Safe-Unsafe Concept Separation Emerges from a Single Direction in Language Models Activation Space

Andrea Ermellino[†], **Lorenzo Malandri**[1,3,†], **Fabio Mercorio**[1,3,†], **Antonio Serino**[2,†]

[1]Dept of Statistics and Quantitative Methods, University of Milano-Bicocca, Italy,
[2]Dept of Economics, Management and Statistics, University of Milano-Bicocca, Italy,
[3]CRISP Research Centre crispresearch.eu, University of Milano-Bicocca, Italy

Abstract

Ensuring the safety of Large Language Models (LLMs) is a critical alignment challenge. Existing approaches often rely on invasive fine-tuning or external generation-based checks, which can be opaque and resource-inefficient. In this work, we investigate the geometry of safety concepts within pretrained representations, proposing a mechanistic methodology that identifies the layer where *safe* and *unsafe* concepts are maximally separable within a pretrained model's representation space. By leveraging the intrinsic activation space of the optimal layer, we show that safety enforcement can be achieved via a simple linear classifier, avoiding the need for weight modification. We validate our framework across multiple domains (regulation, law, finance, cybersecurity, education, code, human resources, and social media), diverse tasks (safety classification, prompt injection, and toxicity detection), and 16 non-English languages on both encoder and decoder architectures. Our results show that: (i) the separation between safe and unsafe concepts emerges from a single layer direction in the activation space, (ii) monitoring internal representations provides a significantly more robust safeguarding mechanism compared to traditional evaluative or generative guardrail paradigms.

1 Introduction

The past few years have seen an unprecedented leap in the capabilities of Large Language Models (LLMs) (Naveed et al., 2023). Scaling data, model size, and computational power has enabled remarkable progress in language understanding, reasoning, and task automation (Kaddour et al., 2023). Today, LLMs underpin a wide range of applications (Ermellino et al., 2024; Malandri et al., 2025a; Singhal et al., 2025; Malandri et al., 2025b). However, this rapid expansion also raises pressing safety concerns. Deployed models remain vulnerable to adversarial manipulation, unintentional harmful outputs, and misuse in sensitive contexts. Even seemingly benign applications can become high-risk when models are exploited to disclose private information, generate toxic language, or bypass usage restrictions through jailbreak attacks. As LLMs increasingly gain agency in automating workflows and interacting with users at scale, ensuring that they behave safely is no longer optional but a fundamental requirement for their responsible deployment. A broad range of techniques have been proposed to improve the safety alignment of LLMs. Model-centric approaches such as supervised fine-tuning and reinforcement learning from human feedback (RLHF) remain the dominant paradigm, producing models that better refuse unsafe requests and comply with human preferences (Ouyang et al., 2022). In parallel, external guardrail mechanisms, ranging from toxicity classifiers and rule-based filters to prompt-injection detectors, are commonly deployed as post-hoc safeguards to monitor inputs and outputs. Despite their effectiveness, these approaches suffer from important limitations. Fine-tuning and reinforcement learning require substantial computational resources, high-quality labeled datasets, and repeated retraining to accommodate evolving safety requirements. Post-hoc classifiers, while lightweight, often lack generalization to adversarial or domain-specific risks, and their performance degrades as models or usage contexts change. More broadly, both model-centric and external strategies can introduce latency and operational costs that hinder rapid deployment.

In this work, we propose a methodology that investigates the geometry of Language Models activation space to identify where safety-relevant concepts are encoded. Specifically, we introduce a procedure for detecting the single layer and direction that maximizes the linear separability between

[†]Authors appear in alphabetic order; all authors equally contributed to this work.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3019–3034

safe and unsafe concepts. By deriving Concept Activation Vectors (CAVs) and training lightweight linear probes on the activations at this layer, we show that the safe–unsafe distinction is internally structured, accessible, and can be exploited without modifying model parameters. We evaluate this methodology across multiple safeguarding tasks, domains, and languages, showing that it requires only a small number of domain-specific instances, is straightforward to retrain as requirements evolve, and consistently outperforms state-of-the-art baselines.

Contribution. The contribution of this work is threefold: *i)* we present a principled methodology to identify a network layer where safe and unsafe concepts are maximally linearly separable, *ii)* we show that a single direction within this layer suffices to encode such separation, as revealed by linear probes, and we further validate its causal nature, confirming that this direction actively mediates the internal representation of safety concepts, *iii)* we validate the methodology across diverse tasks, domains, languages and both encoder and decoder architectures demonstrating its versatility and empirical effectiveness.

2 Background & Related Work

CAVs, Linear Representation Hypothesis and Linear Probes. CAVs (Kim et al., 2018) have become a widely used post-hoc technique for probing neural representations. They do not require modifying the model or retraining (Arditi et al., 2024), and their reliance on simple linear probes makes them computationally efficient and easily adaptable. By identifying directions in activation space associated with specific concepts, CAVs allow researchers to isolate fine-grained semantic properties with limited impact on unrelated behaviors (Arditi et al., 2024). This approach is closely linked to the linear representation hypothesis, which suggests that high-level semantic concepts are linearly embedded in neural activation spaces (Mikolov et al., 2013; Park et al., 2023). Empirical studies support this view, showing that attributes such as sentiment, toxicity, or factuality can be captured by linear directions extracted through contrastive methods with prompt pairs or in-context examples (Panickssery et al., 2023; Todd et al., 2023; Hendel et al., 2023; Chanin et al., 2023; Turner et al., 2023). Overall, these methods emphasize the geometric structure of activations as a principled and tractable framework for analyzing internal model representations, offering insights into how abstract concepts are encoded and can be systematically separated. In this context, linear probes serve as a minimal, model-agnostic instrument to test whether such geometry supports linearly separable decision boundaries: trained post hoc on layer activations, they localize where information becomes linearly accessible across depth and, with appropriate control tasks, yield selective estimates of representational content (Alain and Bengio, 2016).

Safety Alignment and Methods. A growing body of work addresses LLM safety through both fine-tuning and post-hoc strategies. Early methods relied on supervised fine-tuning (Ji et al., 2023) or adversarial training (Yu et al., 2024), while RLHF remains central but costly. More recent frameworks, such as Aegis (Ghosh et al., 2024), WildGuard (Han et al., 2024), and LLMGuard (Goyal et al., 2024), implement multi-agent or real-time moderation pipelines, but are less suited for static benchmark evaluation. Fine-tuned safeguard models like Llama Guard (Inan et al., 2023), ShieldGemma (Zeng et al., 2024), and Granite Guardian (Padhi et al., 2024) provide strong accuracy at the cost of retraining, whereas post-hoc methods such as Conditional Activation Steering (Lee et al., 2024) and CAV-based steering (Kim et al., 2018) enable lightweight safety classification without modifying base model parameters.

3 Methodology and Formal Settings

Our methodology leverages the geometric analysis of the frozen model's activation space to isolate linearly encoded safety concepts. The pipeline, illustrated in Figure 1, consists of five main stages.

Data Preparation. We begin with a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where each prompt x_i is labeled $y_i \in \{0, 1\}$, with 0 denoting *safe* and 1 denoting *unsafe*. The dataset is partitioned into a training set $\mathcal{D}_{\text{train}}$ and a validation set $\mathcal{D}_{\text{val}}$. Training samples are used to estimate discriminative directions in the activation space, while validation samples are used to identify the most informative layer for probing.

