DEFENDING LARGE LANGUAGE MODELS AGAINST DIVERSE JAILBREAK ATTACKS VIA MULTI-CONCEPT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite extensive safety alignment efforts, Large Language Models (LLMs) remain highly susceptible to jailbreak attacks. Although various defenses have been proposed, their effectiveness remains limited against emerging or adaptive threats, largely because the underlying mechanisms of jailbreak attacks remain poorly understood. To elucidate these mechanisms, recent studies leverage the Linear Representation Hypothesis to extract jailbreak concepts from LLM representations. However, our analysis reveals that no single jailbreak concept generalizes across diverse jailbreak techniques, as different attack strategies exhibit distinct representation patterns. To address this limitation, we propose Multi-Concept Detection (MCD), a lightweight defense framework that robustly identifies jailbreak attempts by extracting their multi-concept features. Extensive experiments on five open-source LLMs demonstrate that MCD consistently achieves superior detection performance and remains robust against a wide range of attack strategies.

1 INTRODUCTION

The rapid advancement of Large Language Models (LLMs) has enabled their widespread adoption across diverse applications, ranging from question answering and coding assistance to scientific research and autonomous agents (Achiam et al., 2023; Liu et al., 2024; Yang et al., 2025). To mitigate potential misuse, various safety alignment strategies have been developed, spanning from Reinforcement Learning from Human Feedback (Christiano et al., 2017; Ouyang et al., 2022) to more recent AI-feedback-based approaches (Bai et al., 2022; Lee et al., 2024). Despite the widespread integration of these alignment methodologies in state-of-the-art models, such models remain highly vulnerable to jailbreak attacks (Carlini et al., 2023; Anwar et al., 2024). Whether employing manual prompt engineering (Wei et al., 2023; Shen et al., 2024) , automated attack strategies (Mehrotra et al., 2024; Chao et al., 2025), or adversarial optimization methods (Zou et al., 2023; Andriushchenko et al., 2025; Basani & Zhang, 2026), attackers can effectively bypass existing safeguards and induce LLMs to generate harmful outputs. Such pervasive vulnerabilities pose severe risks to the secure deployment of LLMs, underscoring an urgent need for robust defense mechanisms.

To counter these threats, existing mechanisms can be broadly categorized based on their target of analysis. First, pre-processing approaches aim to detect and filter harmful inputs prior to inference (Alon & Kamfonas, 2023; Robey et al., 2025); however, they remain vulnerable to adaptive attacks. Second, post-processing approaches employ external guardrails or auxiliary models to verify or rectify generated outputs (Inan et al., 2023; Han et al., 2024) . Nonetheless, invoking auxiliary models incurs substantial computational overhead, limiting the feasibility of real-time deployment.

To overcome these limitations, recent studies (Zhou et al., 2024; Hildebrandt et al., 2025; Li et al., 2025; Goren et al., 2026) have increasingly focused on representation-based approaches, which analyze and manipulate internal hidden states during inference to enable early intervention for real-time defense. Among these, Park et al. (2024) formalize the Linear Representation Hypothesis (LRH), which posits that high-level concepts are represented as linear directions in the representation space. Building on this formulation, Zhang et al. (2025) monitors representation concept activations to detect jailbreak attempts in real time.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Despite the efficiency and interpretability of such concept-based defenses, their practical application faces significant challenges. As demonstrated in Section 4.1, our empirical analysis reveals that a single jailbreak concept direction is insufficient to handle diverse jailbreak attacks. This limitation suggests that while LRH provides a solid foundation for concept interpretability, rigidly adhering to linear decision boundaries is overly restrictive. However, directly switching to unconstrained non-linear mappings risks sacrificing the interpretability of representations (Sutter et al., 2026). Thus, the critical question is how to extend LRH beyond the linear assumption to capture complex malicious behaviors without losing interpretability.

