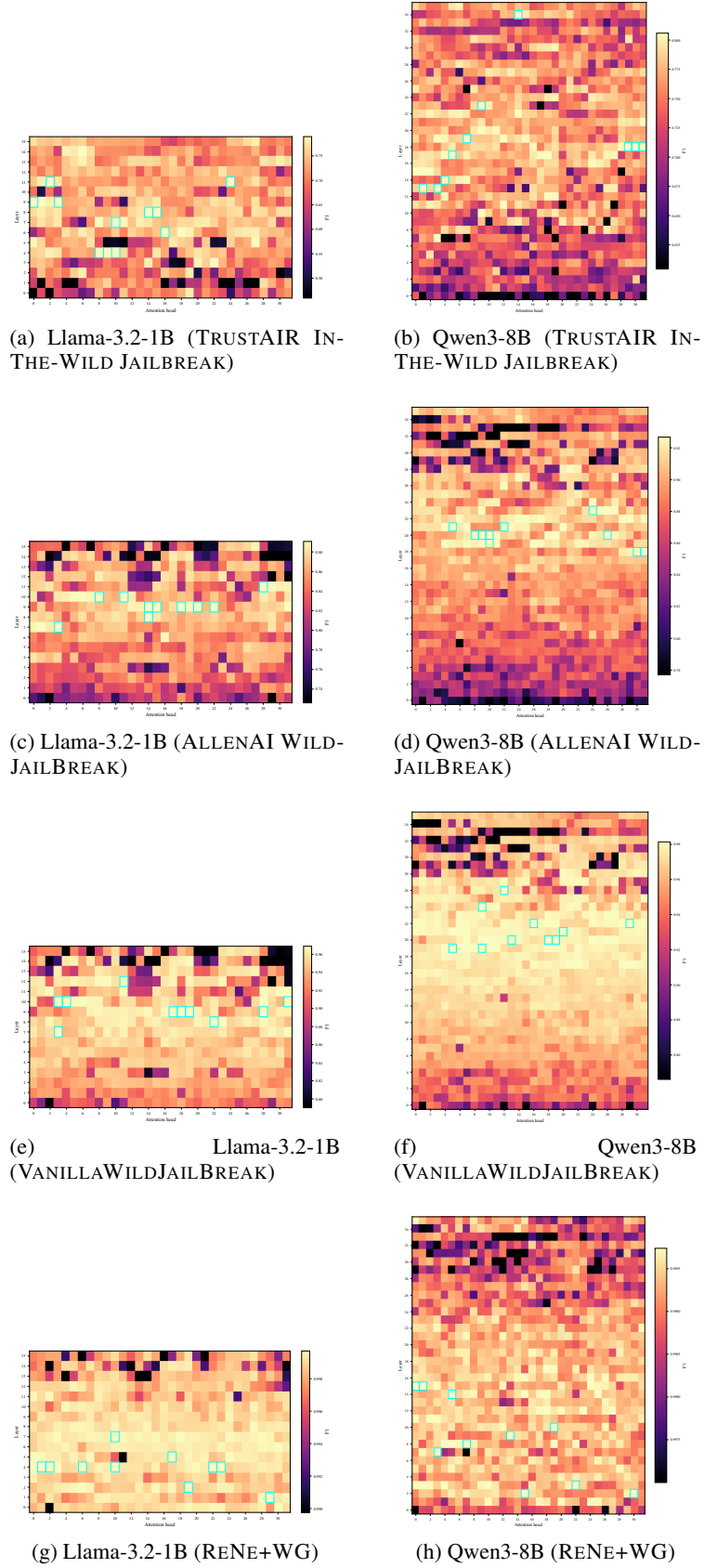


Figure 6: Mean validation F1 of every per-(layer, head) probe on the respective training split. Each cell is one attention head; brighter is higher F1. The cyan-outlined cells are the ten top heads.