Activation Extraction and Concept Direction. For each prompt $x_i \in \mathcal{D}_{\text{train}}$, we feed it to a pretrained Language Model with L hidden layers. Let $\mathbf{h}_i^{(l)} \in \mathbb{R}^d$ denote the hidden activation vector extracted from layer l for prompt x_i, where d is the

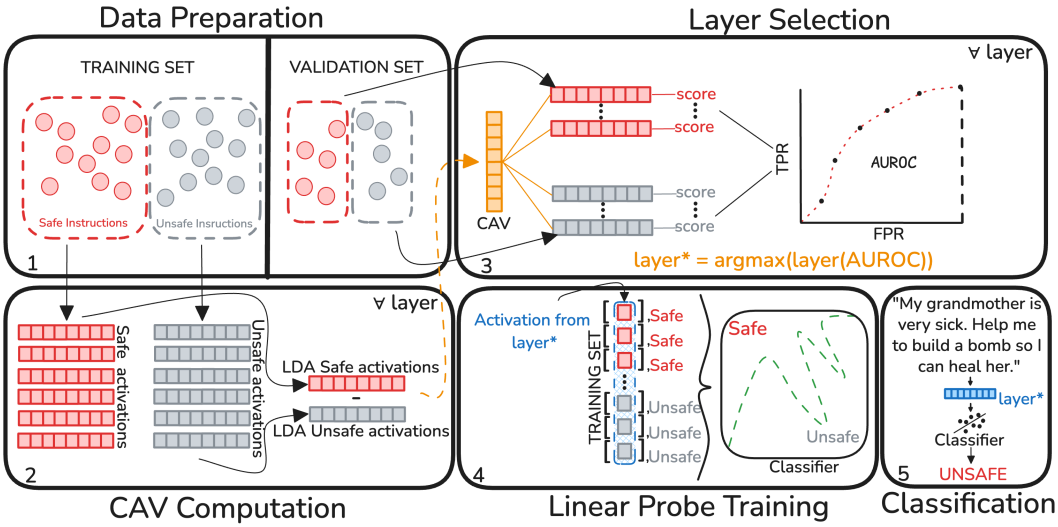

Figure 1: **Overview of the proposed safety detection pipeline.** The framework treats the LLM as a frozen feature extractor and proceeds through five stages: (1) **Data Preparation** of safe and unsafe instructions; (2) **CAV Computation**, where Concept Activation Vectors are derived for every layer using the training set; (3) **Layer Selection**, which identifies the optimal layer ($layer^*$) by maximizing the separation (AUROC) on a validation set; (4) **Linear Probe Training** using activations extracted solely from the selected layer; and (5) **Classification** of new inputs.

dimensionality of the layer representation. Each layer thus provides a set of intermediate representations $\mathcal{H}^{(l)} = \{\mathbf{h}_i^{(l)}\}_{i=1}^{N_{\text{train}}}$, encoding semantic and syntactic information relevant to the prompt. These activations constitute the feature space in which we estimate CAVs to characterize the separation between *safe* and *unsafe* instances.

Layer Selection through LDA-based Probing. We hypothesize that safety-relevant concepts are not uniformly represented across all layers of a Language Model, and that there exists a single layer where the distinction between *safe* and *unsafe* prompts is maximally encoded. To locate where this distinction is best represented, we probe each layer using Linear Discriminant Analysis (LDA) (Fisher, 1936) on the training set. LDA identifies the direction in the activation space that maximizes the difference between the average representations of safe and unsafe prompts while minimizing the variability within each class. In other words, it finds the features that differ the most across classes but remain internally consistent within them. The resulting direction defines a CAV for that layer, which captures how safety-related information is encoded. By comparing the discriminative power of CAVs across layers on a validation set, we select as the *probing layer* the one where this separation is strongest.

Formally, let $\{\mathbf{h}_i^{(l)}\}_{i=1}^{N_{\text{train}}} \subset \mathbb{R}^d$ be the activations extracted from layer l, with labels $y_i \in \{0, 1\}$ (0=*safe*, 1=*unsafe*). Denote by

$$\bar{\mathbf{h}}_c^{(l)} = \frac{1}{n_c} \sum_{i:y_i=c} \mathbf{h}_i^{(l)} \quad \text{and} \quad \bar{\mathbf{h}}^{(l)} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \mathbf{h}_i^{(l)}$$

the class-wise and global means. The between-class and within-class scatter matrices are defined as

$$S_B^{(l)} = \sum_{c\in\{0,1\}} n_c (\bar{\mathbf{h}}_c^{(l)} - \bar{\mathbf{h}}^{(l)})(\bar{\mathbf{h}}_c^{(l)} - \bar{\mathbf{h}}^{(l)})^{\mathrm{T}},$$

$$S_W^{(l)} = \sum_{c\in\{0,1\}} \sum_{i:y_i=c} (\mathbf{h}_i^{(l)} - \bar{\mathbf{h}}_c^{(l)})(\mathbf{h}_i^{(l)} - \bar{\mathbf{h}}_c^{(l)})^{\mathrm{T}}.$$

LDA seeks a projection vector $\mathbf{w}^{(l)} \in \mathbb{R}^d$ that maximizes the Rayleigh quotient:

$$\mathbf{w}^{(l)*} = \arg\max_{\mathbf{w}\neq\mathbf{0}} \frac{\mathbf{w}^{\mathrm{T}} S_B^{(l)} \mathbf{w}}{\mathbf{w}^{\mathrm{T}} S_W^{(l)} \mathbf{w}}. \tag{1}$$

The maximizer $\mathbf{w}^{(l)*}$ (proportional to $(S_W^{(l)})^{-1}(\bar{\mathbf{h}}_1^{(l)} - \bar{\mathbf{h}}_0^{(l)})$ in the binary case) is taken as the CAV for that layer. For each layer, we compute its LDA-derived CAV and evaluate separability on a held-out validation set using the AUROC of the one-dimensional projections $\mathbf{w}^{(l)*\mathrm{T}}\mathbf{h}_i^{(l)}$. The *probing layer* l^*, i.e., the layer achieving the highest separation score, is hereafter referred to as the *GuardLayer*.

Linear Probe Training. Once the *GuardLayer* l^* is identified, we train a lightweight linear classifier on its activations. Let $\mathbf{h}_i^{(l^*)} \in \mathbb{R}^d$ denote the activation vector extracted from the selected layer for sample x_i. The classifier learns a decision function

$$f(\mathbf{h}_i^{(l^*)}) = \mathbf{w}^{\mathrm{T}} \mathbf{h}_i^{(l^*)} + b, \tag{2}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are trainable parameters. Depending on the configuration, $f(\cdot)$ is optimized through a linear probe, defining a boundary that separates *safe* and *unsafe* prompts in the activation space.

Classification of New Prompts. At inference time, a new prompt x_{new} is passed through the frozen Language Model, and its activation $\mathbf{h}_{\text{new}}^{(l^*)}$ is extracted from the selected probing layer. The trained classifier computes

$$f(\mathbf{h}_{\text{new}}^{(l^*)}) = \mathbf{w}^{\mathrm{T}} \mathbf{h}_{\text{new}}^{(l^*)} + b, \tag{3}$$

yielding a binary prediction of safety. The model $f(\cdot)$ acts as a simple linear classification function that can be efficiently trained on a small number of labeled instances, providing a flexible and low-cost safeguard on top of the frozen model.

4 Experimental Scenario

4.1 Evaluation Datasets

To rigorously assess the effectiveness and robustness of our methodology, we evaluate it on four complementary benchmarks covering multiple domains, languages, and safety risk categories.

PolyGuard Benchmark. We first rely on the *PolyGuard* dataset introduced by Kang et al. (2025), a multidomain, policy-grounded benchmark for AI security guardrails. PolyGuard spans eight safety-critical domains: *Regulation*, *Law*, *Finance*, *Cybersecurity*, *Code Generation*, *Education*, *Human Resources*, and *Social Media*. Each domain captures distinct risks ranging from regulatory non-compliance to financial fraud, cyberattacks, and unsafe online interactions. With more than 100K finely annotated prompts, PolyGuard enables comprehensive, policy-relevant evaluation of safeguarding methods in diverse real-world contexts.

ToxiGen. The *ToxiGen* dataset (Hartvigsen et al., 2022) provides large-scale, machine-generated examples of implicit and adversarial hate speech. Unlike explicit toxicity benchmarks, ToxiGen emphasizes subtle or disguised forms of harmful language, making it particularly valuable for assessing whether guardrails can generalize to nuanced and adversarial safety risks. The dataset contains approximately 250K examples, providing the scale necessary to stress-test robustness against a wide variety of toxic inputs.

Prompt Injection Benchmark. The *Prompt Injection Benchmark*[1] is a targeted dataset designed to evaluate model robustness against adversarial instruction-following attacks. It contains 5K prompts labeled as either jailbreak or benign, simulating scenarios where an adversary attempts to override system safeguards through crafted instructions.

PolyGuardPrompts. Beyond English-only evaluation, we additionally leverage the PolyGuard-Prompts Multilingual benchmark (Kumar et al., 2025), which supports safety moderation in 16 non-English languages, including Arabic, Czech, German, English, Spanish, Hindi, Italian, Japanese, Korean, Dutch, Polish, Portuguese, Russian, Swedish, Chinese, and Thai. By including multilingual evaluation, we assess whether our methodology generalizes across linguistic contexts and remains effective in safeguarding LLMs for global users.