To address this challenge, we propose a Multi-Concept Detection (MCD) framework that captures complex malicious patterns without sacrificing interpretability. Instead of relying on a single concept direction, MCD constructs a multi-concept set comprising the harmfulness direction and multiple jailbreak directions. This multi-concept set serves as the projection basis. Given an input prompt, its hidden representations are projected onto each concept direction to produce a structured multi-concept feature. The feature is then fed into a lightweight classifier, enabling MCD to model intricate, non-linear safety boundaries while preserving the interpretability of individual concept projections.

Through extensive evaluations on five open-source LLMs, we show that MCD consistently achieves strong performance across the evaluated settings, ranking at or near the top in most cases. Furthermore, by validating our multi-concept features across five distinct lightweight classifiers, we demonstrate robustness across classifier architectures in our framework.

2 Related Work

2.1 Jailbreak Attacks on LLMs

Jailbreak attacks aim to construct malicious inputs that induce large language models to violate safety guidelines (Das et al., 2025). Early manually designed attacks relied on carefully crafted prompts and role-playing scenarios to circumvent safety mechanisms. To capture real-world vulnerabilities, in-the-wild jailbreak prompts (IJP) characterize the diverse prompt patterns and semantic variations observed in actual deployments (Shen et al., 2024; Yu et al., 2024). Meanwhile, obfuscation-based attacks disguise malicious intent through alternative input representations, such as encoding harmful instructions in Base64 (Wei et al., 2023) or leveraging low-resource languages (LRL) to exploit cross-lingual alignment vulnerabilities (Yong et al., 2023).

Recent studies have increasingly focused on automated jailbreak generation. Optimization-based approaches, including Greedy Coordinate Gradient (GCG) (Zou et al., 2023), Simple Adaptive Attacks (SAA) (Andriushchenko et al., 2025), AutoDAN (Zhu et al., 2024), and Prompt Automatic Iterative Refinement (PAIR) (Chao et al., 2025), iteratively optimize adversarial prompts through gradient guidance or black-box refinement to maximize attack success while improving transferability and readability. To further improve efficiency and target flexibility, recent techniques like GASP (Basani & Zhang, 2026) and AdvPrefix (Zhu et al., 2026) explore continuous latent space optimization and tailored target prefix selection, respectively. Beyond optimization, semantic obfuscation attacks exploit semantic restructuring and implicit framing to conceal malicious intent. Representative examples include DrAttack (Li et al., 2024), which decomposes and reconstructs malicious prompts, and Puzzler (Chang et al., 2024), which masks harmful instructions through partial clues and indirect prompting.

2.2 Jailbreak Detections on LLMs

To safeguard LLMs against malicious exploits, jailbreak detection techniques have evolved progressively deeper. Early paradigms, such as PPL (Jain et al., 2023), focused on shallow input-level statistical anomalies by flagging prompts with high perplexity; however, they remain vulnerable to adaptive attacks. To achieve deeper semantic understanding, subsequent research shifted toward model-based text classification. Specialized guardrail models, such as the Llama-Guard series (Inan et al., 2023; Grattafiori et al., 2024; Llama Team, 2025), and self-evaluating frameworks (Phute et al., 2024; Brown et al., 2024) are employed to assess the harmfulness of generated responses.

More recently, research has shifted toward inspecting the internal mechanisms of LLMs. Li et al. (2025) suggests that the self-safeguarding capability of LLMs is intrinsically associated with distinctive activity patterns in the representation space, which can be identified through contrastive prompt pairs. Based on this insight, JBShield (Zhang et al., 2025) leverages concept activations to detect jailbreak attacks directly within the latent space, while AlignTree (Goren et al., 2026) utilizes both refusal vectors and semantic representations to distinguish malicious prompts.

3 PRELIMINARY

In this section, we provide a review of relevant background knowledge to make this paper self-contained and establish the notation used throughout the paper. Specifically, we introduce the definition of concepts in the context of LLMs, describe their derivation process, and discuss their applications in detection tasks.