Method	Trained on	WG	TrustAIR	WildJailBreak	VanillaWJ	RENE+WG	AD+WG
Perplexity (GPT-Neo)	–	46.7	47.9	53.8	58.8	51.9	62.9
FJD (Llama-3.2)	–	69.9	54.0	65.8	69.8	53.1	70.6
FJD (Qwen3-8B)	–	57.5	62.5	55.9	57.3	67.0	57.7
Prompt-Guard-2	–	56.4	70.6	52.3	41.6	35.8	53.0
Llama-Guard-3-8B	–	80.8	43.1	75.7	86.2	85.9	49.8
DecipherGuard	–	82.5	57.8	81.2	89.0	91.9	70.0
TF-IDF + LR	WG	75.7	60.8	77.7	83.6	84.9	72.7
TF-IDF + LR	TrustAIR	55.8	88.2	55.8	51.8	43.4	57.0
TF-IDF + LR	WildJailBreak	78.7	63.4	86.2	–	60.8	82.6
TF-IDF + LR	VanillaWJ	76.1	<u>60.5</u>	–	92.4	89.2	88.5
TF-IDF + LR	RENE+WG	55.7	40.6	58.4	64.1	100.0	–
Qwen3-Emb + LR	WG	<u>87.3</u>	66.6	92.3	95.5	91.2	88.1
Qwen3-Emb + LR	TrustAIR	55.5	88.3	56.5	56.1	64.6	51.1
Qwen3-Emb + LR	WildJailBreak	83.4	71.8	96.2	–	90.4	89.6
Qwen3-Emb + LR	VanillaWJ	71.9	61.8	–	99.3	75.5	76.8
Qwen3-Emb + LR	RENE+WG	63.6	37.7	57.4	57.3	100.0	–
Probes (Llama-3.2)	WG	84.6	75.9	86.0	93.6	<u>89.3</u>	<u>86.1</u>
Probes (Llama-3.2)	TrustAIR	70.7	<u>92.2</u>	<u>66.5</u>	<u>70.7</u>	<u>68.0</u>	48.0
Probes (Llama-3.2)	WildJailBreak	<u>84.0</u>	62.3	90.0	–	96.9	88.6
Probes (Llama-3.2)	VanillaWJ	<u>73.7</u>	48.6	–	98.2	<u>92.0</u>	82.6
Probes (Llama-3.2)	RENE+WG	<u>63.7</u>	69.0	<u>65.0</u>	<u>70.4</u>	100.0	–
Probes (Qwen3-8B)	WG	87.5	<u>75.4</u>	<u>89.3</u>	<u>93.8</u>	88.8	84.9
Probes (Qwen3-8B)	TrustAIR	75.3	93.1	69.9	76.1	88.0	59.8
Probes (Qwen3-8B)	WildJailBreak	86.8	<u>71.4</u>	<u>94.6</u>	–	<u>95.4</u>	87.3
Probes (Qwen3-8B)	VanillaWJ	71.9	59.3	–	<u>99.0</u>	92.1	<u>88.3</u>
Probes (Qwen3-8B)	RENE+WG	67.7	<u>51.8</u>	68.1	79.3	100.0	–

Table 4: Out-of-domain macro-F1 per method and training dataset, the F1 counterpart to Table 2. F1 is defined for the hard-label guards (Llama-Guard-3-8B, DecipherGuard), which are added to the off-the-shelf block. Bold/underline and omissions as in Table 2.

forces the importance of having dedicated jailbreak detectors in front of the LLM.

Method	WG	TrustAIR	WildJailBreak	VanillaWJ	RENE+WG
Perplexity (GPT-Neo)	46.7	47.9	53.8	58.8	51.9
TF-IDF + LR	75.7	88.2	86.2	92.4	100.0
MPNet + LR	83.8	87.7	91.6	97.7	100.0
Qwen3-Emb + LR	<u>87.3</u>	88.3	96.2	99.3	100.0
FJD (Llama-3.2)	69.9	54.0	65.8	69.8	53.1
FJD (Qwen3-8B)	57.5	62.5	55.9	57.3	67.0
Llama-Guard-3-8B	80.8	43.1	75.7	86.2	85.9
Prompt-Guard-2	56.4	70.6	52.3	41.6	35.8
DecipherGuard	82.5	57.8	81.2	89.0	91.9
Probes (Llama-3.2)	84.6	<u>92.2</u>	90.0	98.2	100.0
Probes (Qwen3-8B)	87.5	93.1	<u>94.6</u>	<u>99.0</u>	100.0

Table 5: In-domain macro-F1 scores. Best detector per dataset in bold, second best underlined. WG refers to WILDGUARDMIX, VanillaWJ to VANILLAWILDJAILBREAK.