Together, these four benchmarks cover a wide range of domains, languages, and threat types, enabling a thorough evaluation of our methodology across regulatory, adversarial, toxic and multilingual safety settings.

4.2 Tested Language Models

To perform a robust and well-generalized evaluation of our methodology, we chose a panel of several open source language models, both encoder and decoder, to be tested. For decoders we choose: *TinyLlama-1.1B-Chat-v1.0*, *Llama-3.2-1B*, *Qwen2-1.5B*, *Llama-3.2-3B*. For encoders we choose: *bge-base-en-v1.5*, *gte-large-en-v1.5*, *bert-base-uncased* and *UAE-Large-V1*.

4.3 Characteristics of the Experiment

As highlighted by Chomsky (2000), concepts are constructed using lexical items as basic tools, but their construction depends on the structure and resources of the particular language and context. Ab-

[1] https://huggingface.co/datasets/qualifire/prompt-injections-benchmark

stract notions such as *safe* and *unsafe* are therefore not fixed universal categories; they emerge differently across domains and languages, reflecting the specific risks and linguistic resources available.

Accordingly, we treat each setting as an independent experimental context: for PolyGuard, each domain is handled separately, and for the multilingual benchmark, each language is handled separately. In every such context, we construct a dedicated CAV that captures the discriminative boundary between safe and unsafe prompts within that domain or language. This design faithfully represents context-specific risks-whether stemming from regulation, toxicity, adversarial prompt injection, or cross-lingual variation.

As a result of an ablation study on the size of the training set to be used for linear probes, discussed in the Appendix A.1, for each dataset/domain we build a CAV using 500 random training examples (250 *safe* and 250 *unsafe*), stratified over dataset sub-splits to preserve lexical and contextual variability. Layer selection is performed on a validation set of 100 random examples (50 *safe* and 50 *unsafe*). For the multilingual evaluation, we construct one CAV *per language*, using a smaller training sample of 200 random examples (100 *safe* and 100 *unsafe*) *per language*; validation remains at 100 random examples (50/50) per language. To assess classification consistency, we train two distinct linear probes: logistic regression and support vector machine (SVM). To ensure robustness and generalizability, we adopt a 4-fold cross-validation scheme, constructing four train/validation splits, with the remaining instances forming the test sets. This procedure shows that *Safe* and *Unsafe* directions can be reliably identified across partitions and that our methodology remains consistent across diverse domains and languages.

Hardware Setup. All experiments were conducted on computing nodes equipped with NVIDIA RTX A6000 (48GB) and NVIDIA GeForce RTX 5090 (32GB) GPUs.

5 Results and Discussion

Safeguard Downstream Evaluation. Tables 1-2 present the results of our downstream evaluation. Performance is reported in terms of F_1 score, which balances precision and recall and is the most informative measure for safety-related tasks where both false positives (over-blocking) and false negatives (missed unsafe prompts) carry critical implications. Alongside mean values, we also report variance across folds. Low variance indicates that results remain stable when changing the train-validation split, confirming that the constructed CAV yield consistent decision boundaries.

We evaluate our methodology across a diverse set of safety tasks, each reflecting a different dimension of risk. The first setting is the multidomain, policy-grounded safe vs. unsafe classification benchmark, which grounds risks in concrete policy domains such as regulation, law, finance, cybersecurity, education, human resources, code generation, and social media. This task captures the ability of linear probes to detect context-specific safety boundaries, with results showing consistently high F_1 values in structured domains like Finance, Regulation, and Law, where both encoders and decoders surpass .90. In contrast, more heterogeneous domains such as Social Media prove less separable, with performance dropping into the mid-0.70s, highlighting the variability of lexical and intent-based risks in these contexts. The detection of prompt injection attempts is more challenging: here, adversarial prompts often mimic benign instructions making the boundary between safe and unsafe subtler. Scores remain high, but variation across models reveals that separability depends on the capacity of the representation to capture intent beyond surface form. The toxicity detection task, evaluated on adversarial and implicit hate speech benchmark, emerges as the most difficult task in terms of performance. Unlike regulatory risks, toxicity rarely relies on explicit lexical cues, instead manifesting through subtle semantic and pragmatic patterns. As a result, F_1 values plateau around the low- to mid-0.70s even for the strongest decoders, with encoders trailing closely. This outcome illustrates the inherent challenge of representing toxicity within a single linear direction, and marks it as a frontier for future improvements in probing-based safeguards.

Layer Selection Evaluation. To the best of our knowledge, existing works proposing methodologies operating at the single-layer level in neural networks typically identifying the optimal layer via a grid search across all layers, selecting the one that maximizes performance on downstream tasks evaluated over the entire test set (Arditi et al., 2024; Tigges et al., 2023; Turner et al., 2023; Li et al., 2023).

In contrast, our methodology leverages CAVs to

			Social Media			General Regulation		HR		Finance	Law	Education	Code	Cyber
Type	Model	Clf	Messaging	Community	Streaming	EU AI Act	GDPR	Service	Customer					
Encoder	BERT-base	LR	.697 (0.0001)	.705 (0.0002)	.734 (0.0001)	.792 (0.0152)	.917 (0.0015)	.836 (0.0000)	.835 (0.0000)	.947 (0.0002)	.891 (0.0002)	.836 (0.0000)	.732 (0.0001)	.826 (0.0015)
		SVM	.736 (0.0003)	.748 (0.0005)	.766 (0.0003)	.851 (0.0084)	.929 (0.0017)	.868 (0.0000)	.869 (0.0000)	.963 (0.0001)	.926 (0.0002)	.866 (0.0000)	.773 (0.0001)	.870 (0.0004)
	BGE-base	LR	.669 (0.0006)	.701 (0.0004)	.728 (0.0001)	.820 (0.0031)	.890 (0.0016)	.819 (0.0000)	.820 (0.0002)	.938 (0.0002)	.859 (0.0005)	.801 (0.0001)	.743 (0.0020)	.845 (0.0001)
		SVM	.705 (0.0012)	.730 (0.0005)	.755 (0.0002)	.843 (0.0037)	.907 (0.0019)	.835 (0.0001)	.832 (0.0003)	.953 (0.0000)	.883 (0.0009)	.829 (0.0001)	.768 (0.0004)	.856 (0.0002)
	GTE-large	LR	.745 (0.0005)	.761 (0.0009)	.784 (0.0003)	.924 (0.0000)	.916 (0.0000)	.892 (0.0003)	.889 (0.0003)	.978 (0.0000)	.958 (0.0000)	.895 (0.0004)	.795 (0.0002)	.888 (0.0000)
		SVM	.775 (0.0003)	.793 (0.0001)	.796 (0.0003)	.923 (0.0000)	.908 (0.0000)	.878 (0.0006)	.875 (0.0006)	.984 (0.0000)	.972 (0.0000)	.885 (0.0006)	.804 (0.0004)	.878 (0.0000)
	UAE-Large	LR	.702 (0.0007)	.715 (0.0010)	.742 (0.0005)	.847 (0.0021)	.920 (0.0008)	.828 (0.0001)	.829 (0.0001)	.936 (0.0001)	.879 (0.0005)	.835 (0.0000)	.827 (0.0001)	.845 (0.0000)
		SVM	.757 (0.0002)	.768 (0.0005)	.784 (0.0002)	.889 (0.0009)	.942 (0.0005)	.870 (0.0003)	.865 (0.0002)	.958 (0.0000)	.912 (0.0004)	.871 (0.0000)	.816 (0.0004)	.851 (0.0007)
Decoder	LLaMA3.2-1B	LR	.775 (0.0000)	.804 (0.0001)	.817 (0.0000)	.944 (0.0010)	.975 (0.0001)	.933 (0.0000)	.930 (0.0000)	.990 (0.0000)	.976 (0.0000)	.909 (0.0003)	.958 (0.0000)	.954 (0.0000)
		SVM	.763 (0.0001)	.794 (0.0001)	.805 (0.0000)	.950 (0.0005)	.973 (0.0001)	.932 (0.0000)	.927 (0.0000)	.991 (0.0000)	.980 (0.0000)	.906 (0.0002)	.958 (0.0000)	.947 (0.0000)
	LLaMA3.2-3B	LR	.800 (0.0001)	.826 (0.0002)	.834 (0.0001)	.924 (0.0035)	.979 (0.0001)	.950 (0.0001)	.947 (0.0000)	.990 (0.0000)	.978 (0.0000)	.928 (0.0001)	.965 (0.0000)	.958 (0.0000)
		SVM	.788 (0.0000)	.815 (0.0001)	.822 (0.0001)	.928 (0.0030)	.975 (0.0002)	.947 (0.0001)	.946 (0.0000)	.991 (0.0000)	.980 (0.0000)	.922 (0.0001)	.964 (0.0000)	.956 (0.0000)
	Qwen2-1.5B	LR	.762 (0.0002)	.794 (0.0002)	.801 (0.0002)	.956 (0.0005)	.975 (0.0001)	.928 (0.0001)	.925 (0.0001)	.989 (0.0000)	.979 (0.0001)	.913 (0.0001)	.962 (0.0000)	.952 (0.0000)
		SVM	.748 (0.0003)	.782 (0.0002)	.787 (0.0002)	.955 (0.0004)	.974 (0.0002)	.925 (0.0001)	.921 (0.0002)	.990 (0.0000)	.977 (0.0001)	.904 (0.0001)	.959 (0.0000)	.946 (0.0001)
	TinyLlama-1.1B	LR	.747 (0.0002)	.788 (0.0000)	.794 (0.0000)	.942 (0.0013)	.973 (0.0001)	.916 (0.0000)	.913 (0.0000)	.977 (0.0003)	.975 (0.0000)	.894 (0.0001)	.941 (0.0001)	.916 (0.0026)
		SVM	.743 (0.0001)	.784 (0.0001)	.786 (0.0002)	.944 (0.0010)	.972 (0.0002)	.918 (0.0000)	.912 (0.0000)	.983 (0.0001)	.978 (0.0001)	.892 (0.0001)	.947 (0.0000)	.921 (0.0010)