3.1 CONCEPT

A concept is fundamentally defined as a factor of variation within data that can change independently of other attributes (Wang et al., 2023). Operationally, such a concept can be characterized by a set of paired exemplars that exhibit a specific semantic relation. For instance, the concept of a capital city can be represented by country-capital pairs, $\{(\text{China}, \text{Beijing}), (\text{Germany}, \text{Berlin}), \dots\}$, where each pair captures a country-to-capital mapping (Mikolov et al., 2013). Similar pairwise formulations have been widely adopted for learning interpretable latent concepts in representation learning (Elhage et al., 2022; Wang et al., 2023). To formalize this structure in the context of LLMs, Park et al. (2024) introduce the Linear Representation Hypothesis (LRH) through a counterfactual framework, under which high-level concepts are represented as linear directions in the representation space. Empirically, Zhang et al. (2025) show that such concept directions can be effectively estimated from concept counterfactual pairs.

3.2 CONSTRUCTION OF CONCEPT DIRECTION

The estimation of concept directions relies on the hidden representations of the target LLM. Let $\mathbf{H}_l(x) \in \mathbb{R}^{m \times d}$ denote the hidden representations of an input prompt x at the l-th transformer layer in the LLM, where m is the sequence length and d is the hidden dimension.

3.2.1 CONCEPT COUNTERFACTUAL PAIR FORMATION

Following the approach proposed by Park et al. (2024), each high-level concept, such as harmfulness or jailbreak intent, can be modeled using concept counterfactual pairs that contrast along the target semantic axis. To illustrate how these pairs are formed, we take harmfulness as an example; the case of jailbreak intent follows the same procedure, differing only in the required prompt sets. Specifically, estimating the harmfulness concept requires two prompt sets of the same size N: a harmful prompt set X^{h} and a benign prompt set X^{b}. In the case of jailbreak intent, a jailbreak prompt set and a harmful prompt set are required instead. While ideal concept counterfactual pairs should differ by only a single word to minimize variance (Li et al., 2025), such fine-grained pairs are rarely observed in real-world datasets. Therefore, alternative pairs are formed by randomly sampling and pairing prompts from X^{h} and X^{b} without any predefined correspondence (Zhang et al., 2025), yielding the sequence $(x_1^{\text{h}}, x_1^{\text{b}}), (x_2^{\text{h}}, x_2^{\text{b}}), \dots, (x_N^{\text{h}}, x_N^{\text{b}})$ to approximate the ideal concept counterfactual setting.

Since prompts consist of discrete tokens, they do not admit a continuous geometric structure amenable to concept direction estimation (Yuan et al., 2023). As a result, the hidden representations of prompts are adopted; more specifically, the last-token representation in $\mathbf{H}_l(x)$ is used as the prompt embedding due to its effectiveness in aggregating context information at the l-th layer and strongly influencing information flow to subsequent layers (Azaria & Mitchell, 2023; Chen et al., 2024). We denote the embedding of prompt x at the l-th layer as $\boldsymbol{e}_l(x) \in \mathbb{R}^d$. Accordingly, each concept counterfactual pair is represented as $(\boldsymbol{e}_l(x_i^{\text{h}}), \boldsymbol{e}_l(x_i^{\text{b}}))$.

3.2.2 CONCEPT DIRECTION EXTRACTION

To extract the concept direction, denoted by $\boldsymbol{v} \in \mathbb{R}^d$, from the concept counterfactual pairs, a semantic difference matrix $\mathbf{D} \in \mathbb{R}^{N \times d}$ is constructed as:

$$\mathbf{D} = \left[\boldsymbol{e}_l(x_1^{\mathrm{h}}) - \boldsymbol{e}_l(x_1^{\mathrm{b}}), \boldsymbol{e}_l(x_2^{\mathrm{h}}) - \boldsymbol{e}_l(x_2^{\mathrm{b}}), \ldots, \boldsymbol{e}_l(x_N^{\mathrm{h}}) - \boldsymbol{e}_l(x_N^{\mathrm{b}})\right]^\top, \tag{1}$$

where each row is the element-wise difference between the embeddings of a paired harmful and benign prompts.

Principal Component Analysis (PCA) is then applied to $\mathbf{D}$, defined in Eq. (1), to obtain the first principal component, which captures the principal semantic variation shared across paired embedding differences. This first principal component serves as the concept direction $\boldsymbol{v}$, which can also be called the harmfulness concept direction.