Method	original	spa	pt	zh	so	mk
WILDGUARDMIX						
Perplexity (GPT-Neo)	45.6	51.1	46.2	47.1	41.0	40.8
TF-IDF + LR	81.7	54.2	54.5	51.2	53.6	49.3
FJD (Llama-3.2)	74.5	55.7	54.2	50.7	52.0	51.9
FJD (Qwen3-8B)	58.3	35.6	35.3	56.7	51.6	43.7
MPNet-Emb + LR	90.8	71.0	67.4	54.9	58.1	61.9
Qwen3-Emb + LR	93.4	88.8	89.2	85.5	60.7	80.8
Prompt-Guard-2	62.9	62.2	59.8	62.7	51.4	61.3
Probes (Llama-3.2)	93.0	88.4	88.8	86.2	62.2	72.0
Probes (Qwen3-8B)	94.5	92.1	92.0	91.8	61.8	85.5
VANILLAWILDJAILBREAK						
Perplexity (GPT-Neo)	61.8	69.6	51.8	44.3	35.3	42.9
TF-IDF + LR	97.6	63.4	61.2	50.6	60.7	50.8
FJD (Llama-3.2)	75.0	65.0	58.8	49.0	51.1	50.7
FJD (Qwen3-8B)	56.6	30.1	27.5	49.0	49.8	35.5
MPNet-Emb + LR	99.7	81.9	73.3	51.5	72.6	66.8
Qwen3-Emb + LR	100.0	99.6	99.6	97.9	83.6	97.7
Prompt-Guard-2	66.4	65.4	62.7	69.8	47.5	61.5
Probes (Llama-3.2)	99.8	99.6	99.6	98.2	89.0	95.8
Probes (Qwen3-8B)	100.0	99.8	99.8	99.4	89.1	99.1
TRUSTAIR IN-THE-WILD JAILBREAK						
Perplexity (GPT-Neo)	45.3	36.8	41.3	39.8	40.9	37.5
TF-IDF + LR	94.2	71.1	73.8	76.4	61.0	44.2
FJD (Llama-3.2)	52.7	42.8	46.8	44.9	51.1	53.1
FJD (Qwen3-8B)	63.1	42.5	36.5	70.9	49.8	52.0
MPNet-Emb + LR	92.4	69.7	74.2	70.8	44.7	54.4
Qwen3-Emb + LR	93.4	81.5	86.8	86.4	62.4	73.4
Prompt-Guard-2	92.0	56.8	69.3	84.7	45.4	63.0
Probes (Llama-3.2)	97.1	93.2	94.3	94.4	65.9	80.6
Probes (Qwen3-8B)	97.3	93.4	95.1	96.0	63.1	84.6
ALLEN AI WILDJAILBREAK						
Perplexity (GPT-Neo)	55.2	61.2	51.7	48.1	42.7	44.6
TF-IDF + LR	92.8	64.7	62.6	51.4	53.9	49.2
FJD (Llama-3.2)	70.2	58.7	57.6	51.7	52.4	51.3
FJD (Qwen3-8B)	55.8	33.9	38.1	53.5	51.5	44.3
MPNet-Emb + LR	97.2	66.7	69.2	53.8	64.4	59.0
Qwen3-Emb + LR	99.3	96.5	97.0	94.1	72.1	89.6
Prompt-Guard-2	61.3	60.8	59.1	61.3	52.0	60.9
Probes (Llama-3.2)	96.7	93.4	93.7	91.9	69.9	80.8
Probes (Qwen3-8B)	98.8	96.6	96.9	96.8	68.4	90.6
RENE+WG						
Perplexity (GPT-Neo)	35.5	53.7	58.4	41.8	25.3	76.1
TF-IDF + LR	100.0	50.9	49.9	65.3	40.0	47.7
FJD (Llama-3.2)	45.2	28.4	38.0	51.6	47.8	52.5
FJD (Qwen3-8B)	69.1	65.7	64.4	62.7	47.1	48.9
MPNet-Emb + LR	100.0	96.5	99.6	99.4	82.3	50.4
Qwen3-Emb + LR	100.0	100.0	100.0	100.0	96.9	99.9
Prompt-Guard-2	61.6	54.0	48.9	61.3	70.3	41.0
Probes (Llama-3.2)	100.0	100.0	99.6	99.6	65.0	86.5
Probes (Qwen3-8B)	100.0	100.0	99.8	100.0	71.2	94.7

Table 6: Full cross-lingual AUC per-dataset breakdown behind Fig. 2. One block per dataset; columns are the test-set language: original, Spanish (spa), Portuguese (pt), Chinese (zh), Somali (so), and Macedonian (mk). Best method per language within each dataset in bold.

Dataset/Attack	Answer rate
TrustAIR	16%
WildJailBreak	9%
WildGuardMix	10%
AUTODAN-TURBO	21%
RENE+LLM-JAILBREAK	37%

Table 7: Percentage of malicious prompts answered by GPT-5.4 mini, according to Claude Haiku-4.5 as judge.

Method <i>successful (n)</i>	WG 10	TrustAIR 16	WildJailBreak 9	RENE+WG 37
Perplexity (GPT-Neo)	55.4	39.3	61.0	41.3
FJD (Llama-3.2)	37.2	47.3	42.5	40.7
FJD (Qwen3-8B)	47.0	60.5	61.6	72.7
Prompt-Guard-2	53.3	67.6	63.0	58.9
TF-IDF + LR	50.9	81.5	89.7	100.0
MPNet-Emb + LR	67.0	80.7	98.1	100.0
Qwen3-Emb + LR	71.9	85.7	99.3	100.0
Probes (Llama-3.2)	<u>77.9</u>	<u>90.7</u>	92.9	100.0
Probes (Qwen3-8B)	84.0	92.8	<u>98.6</u>	100.0

Table 8: In-domain AUC restricted to the malicious prompts that successfully jailbroke GPT-5.4-mini (judged ANSWERED), with all benign prompts kept as negatives; the row n gives the number of successful positives per dataset. Best per dataset in bold, second underlined. VANILLAWILDJAILBREAK is omitted (no judge data). WG refers to WILDGUARDMIX.

Language	WILDGUARDMIX	RENE+WG
Untranslated	10%	37%
Spanish	16%	49%
Portuguese	15%	57%
Chinese	14%	29%
Somali	31%	91%
Macedonian	27%	80%

Table 9: Percentage of malicious prompts answered by GPT-5.4 mini, by language, according to Claude Haiku-4.5 as judge.