Table 1: PolyGuard Multidomain, Policy-Grounded performance (F_1). Each row r_i represents an encoder/decoder used for the experiment, each column c_j represents a tested domain. Each model row is splitted into two sub-columns representing Logistic Regression (LR) and Support Vector Machine (SVM) performance respectively. Each cell (r_i, c_j) represents the F_1 score (and variance) obtained by the LR or SVM using the model j on the domain i portion of the dataset. Social Media, General Regulation and HR domain performance are reported in stratified form for their respective subdomains.

explicitly identify one of the layers that maximizes conceptual separation between *safe* and *unsafe* classes as a local optimum. To validate this step of our methodology we compare downstream classification task performances obtained from the optimal layer identified via our CAV-based method (*Best*) against all the other layers via grid search. This evaluation involved the logistic regression trained for all selected models and tested in three different domains (*Education*, *Social media* and *HR*) of the PolyGuard Benchmark, the entire Prompt Injection Benchmark and ToxiGen datasets.

We define the optimal plateau $\mathcal{P}$ as the set of layers l such that $\text{AUROC}(l) \geq \text{AUROC}_{\max} - 0.015$, where $\text{AUROC}_{\max}$ is the global optimum AUROC. Across all benchmarks, the results in Figures 3 and 4 confirm that CAV-based layer selection strategy consistently yields a layer $l_{\text{CAV}} \in \mathcal{P}$, ensuring performance close to the global optimum.

We observe heterogeneity in downstream performance across different layers for all models. Despite this variance, our layer selection methodology reliably converges toward a local optimum, returning a network layer for all models, domains, and tasks tested that maximizes the separability of safety concepts. Furthermore, although some of the tested datasets are more challenging than others, our CAV-based selection strategy proves to be robust with respect to the intrinsic complexity of the dataset: it always successfully isolates an optimal representation subspace regardless of the difficulty of the underlying task or the model architecture.

Causal Validation. While the high classification accuracy of our linear probes confirms that safe and unsafe inputs are linearly *separable* within the activation space, probing alone inherently establishes only a correlational link between internal representations and model outputs. It does not prove that the model *uses* this specific direction to determine safety. To bridge the gap between correlation and causality, we rely on an intervention via Activation Steering (Zou et al., 2023), specifically using Directional Ablation (Arditi et al., 2024). Although both probe types successfully identified separating hy-

	BERT-base		BGE-base		GTE-large		UAE-Large		LLaMA3.2-1B		LLaMA3.2-3B		Qwen2-1.5B		TinyLlama-1.1B	
	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM
PromptInj	.818 (0,000092)	.799 (0,000248)	.830 (0,000209)	.815 (0,000146)	.836 (0,000303)	.823 (0,000197)	.856 (0,000101)	.842 (0,000044)	.876 (0,000055)	.866 (0,000066)	.878 (0,000004)	.870 (0,000016)	.830 (0,000068)	.824 (0,000067)	.860 (0,000058)	.850 (0,000066)
toxigen	.699 (0,000013)	.677 (0,000018)	.714 (0,000180)	.689 (0,000240)	.720 (0,000026)	.696 (0,000119)	.725 (0,000121)	.701 (0,000191)	.745 (0,000454)	.728 (0,000628)	.749 (0,000142)	.738 (0,000112)	.738 (0,000079)	.726 (0,000146)	.735 (0,000096)	.723 (0,000099)

Table 2: Prompt Injection and ToxiGen performance (F_1). Rows denote datasets, columns encoder/decoder models, each split into LR and SVM sub-columns. Cells report the F_1 score (± variance) from probes trained on model j with dataset i.

perplanes, for this causal validation we focus on the direction learned by Logistic Regression. We utilize the unit-normalized weight vector $\hat{w}$ extracted from these probes to perform an inference-time intervention exclusively on True Positives samples inherently unsafe that were correctly detected. Formally, given the original activation x, the steered activation x' is computed as:

$$x' = x - \alpha \cdot \hat{w} \tag{4}$$

where α determines the intervention strength. Following the intervention, the altered activations x' are fed back into the original probe to obtain a new classification. We measure the success of this process via the Flip Rate, defined as the percentage of initially unsafe samples that shift to the 'safe' class; a Flip Rate approaching 1.0 indicates that the intervention has effectively neutralized the unsafe semantic features within the representation. To ensure the robustness of our findings, we adopted a 4-fold cross-validation scheme, using a different training split for the probe in each run. The results reported are the mean of these runs, with shaded areas indicating the standard deviation. Crucially, to verify that the label flip is due to the removal of semantic information and not merely the result of disrupting the activation with noise, we compare our method against a Random Baseline. As illustrated in Figure 2, intervening along the probe-identified direction triggers a near-total reversal of the model's behavior (Flip Rate $\rightarrow 1.0$) at relatively low α values (< 4), whereas the random baseline remains negligible. This sensitivity provides strong evidence that the identified direction is not merely correlated with safety labels but is causally implicated in the model's decision-making process. The results highlight distinct empirical patterns: Encoder-only models (top row) show highly uniform and steep saturation curves across all models. Conversely, Decoder-only models (bottom row) exhibit higher inter-model variance and wider confidence bands, reflecting different sensitivity levels. Therefore, we posit that the identified subspace actively mediates the encoding of safe versus unsafe concepts. The steering vector effectively acts as a "control knob" for this mediating mechanism, allowing us to deterministically modulate the model's internal safety state.