3.3 USAGE OF CONCEPT DIRECTION

3.3.1 ANCHORING AND SCORING

With the harmfulness concept direction $\boldsymbol{v}$ obtained, the harmfulness of a prompt x can be measured. Before the measurement, the prompt embedding of x needs to be aligned with the coordinate system of the concept direction. To perform the alignment, the anchor is first computed as:

$$\boldsymbol{\mu}_{\mathrm{b}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{e}_l(x_i^{\mathrm{b}}), \tag{2}$$

which is the mean of the embeddings of all prompts in X^{b}. Then, $\boldsymbol{e}_l(x)$ is centered by the anchor $\boldsymbol{\mu}_{\mathrm{b}}$, defined in Eq. (2), yielding the aligned embedding $\tilde{\boldsymbol{e}}_l(x) = \boldsymbol{e}_l(x) - \boldsymbol{\mu}_{\mathrm{b}}$.

Once the embedding is aligned, the intensity of the harmfulness can be measured by computing the cosine similarity between $\tilde{\boldsymbol{e}}_l(x)$ and $\boldsymbol{v}$ as:

$$S_{\boldsymbol{v}}(x) = \frac{\tilde{\boldsymbol{e}}_l(x)^\top \boldsymbol{v}}{\|\tilde{\boldsymbol{e}}_l(x)\| \|\boldsymbol{v}\|}. \tag{3}$$

The value $S_{\boldsymbol{v}}(x) \in [-1, 1]$ directly reflects the intensity of the harmfulness within the prompt. A higher $S_{\boldsymbol{v}}(x)$ indicates a stronger association with the concept direction $\boldsymbol{v}$.

4 METHOD

Although jailbreak concepts have been suggested to exhibit linear structures in the representation space of LLMs (Park et al., 2024), it remains unclear whether a single concept direction can generalize across diverse jailbreak attacks. We hypothesize that no single concept direction generalizes well across diverse jailbreak attacks, as each attack may exhibit distinct, method-specific characteristics that are not captured by a shared concept direction. In this section, we first present empirical evidence to support the hypothesis, and then detail the proposed Multi-Concept Detection (MCD) framework.

4.1 CONCEPT SPACE ANALYSIS

In this study, in addition to a harmful prompt set X^{h} and a benign prompt set X^{b}, we consider K jailbreak attacks and thus have K corresponding jailbreak prompt sets, denoted by $\{X_k^{\mathrm{j}}\}_{k=1}^K$. Each jailbreak prompt set is generated by the corresponding jailbreak attack, with the harmful prompt set as input. Following the concept direction extraction procedure introduced in Section 3, the harmfulness concept direction $\boldsymbol{v}$ is extracted from X^{h} and X^{b}, while, for each k, the jailbreak concept direction $\boldsymbol{u}_k$ is extracted from X_k^{j} and X^{h}.

4.1.1 CONCEPT SPACE VISUALIZATION

We visualize prompts in the two-dimensional space constructed by the harmfulness concept direction and one of the jailbreak concept directions, as shown in Figure 1. For each jailbreak attack, its

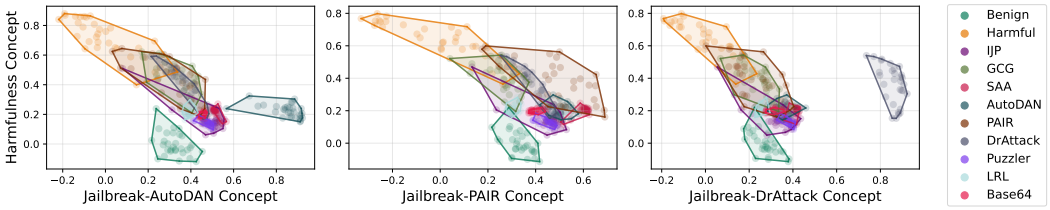


Figure 1: Visualization of prompts from 11 categories mapped into three concept spaces, each spanned by the harmfulness concept direction and one of three jailbreak concept directions, derived from AutoDAN, PAIR, and DrAttack, respectively. (More visualizations in the Appendix A.)

generated prompts align with the corresponding concept direction, albeit with a small fraction falling within the benign or harmful prompt regions. In contrast, prompts generated by other jailbreak attacks are not well-separated in this space and largely overlap with the benign and harmful prompt regions, indicating that each jailbreak concept direction is method-specific rather than universally shared across diverse jailbreak attacks.

This observation suggests that jailbreak semantics vary substantially across attacks and cannot be fully captured by a single concept direction. This motivates the use of multiple concept directions to capture the semantic variations associated with different jailbreak attacks.

4.2 Multi-Concept Detection

To address the limitation of relying on a single jailbreak concept direction, we propose MCD, which determines detection decisions based on multiple jailbreak concept directions. An overview of the proposed framework is presented in Figure 2. The pseudocode is depicted in Appendix B.

4.2.1 Layer Selection

Because hidden representations differ across LLMs, the layer that best separates prompts from different categories may vary across models. Following prior representation-based layer selection methods (Jiang et al., 2025; Zhang et al., 2025), we perform layer selection separately for each target LLM using the average Silhouette score over the comparisons between prompt sets defined during concept construction as the selection criterion. All computations are performed exclusively on the training data; further details are provided in Appendix C. The selected layers are 23, 22, 28, 21, and 17 for Mistral-7B, Vicuna-7B, Vicuna-13B, Llama2-7B, and Llama3-8B, respectively.

4.2.2 Multi-Concept Feature

In brief, MCD makes a jailbreak detection decision for an input prompt through a classifier that takes the multi-concept feature of the prompt as input and outputs the decision. Before detailing how a prompt is mapped to its multi-concept feature, we introduce the multi-concept set, which serves as the key component for computing multi-concept features.

We construct the multi-concept set $\mathcal{C}$ by combining the first P principal components of the harmfulness concept direction with those of the K jailbreak concept directions:

$$\mathcal{C} = \{\boldsymbol{v}^{(p)}, \boldsymbol{u}_1^{(p)}, \ldots, \boldsymbol{u}_K^{(p)} \mid p \in \{1, 2, \ldots, P\}\}. \tag{4}$$

Given a prompt x, its multi-concept feature $\boldsymbol{s}$ is constructed by concatenating the scores computed via Eq. (3) for each element in $\mathcal{C}$ (Eq. (4)):

$$\boldsymbol{s} = \Big\Vert_{p=1}^{P} \left[S_{\boldsymbol{v}^{(p)}}(x),\ S_{\boldsymbol{u}_1^{(p)}}(x),\ \ldots,\ S_{\boldsymbol{u}_K^{(p)}}(x)\right]^\top, \tag{5}$$

where $\Vert$ denotes the vector concatenation operator across all P concept groups.

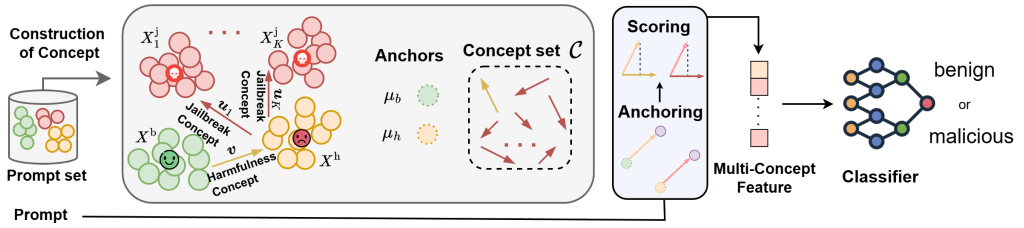


Figure 2: An illustration of Multi-Concept Detection. Once concepts are constructed, the classifier is trained on multi-concept features. During inference, target prompts are passed through the classifier to distinguish between benign and malicious requests.

The multi-concept feature jointly captures the prompt's alignment with the harmfulness concept direction and multiple jailbreak concept directions, allowing the classifier to leverage complementary concept information for jailbreak detection.