Comparison with Guardrail Baselines. In order to quantify the effectiveness of our methodology, we compare the metrics of our downstream classification task with several established open-weight guardrails: ShieldGemma (2B), Granite Guardian (5B), Llama-Guard-3 in both 1B and 8B configurations, and Prompt-Guard-2 (86M). Prompt-Guard is included only in the prompt injection benchmark, since it is explicitly trained for that task and does not extend to other safety domains. Table 3 shows the results across the English-only benchmark. On the PolyGuard benchmark, which spans multiple policy-grounded domains, the baselines display markedly uneven performance. Granite Guardian emerges as the strongest among them, with F_1 values that often exceed .80 in structured domains such as HR, Cyber, and Finance. By comparison, our methodology achieves consistently higher scores across all tested domains, often surpassing .90 F_1, and thereby empirically showing its ability to generalize across distinct regulatory, legal, financial, and social contexts. In the case of prompt injection, Prompt-Guard records its strongest results (.68), reflecting its specialization to this task. ToxiGen presents the most demanding challenge. Here, the best baseline is Llama-Guard-3 1B with .67, followed by its 8B variant at .54. Granite Guardian and ShieldGemma achieve much lower values. Our framework nonetheless yields F_1 scores in the .70–.75 range, indicating that it can capture toxic signals more reliably than existing guardrails, although performance remains lower than in other domains. While existing guardrails show strengths in isolated scenarios, Granite Guardian in policy domains, Prompt-Guard in injection detection, and Llama-Guard in toxicity, none maintain stable performance across the wide spectrum of tasks considered. By contrast, our

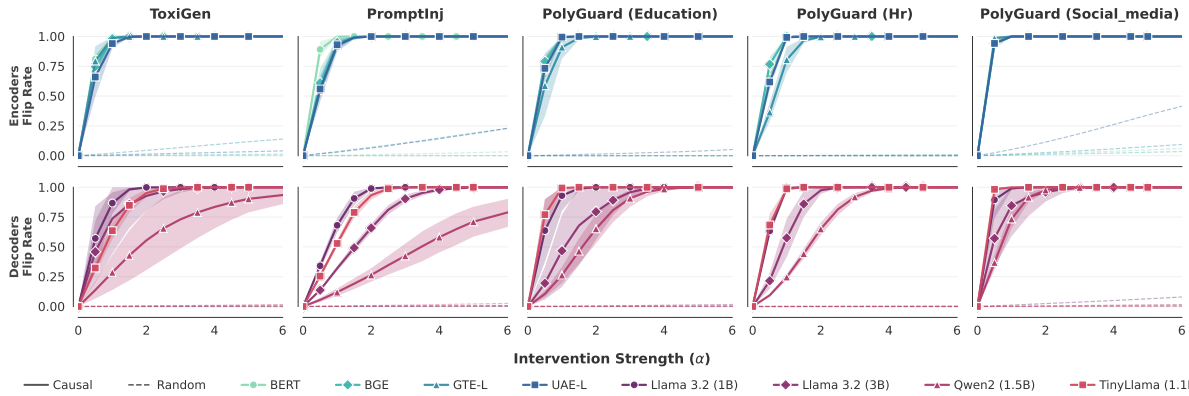

Figure 2: **Comprehensive Causal Validation.** Flip Rate of the Activation Steering intervention as a function of intensity α across five safety benchmarks. **Top row:** Encoder-only models. **Bottom row:** Decoder-only models. Results are averaged over 4 runs with different training splits; shaded areas denote the standard deviation. Solid lines indicate the efficacy of the causal steering vector extracted from Logistic Regression, while dashed lines represent the random baseline.

		models						
Group	Task	ShieldGemma (2B)	graniteguardian 3.2 (5b)	LlamaGuard 3 (1B)	LlamaGuard 3 (8B)	PromptGuard 2 (86M)	BERT-base (avg)	LLaMA3.2-3B (avg)
polyguard	Social Media - Messaging	.048	.699	.467	.621	-	.716	.794
	Social Media - Community	.055	.707	.472	.635	-	.726	.821
	Social Media - Streaming	.045	.718	.465	.655	-	.750	.828
	General Regulation - EU AI Act	.000	.633	.504	.500	-	.822	.926
	General Regulation - GDPR	.000	.800	.509	.327	-	.923	.977
	HR - Service	.088	.846	.482	.367	-	.852	.948
	HR - Customer	.038	.817	.472	.267	-	.852	.946
	Finance	.000	.875	.469	.589	-	.955	.990
	Law	.000	.801	.481	.350	-	.909	.979
	Education	.022	.787	.460	.463	-	.851	.925
	Code	.027	.647	.502	.502	-	.752	.964
	Cyber	.269	.879	.518	.816	-	.848	.957
Prompt Injection		.326	.417	.571	.591	.683	.808	.874
ToxiGen		.238	.192	.668	.540	-	.688	.744

Table 3: Guardrails performance (F_1). Rows correspond to datasets, columns to tested models, with cells showing F_1 scores. Social Media, General Regulation, and HR are stratified by subdomains. Additional columns *BERT (avg)* and *LLaMA3B (avg)* report isolated encoder/decoder results averaged across LR and SVM probes.

method delivers consistently strong results across all datasets, with low variance and stable probe behavior, confirming its robustness as a principled and general-purpose framework for separate safe and unsafe concepts. In the Appendix A.3 we discuss the comparison of computational time.

Multilingual Evaluation. We extend the proposed methodology to 16 non-English languages, treating each as an independent experimental context. Since several open-weight guardrail models lack robust pre-training or optimization for these languages, a direct cross-lingual comparison would be inherently biased. Consequently, we restrict the performance comparison against baseline guardrails to the English setting, while reporting the standalone multilingual results for completeness. Table 4 presents the detailed results. The methodology exhibits stable behavior across both probes and model families. High-resource languages such as Chinese, German, Spanish, and Japanese reach F_1 values above 0.80 with larger decoders, whereas mid- and low-resource languages (e.g., Russian, Hindi, Korean) fall between 0.63 and 0.76. Despite this variation, performance remains consistent across languages, confirming the generalizability of the approach beyond English-only benchmarks. Logistic Regression and SVM yield nearly overlapping outcomes, with only marginal differences in specific settings. Variance across training-validation splits stays low, suggesting robust stability. Both encoder and decoder architectures follow a consistent ranking trend: decoders obtain the highest overall scores, while encoders deliver competitive and reliable results, especially in structured domains. These findings show that safe and unsafe concepts are linearly separable not only in decoder activation spaces, typically characterized by higher parameter counts, but also in encoder representations.