4.2.3 Classifier Training

Once the multi-concept features of all training prompts are obtained, we can train a lightweight classifier that takes the feature of a query prompt as input to predict whether it is benign, harmful, or jailbreak. The classifier is trained by minimizing the empirical objective over the training dataset $\mathcal{D}$, which is constructed from X^{b}, X^{h}, and $\{X_k^{\mathrm{j}}\}_{k=1}^{K}$, with each prompt labeled as benign, harmful, or jailbreak accordingly. The minimization problem is formulated as:

$$\min_{\theta} \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \mathcal{L}\left(f_\theta(\boldsymbol{s}), y\right). \tag{6}$$

Here, $\mathcal{L}$ denotes a classifier-specific loss function, f_θ denotes the classifier, parameterized by θ, $y \in \{0, 1, 2\}$ is the label with 0, 1 and 2 representing benign, harmful and jailbreak prompts, respectively, and $\boldsymbol{s}$ is the multi-concept feature of x, as defined in Eq. (5). Note that the proposed framework is classifier-agnostic, i.e., it is compatible with a wide range of classifier architectures.

4.2.4 Usage of Framework

Once the classifier is trained by solving the minimization problem in Eq. (6), MCD can detect whether a prompt is malicious. The detection procedure first maps the prompt to its multi-concept feature, as defined in Eq. (5), and then feeds this feature into the classifier f_θ to determine whether the prompt is benign or malicious. Here, both harmful and jailbreak prompts are considered malicious.

5 Experiments

5.1 Experimental Setup

Dataset We constructed a dataset comprising three prompt categories: benign, harmful, and jailbreak. We sampled 850 benign prompts from Alpaca (Taori et al., 2023) and constructed 850 harmful prompts by combining 520 prompts from AdvBench (Zou et al., 2023) with 330 prompts from Hex-PHI (Qi et al., 2024). For jailbreak prompts, we used nine jailbreak methods. We directly sampled 850 prompts from IJP, while jailbreak prompts for the remaining eight methods were generated using the harmful prompts as target instructions. GCG, SAA, AutoDAN, PAIR, LRL, and Base64 each produced 850 prompts, whereas DrAttack and Puzzler generated only 520 and 50 prompts, respectively, due to OpenAI API cost constraints under their default settings. For concept construction and classifier training, we selected 300 benign prompts, 30 harmful prompts, and 30 prompts from each jailbreak method, reserving all remaining prompts for evaluation.

Table 1: Detection accuracy (%) comparison of different detection methods against various prompts and jailbreak attacks.

LLMs	Methods	Benign	Harmful	Jailbreak Methods								
				IJP	GCG	SAA	AutoDAN	PAIR	DrAttack	Puzzler	LRL	Base64
Mistral-7B	PPL	**100.00**	0.12	0.24	25.24	0.00	0.00	0.49	0.00	0.00	0.24	0.00
	Self-Defense	62.73	71.71	42.23	72.20	0.62	75.37	67.69	11.22	**100.00**	99.76	8.17
	Llama-Guard-3	99.64	97.80	25.49	98.05	**100.00**	60.24	89.15	70.82	90.00	55.24	**100.00**
	Llama-Guard-4	99.64	93.17	46.03	94.15	**100.00**	86.46	85.61	47.96	40.00	81.59	**100.00**
	JBShield	97.82	73.87	81.16	97.20	96.65	95.00	85.12	82.65	**100.00**	99.51	99.76
	AlignTree	**100.00**	80.12	**97.20**	96.34	**100.00**	**100.00**	**93.54**	**100.00**	**100.00**	**99.88**	**100.00**
	MCD	98.81	**97.86**	96.18	**99.35**	**100.00**	**100.00**	92.05	**100.00**	**100.00**	99.57	**100.00**
Vicuna-7B	PPL	95.82	0.12	0.85	69.88	0.00	0.24	0.00	0.00	0.00	30.61	0.00
	Self-Defense	60.91	61.83	64.35	35.85	5.60	19.76	51.83	9.59	70.00	99.88	1.95
	Llama-Guard-3	99.64	**97.80**	25.49	98.54	**100.00**	74.76	82.20	68.78	90.00	55.24	**100.00**
	Llama-Guard-4	99.64	93.17	46.28	96.83	**100.00**	81.10	77.80	47.55	40.00	81.59	**100.00**
	JBShield	96.00	64.33	76.34	90.67	96.89	89.57	88.29	**99.90**	77.50	**99.94**	99.88
	AlignTree	**99.88**	85.73	**94.63**	96.34	**100.00**	99.51	**91.59**	99.80	95.00	98.90	**100.00**
	MCD	99.00	93.84	93.59	**99.18**	**100.00**	**100.00**	89.23	99.86	**100.00**	99.79	**100.00**
Vicuna-13B	PPL	98.00	0.12	0.85	83.90	0.00	0.00	0.85	0.00	0.00	16.46	0.00
	Self-Defense	56.55	54.27	61.31	74.15	17.79	76.10	25.18	17.35	90.00	98.17	0.49
	Llama-Guard-3	**99.64**	**97.80**	25.49	98.05	**100.00**	70.00	78.05	71.02	90.00	55.24	**100.00**
	Llama-Guard-4	**99.64**	93.17	46.28	95.61	**100.00**	80.24	71.46	48.37	40.00	81.59	**100.00**
	JBShield	96.73	73.90	77.38	95.49	94.94	86.22	86.95	93.67	70.00	99.70	99.82
	AlignTree	99.63	94.51	**96.46**	**99.51**	**100.00**	**100.00**	**96.71**	**100.00**	**100.00**	**99.88**	**100.00**
	MCD	98.82	96.20	94.51	97.90	**100.00**	**100.00**	84.52	99.71	**100.00**	99.21	**100.00**
Llama2-7B	PPL	98.73	0.12	0.85	81.34	0.00	0.85	0.00	0.00	0.00	13.41	0.00
	Self-Defense	33.09	31.95	54.87	76.71	3.36	72.93	29.63	96.12	80.00	**100.00**	16.46
	Llama-Guard-3	99.64	97.80	25.49	98.90	**100.00**	97.80	53.41	65.10	90.00	55.24	**100.00**
	Llama-Guard-4	99.64	93.17	46.28	96.83	**100.00**	93.78	52.56	46.94	40.00	81.59	**100.00**
	JBShield	**100.00**	87.34	85.37	98.60	93.96	96.77	88.60	85.31	92.50	91.16	90.61
	AlignTree	99.51	94.63	**98.66**	**99.63**	**100.00**	**100.00**	**96.71**	**100.00**	95.00	99.51	**100.00**
	MCD	99.20	**99.02**	96.85	99.57	99.76	99.98	95.32	99.58	**100.00**	99.99	**100.00**
Llama3-8B	PPL	98.73	0.24	0.98	85.98	0.00	6.71	0.12	0.00	0.00	1.34	0.00
	Self-Defense	73.27	67.56	69.79	84.02	86.44	93.41	74.02	81.84	85.00	99.39	0.37
	Llama-Guard-3	99.64	97.80	25.49	97.68	**100.00**	95.00	50.85	64.49	90.00	55.24	**100.00**
	Llama-Guard-4	99.64	93.17	46.28	96.46	**100.00**	92.80	49.39	49.80	40.00	81.59	**100.00**
	JBShield	**100.00**	56.44	88.29	92.62	99.88	97.32	70.98	96.43	**100.00**	99.51	95.55
	AlignTree	99.76	96.71	**98.66**	**100.00**	**100.00**	**100.00**	**95.73**	**100.00**	**100.00**	**99.88**	**100.00**
	MCD	98.98	**98.23**	97.63	98.82	**100.00**	99.89	88.66	**100.00**	**100.00**	99.52	**100.00**

Concept set Construction Following the concept extraction procedure described in Section 3, we performed PCA separately on each concept pair. Specifically, we applied PCA to the benign-harmful pair to obtain harmfulness concept directions and to each of the nine harmful-jailbreak pairs to obtain the corresponding jailbreak concept directions. For each pair, we retained the first two principal components ($P = 2$) as concept directions. Since there is one benign-harmful pair and nine harmful-jailbreak pairs, the resulting multi-concept set contains $P(K + 1) = 2 \times (9 + 1) = 20$ concept directions in total.