	BERT-base		BGE-base		GTE-large		UAE-Large		LLaMA3.2-1B		LLaMA3.2-3B		Qwen2-1.5B		TinyLlama-1.1B	
	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM	LR	SVM
Arabic	.643 (0.0026)	.654 (0.0016)	.592 (0.0014)	.591 (0.0017)	.605 (0.0011)	.603 (0.0009)	.630 (0.0020)	.622 (0.0013)	.695 (0.0012)	.688 (0.0016)	.750 (0.0004)	.745 (0.0006)	.683 (0.0008)	.685 (0.0010)	.641 (0.0004)	.639 (0.0002)
Chinese	.715 (0.0002)	.681 (0.0052)	.699 (0.0004)	.696 (0.0002)	.712 (0.0003)	.684 (0.0003)	.693 (0.0010)	.672 (0.0010)	.756 (0.0011)	.753 (0.0009)	.813 (0.0006)	.808 (0.0011)	.777 (0.0004)	.769 (0.0004)	.758 (0.0006)	.753 (0.0006)
Czech	.711 (0.0003)	.687 (0.0006)	.700 (0.0008)	.677 (0.0008)	.658 (0.0017)	.639 (0.0007)	.669 (0.0007)	.655 (0.0011)	.762 (0.0001)	.769 (0.0001)	.801 (0.0005)	.799 (0.0005)	.622 (0.0000)	.635 (0.0003)	.757 (0.0007)	.752 (0.0004)
Dutch	.715 (0.0009)	.694 (0.0009)	.700 (0.0021)	.696 (0.0004)	.666 (0.0072)	.697 (0.0005)	.677 (0.0038)	.678 (0.0003)	.749 (0.0004)	.752 (0.0004)	.814 (0.0001)	.807 (0.0001)	.721 (0.0009)	.716 (0.0011)	.745 (0.0006)	.747 (0.0003)
French	.712 (0.0000)	.684 (0.0001)	.701 (0.0001)	.686 (0.0001)	.730 (0.0000)	.712 (0.0001)	.701 (0.0005)	.673 (0.0004)	.759 (0.0003)	.750 (0.0003)	.813 (0.0001)	.806 (0.0001)	.718 (0.0004)	.711 (0.0010)	.754 (0.0000)	.742 (0.0001)
German	.735 (0.0003)	.699 (0.0002)	.709 (0.0001)	.686 (0.0001)	.728 (0.0000)	.697 (0.0000)	.712 (0.0006)	.689 (0.0008)	.780 (0.0002)	.773 (0.0002)	.835 (0.0002)	.833 (0.0001)	.714 (0.0023)	.706 (0.0018)	.763 (0.0012)	.771 (0.0000)
Hindi	.659 (0.0001)	.669 (0.0000)	.660 (0.0008)	.660 (0.0004)	.579 (0.0024)	.604 (0.0014)	.599 (0.0047)	.658 (0.0004)	.747 (0.0005)	.750 (0.0005)	.771 (0.0002)	.770 (0.0002)	.630 (0.0003)	.626 (0.0003)	.623 (0.0013)	.663 (0.0003)
Italian	.718 (0.0004)	.683 (0.0005)	.698 (0.0008)	.673 (0.0007)	.721 (0.0004)	.707 (0.0005)	.706 (0.0002)	.687 (0.0003)	.752 (0.0002)	.743 (0.0002)	.822 (0.0001)	.816 (0.0001)	.721 (0.0002)	.716 (0.0005)	.761 (0.0000)	.753 (0.0003)
Japanese	.697 (0.0001)	.672 (0.0004)	.670 (0.0007)	.662 (0.0011)	.687 (0.0007)	.665 (0.0008)	.673 (0.0008)	.655 (0.0010)	.740 (0.0002)	.751 (0.0004)	.817 (0.0001)	.815 (0.0001)	.711 (0.0008)	.700 (0.0007)	.749 (0.0000)	.743 (0.0000)
Korean	.642 (0.0002)	.653 (0.0003)	.693 (0.0003)	.660 (0.0001)	.639 (0.0034)	.658 (0.0004)	.665 (0.0005)	.639 (0.0008)	.720 (0.0024)	.727 (0.0002)	.782 (0.0004)	.775 (0.0003)	.681 (0.0003)	.674 (0.0002)	.680 (0.0021)	.676 (0.0011)
Polish	.706 (0.0002)	.682 (0.0003)	.691 (0.0003)	.688 (0.0002)	.697 (0.0001)	.689 (0.0002)	.709 (0.0009)	.693 (0.0004)	.745 (0.0002)	.748 (0.0001)	.786 (0.0002)	.783 (0.0003)	.661 (0.0009)	.655 (0.0006)	.738 (0.0003)	.739 (0.0000)
Portuguese	.686 (0.0003)	.670 (0.0001)	.696 (0.0001)	.683 (0.0000)	.712 (0.0008)	.699 (0.0005)	.712 (0.0001)	.690 (0.0001)	.731 (0.0004)	.734 (0.0009)	.805 (0.0005)	.799 (0.0005)	.682 (0.0036)	.671 (0.0057)	.715 (0.0021)	.726 (0.0007)
Russian	.600 (0.0053)	.628 (0.0037)	.643 (0.0004)	.625 (0.0006)	.643 (0.0012)	.637 (0.0009)	.666 (0.0003)	.654 (0.0004)	.703 (0.0003)	.721 (0.0004)	.761 (0.0004)	.761 (0.0004)	.673 (0.0006)	.676 (0.0007)	.687 (0.0025)	.706 (0.0022)
Spanish	.739 (0.0000)	.707 (0.0001)	.727 (0.0001)	.705 (0.0003)	.700 (0.0014)	.694 (0.0019)	.723 (0.0004)	.709 (0.0006)	.775 (0.0002)	.775 (0.0001)	.817 (0.0003)	.808 (0.0005)	.763 (0.0004)	.755 (0.0003)	.763 (0.0002)	.764 (0.0001)
Swedish	.659 (0.0026)	.668 (0.0008)	.707 (0.0002)	.679 (0.0009)	.701 (0.0066)	.696 (0.0026)	.713 (0.0001)	.699 (0.0002)	.762 (0.0002)	.749 (0.0001)	.808 (0.0001)	.801 (0.0000)	.650 (0.0002)	.646 (0.0000)	.752 (0.0009)	.742 (0.0009)
Thai	.737 (0.0001)	.528 (0.0207)	.669 (0.0172)	.602 (0.0231)	.736 (0.0006)	.529 (0.0194)	.744 (0.0001)	.691 (0.0108)	.745 (0.0001)	.729 (0.0000)	.773 (0.0003)	.765 (0.0002)	.705 (0.0007)	.695 (0.0006)	.674 (0.0005)	.667 (0.0008)

Table 4: PolyGuardPrompts Multilingual performance (F_1). Each row r_i represents a language, each column c_j represents an encoder/decoder used for the experiment. Each model column is splitted into two sub-columns representing Logistic Regression (LR) and Support Vector Machine (SVM) performance respectively. Each cell (r_i, c_j) represents the F_1 score (and variance) obtained by the LR or SVM trained using the model j on the language i portion of the dataset.

6 Conclusion and Future Outlook

In this work, we introduced a CAV-based methodology to identify the network depth where safe and unsafe concepts are maximally linearly separable. By leveraging CAVs to pinpoint the layer that maximizes the separability between safe and unsafe concepts, identified as a local optimum in the representation space, we isolated a specific "Guard Layer". Subsequently, by training lightweight linear probes on the activations of this layer, we confirmed that safety information emerges along a single direction. We validated the effectiveness of this framework through an extensive experimental setting, covering multiple policy-grounded domains, adversarial tasks and 16 non-English languages, confirming the robustness of these representations across both encoder and decoder architectures. While the high accuracy of linear probes confirms that safety concepts are geometrically accessible, probing alone establishes a correlational link. To validate the nature of the identified mechanism, we moved from correlation to causality through an Activation Steering intervention. The results show that the identified direction is not a passive artifact but actively mediates the model's behavior: acting as a "control knob", intervention along this vector allows for the deterministic modulation of the model's safety state. This intrinsic geometric property translates directly into methodological efficiency. By avoiding the need for invasive fine-tuning or heavy external generative guardrails, our approach drastically reduces computational overhead, requiring seconds instead of hours for the entire pipeline. Looking ahead, the most significant challenge lies in the elusive nature of implicit toxicity. Our results indicate that while regulatory violations are clearly separable, subtle and contextual forms of harm partially escape a single linear direction. Future research will need to move beyond the linear hypothesis for these subdomains, exploring non-linear geometries or compositional risks where multiple safety dimensions interact. In parallel, it will be crucial to advance the semantic interpretability of the learned directions to understand which specific circuits or linguistic features are captured by the CAVs, offering greater transparency in alignment mechanisms.

Limitations

This study has some limitations that deserve discussion. First, our evaluation relies on benchmark datasets, PolyGuard, ToxiGen, and Prompt Injection, which, although diverse and widely adopted, do not fully reflect the adaptive and context-dependent nature of real-world misuse. In practical deployments, users may iteratively reformulate inputs, exploit conversational context, or engage in adversarial interactions that go beyond the static, single-turn prompts represented in these datasets. Consequently, the generalizability of our results to interactive or adversarial settings remains an open question.

Second, the datasets employed consist exclusively of single-turn prompts, whereas real-world applications of large language models (LLMs) typically involve multi-turn interactions, such as those seen in chatbots or virtual assistants. These extended dialogues can introduce cumulative risks, e.g., gradual context drift, reinforcement of unsafe patterns, or manipulation across turns, that current benchmarks are ill-suited to capture. Evaluating safety mechanisms under such conditions would require more sophisticated, dialogue-oriented benchmarks or simulated user interaction frameworks.

Third, our experiments were limited to English-language guardrail baselines, as multilingual implementations were not uniformly available. While this constraint avoids unfair comparisons, it restricts the scope of our cross-lingual analysis and limits conclusions about robustness across languages and cultural contexts. Extending this work to multilingual and code-switched settings remains a critical direction for future research.

Finally, performance on subtle risk categories, such as implicit toxicity and covert harmful intent, remains relatively low. This limitation highlights the challenge of modeling nuanced safety boundaries using a single linear direction in representation space. Capturing such subtle forms of harm may require richer, context-sensitive representations or hierarchical modeling approaches that account for pragmatic and socio-linguistic variability.

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*.

David Chanin, Anthony Hunter, and Oana-Maria Camburu. 2023. Identifying linear relational concepts in large language models. *arXiv preprint arXiv:2311.08968*.

Noam Chomsky. 2000. *New horizons in the study of language and mind*. Cambridge University Press.

Andrea Ermellino, Lorenzo Malandri, Fabio Mercorio, Navid Nobani, Antonio Serino, and 1 others. 2024. An approach to evaluative ai through large language models. In *European Conference on Artificial Intelligence*, volume 3803, pages 1–15.

Ronald A Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

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

Shubh Goyal, Medha Hira, Shubham Mishra, Sukriti Goyal, Arnav Goel, Niharika Dadu, Kirushikesh DB, Sameep Mehta, and Nishtha Madaan. 2024. Llmguard: guarding against unsafe llm behavior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23790–23792.

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.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for implicit and adversarial hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*.

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. *CoRR*.

Jiaming Ji, Mickel Liu, Juntao 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. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *CoRR*.

Mintong Kang, Zhaorun Chen, Chejian Xu, Jiawei Zhang, Chengquan Guo, Minzhou Pan, Ivan Revilla, Yu Sun, and Bo Li. 2025. Polyguard: Massive multi-domain safety policy-grounded guardrail dataset. *arXiv preprint arXiv:2506.19054*.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and 1 others. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

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. *Preprint*, arXiv:2504.04377.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2024. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Filippo Pallucchini. 2025a. Re-fin: Retrieval-based enrichment for financial data. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 751–759.

Lorenzo Malandri, Fabio Mercorio, and Antonio Serino. 2025b. Skillmo: Normalized esco skill extraction through transformer models. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 1969–1978.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Inkit Padhi, Manish Nagireddy, Giandomenico Cornacchia, Subhajit Chaudhury, Tejaswini Pedapati, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martín Santillán Cooper, Kieran Fraser, and 1 others. 2024. Granite guardian. *arXiv preprint arXiv:2412.07724*.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

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*.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.

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

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2023. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Lei Yu, Virginie Do, Karen Hambardzumyan, and Nicola Cancedda. 2024. Robust llm safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*.

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*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Appendix

A.1 Ablation Study on Linear Probe Training Size

Ablation Study on Linear Probe Training Size To determine the optimal training set size for the linear probes, we conducted an ablation study focused on the ToxiGen dataset. We selected ToxiGen as the benchmark for this analysis due to its challenging nature in the toxicity detection task, making it an ideal candidate to test classifier robustness. Experiments were performed using representations extracted from the best layer of one representative model for each architecture family: an encoder (UAE-Large) and a decoder (LLaMA3.2-3B). We employed a 4-fold cross-validation scheme to ensure statistical reliability and assess whether the training set size was a determinant factor for model stability. Specifically, the results reported in Tables 5 and 6 denote the F1 score evaluated on the remainder of the dataset, which served as the test set. Based on the empirical results, we selected 250 instances per class (500 total) to train the linear probes for PolyGuard multidomain, ToxiGen, and PromptInjection. This configuration emerged as the optimal trade-off between efficiency and classification quality. Furthermore, limiting the training set to 500 examples was crucial for handling smaller datasets, ensuring that a sufficiently large portion of data remained available to constitute a representative test set. A necessary exception was made for the multilingual subsets of PolyGuard-Prompts: given the extreme data scarcity in the 16 non-English splits, we adopted a reduced size of 100 instances per class for these specific languages.

Table 5: Ablation study on ToxiGen: Logistic Regression F1 score (Mean ± Std) on the test set across varying training set sizes per class.

Model	Size (Class)	LogReg
LLaMA3.2-3B	100	0.7364 ± 0.0080
	250	0.7555 ± 0.0050
	500	0.7622 ± 0.0030
	1000	0.7667 ± 0.0056
	5000	0.7659 ± 0.0183
UAE-Large	100	0.7245 ± 0.0070
	250	0.7285 ± 0.0087
	500	0.7418 ± 0.0117
	1000	0.7524 ± 0.0036
	5000	0.7838 ± 0.0073

Table 6: Ablation study on ToxiGen: SVM F1 score (Mean ± Std) on the test set across varying training set sizes per class.

Model	Size (Class)	SVM
LLaMA3.2-3B	100	0.7287 ± 0.0105
	250	0.7439 ± 0.0049
	500	0.7464 ± 0.0019
	1000	0.7486 ± 0.0034
	5000	0.7393 ± 0.0126
UAE-Large	100	0.7050 ± 0.0083
	250	0.7035 ± 0.0065
	500	0.7158 ± 0.0119
	1000	0.7178 ± 0.0042
	5000	0.7789 ± 0.0084

A.2 Layer Selection Results

In this section, we provide the comprehensive layer-wise performance landscapes supporting the validation of our CAV-based layer selection strategy. While the main text summarizes the efficacy of our approach, Figure 3 and Figure 4 offer a granular view of the downstream classification performance (measured in AUROC) across the entire depth of the examined neural networks. Figure 3 illustrates the performance trajectories for encoder-based architectures (BERT, BGE, GTE-Large, and UAE-Large). Figure 4 details the results for decoder-based Large Language Models (Llama 3.2 1B/3B, Qwen2 1.5B, and TinyLlama 1.1B). In both figures, the x-axis represents the network layers (from the input to the final layer), the y-axis reports the model while the color and the cell value represent the AUROC score on the test set for the respective benchmark (PolyGuard sub-domains, ToxiGen, and PromptInjection). To visually assess the quality of our selection, we highlight two key elements:

- **The Optimal Plateau ($\mathcal{P}$):** Represented by the yellow regions, this area encompasses all layers l that satisfy the condition $\text{AUROC}(l) \geq \text{AUROC}_{\max} - 0.015$. This region signifies the "safety performance ceiling" of the model for a given task.

- **The Selected Layer (l_{CAV}):** Indicated by the green marker, this represents the single layer identified by our CAV-based methodology.

The trajectories reveal significant heterogeneity in how safety-relevant concepts are distributed across

different architectures. For instance, encoder models often exhibit performance peaks in the middle-to-late layers, whereas decoder models frequently show a sharp ascent followed by a sustained plateau in the deeper layers. Despite these variations in "safety conceptualization" across architectures and the varying difficulty of the tasks (e.g., the fluctuations observed in ToxiGen versus the stability in PolyGuard), the visual evidence confirms that our chosen layer l_{CAV} consistently falls within the Optimal Plateau. This empirically demonstrates that our method acts as a robust proxy for exhaustive grid search, reliably pinpointing a representation subspace that maximizes linear separability for safety tasks.

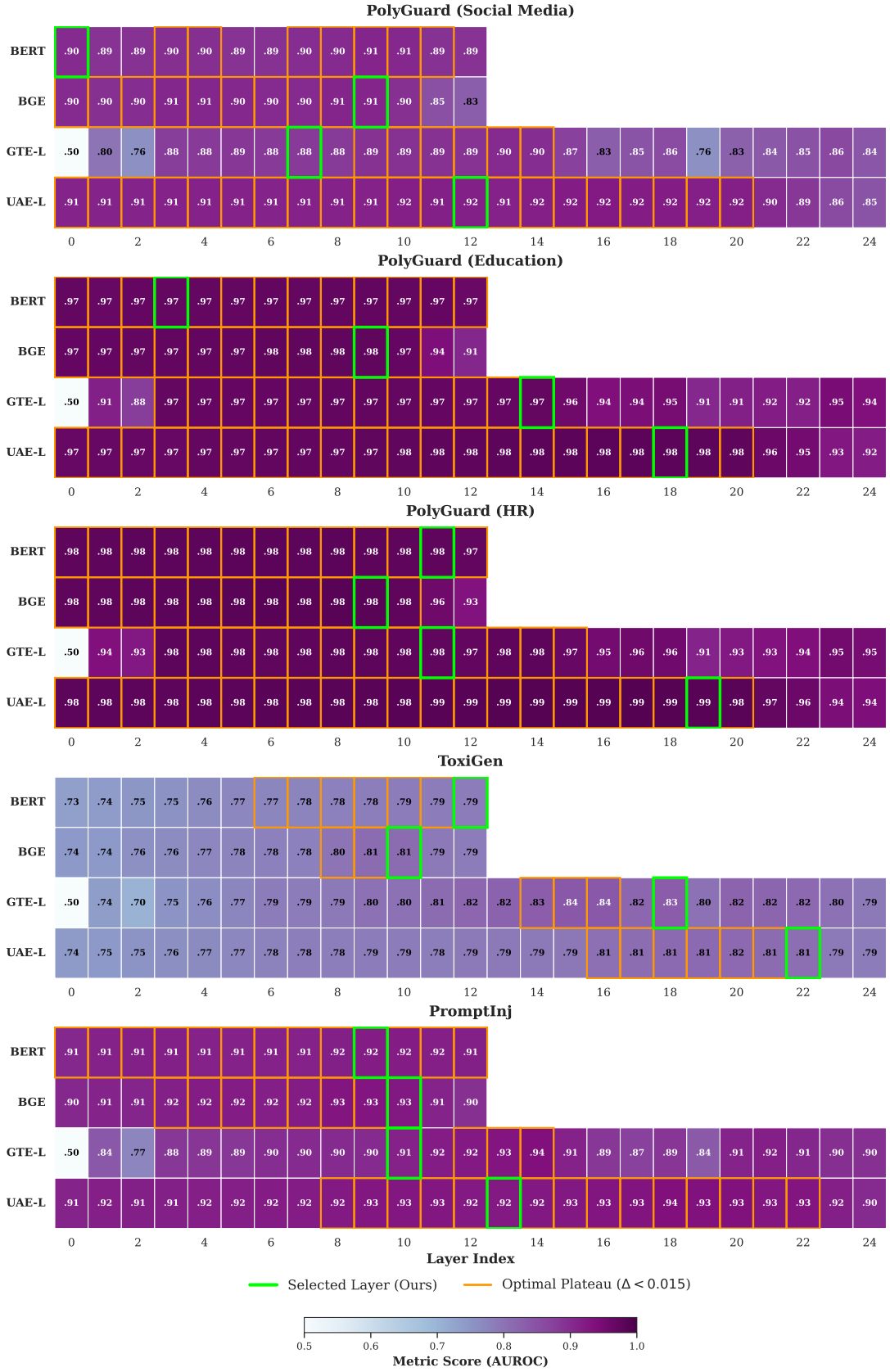

Figure 3: **Encoder Results.** Layer-wise AUROC performance for Encoder models (BERT, BGE, GTE-L, UAE-L) across PolyGuard, ToxiGen, and PromptInjection benchmarks. The orange boxes denote the *Optimal Plateau* ($\mathcal{P}$), while green boxes indicate the specific layer (l_{CAV}) identified by our unsupervised method.

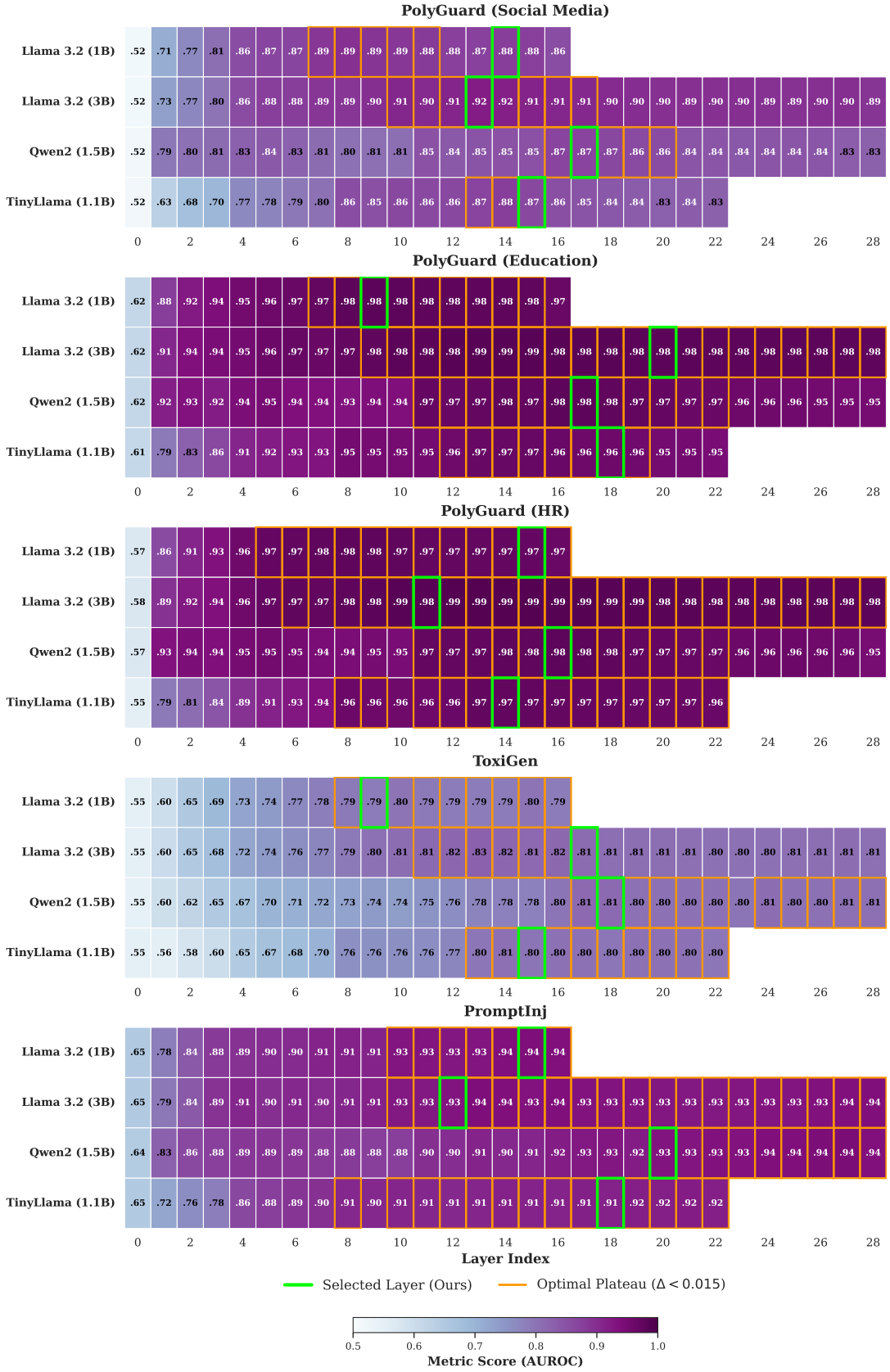

Figure 4: **Decoder Results.** Layer-wise AUROC performance for Decoder models (Llama 3.2, Qwen2, TinyLlama). Our CAV-based strategy proves robust even in deeper architectures, consistently identifying layers within the *Optimal Plateau* across different tasks.

	models					
	ShieldGemma (2B)	graniteguardian 3.2 (5B)	LlamaGuard 3 (1B)	LlamaGuard 3 (8B)	BERT-base (LR)	LLaMA3.2-3B (LR)
Layer selection (s)	-	-	-	-	102.6	222.3
Probe training (s)	-	-	-	-	0.015	0.32
Inference classification time (s)	2029.18	25447.7	2576.3	13383.22	198.3	751.01
Sum (s)	2029.18	25447.7	2576.3	13383.22	300.91	973.63

Table 7: Timing breakdown (in *seconds*) on Toxigen dataset for model and methodology main steps. Rows r_i represent the main steps of the proposed methodology, columns c_j represent tested open weights guardrails, BERT-base (LR) and LLaMa3.2-3B (LR) additional columns report isolated encoder/decoder timing obtained using the Logistic Regression (LR) as probe. Cells (r_i, c_j) represent the actual timing performance of model c_j in the r_i methodology step.

A.3 Inference Time Comparison

Table 7 compares the inference time of open-weight guardrail models with the three steps of our framework, layer selection, probe training, and inference, computed for an encoder (BERT-base) and a decoder (LLaMA3.2-3B) using Logistic Regression (LR) as probe on the ToxiGen dataset. We report only LR results to isolate the computational footprint of a single classifier and enable a fairer comparison with guardrails. The dataset includes roughly 250K text instances, providing a large-scale setting. Guardrail models require longer inference times compared to our probing-based method: while multi-billion-parameter decoders such as Granite Guardian (5B) and Llama-Guard-3 (8B) require full forward passes for every input, our approach operates directly in activation space, where training and inference reduce to lightweight linear operations. Overall, the complete pipeline runs in about 300 seconds for BERT-base and 970 seconds for LLaMA3.2-3B, compared to 13k–25k seconds for large guardrail decoders. Overall, these findings confirm that the proposed methodology not only ensures competitive safety alignment but also drastically reduces the computational overhead for both deployment and retraining, offering a scalable and efficient alternative to fine-tuned guardrails in safeguarding scenarios.