Implementation Details LightGBM (Ke et al., 2017) was selected as our classifier (more classifier results and settings in Appendix D). All experiments were conducted on an NVIDIA RTX 6000 Ada Generation GPU.

LLMs We evaluated our method on five open-source LLMs: Mistral-7B (Jiang et al., 2023), Vicuna-7B/13B (Zheng et al., 2023), Llama2-7B (Touvron et al., 2023), and Llama3-8B (Grattafiori et al., 2024). These models span three model families with different architectures, model sizes, and alignment strategies.

Baselines We compared our method against six baselines: PPL (Jain et al., 2023), Self-Defense (Phute et al., 2024), Llama-Guard-3 (Grattafiori et al., 2024), Llama-Guard-4 (Llama Team, 2025), JBShield (Zhang et al., 2025), and AlignTree (Goren et al., 2026).

Metrics We report detection accuracy separately for benign prompts, harmful prompts, and jailbreak prompts generated by each jailbreak method. For benign prompts, accuracy reflects the ability

Table 2: Detection accuracy (%) of MCD and AlignTree on XSTest and WildGuardMix.

LLMs	Methods	XSTest		WildGuardMix			Mean ± Std
		benign	harmful	benign	harmful	jailbreak	
Mistral-7B	AlignTree	**100.00**	13.50	**99.18**	40.92	**83.21**	67.36 ± 34.44
	MCD	92.42	**86.83**	94.65	**78.35**	76.69	**85.79** ± 7.23
Vicuna-7B	AlignTree	**97.60**	55.50	**97.76**	58.35	72.35	76.31 ± 18.36
	MCD	91.70	**83.87**	94.96	**72.25**	**89.77**	**86.51** ± 7.99
Vicuna-13B	AlignTree	**91.60**	80.50	91.84	75.30	**89.26**	85.70 ± 6.64
	MCD	88.70	**92.50**	**94.10**	**80.39**	79.26	**86.99** ± 6.12
Llama2-7B	AlignTree	76.80	**90.00**	77.14	81.60	**95.56**	84.22 ± 7.40
	MCD	**79.06**	89.60	**81.24**	**88.68**	94.19	**86.56** ± 5.59
Llama3-8B	AlignTree	68.40	**92.50**	80.82	**88.86**	**98.77**	**85.87** ± 10.49
	MCD	**87.40**	78.60	**92.74**	82.80	81.93	84.69 ± 4.91

to avoid false positives, whereas for harmful and jailbreak prompts, it reflects the ability to correctly detect malicious inputs.

5.2 Experimental Results

5.2.1 Detection Performance.

Table 1 reports the detection accuracy of MCD and the baselines across five LLMs and nine jailbreak attacks. For MCD and AlignTree, we train each with 20 random seeds and report the average detection accuracy over the 20 runs. MCD achieves strong performance across the evaluated settings. This indicates that the proposed detector is effective in identifying malicious inputs while maintaining high accuracy on benign prompts.

Among the baselines, PPL exhibits limited effectiveness on harmful and jailbreak prompts, suggesting that perplexity-based signals alone are insufficient for reliable safety detection. Self-Defense performs well on only a subset of jailbreak methods, but its performance declines substantially on the remaining attacks, indicating limited generalization across attack types. Llama-Guard-3 and Llama-Guard-4 achieve high accuracy on benign prompts, reflecting strong control over false positives, but their performance is less stable on several jailbreak methods. Although JBShield delivers competitive results overall, it relies on prior knowledge of the attack type to select the corresponding concept, as shown in Figure 1. This limitation hinders the practicality of JBShield in realistic deployment scenarios. In contrast, MCD does not require prior knowledge of the attack type and instead leverages multiple-concepts for detection. AlignTree also achieves strong performance in many settings, but its accuracy on harmful prompts is noticeably lower than on the other prompt categories, suggesting reduced robustness when malicious content is not expressed in a jailbreak-specific form. These results demonstrate the superior effectiveness of our method in detecting various jailbreaks across different LLMs.

5.2.2 Cross-Dataset Generalization

To assess cross-dataset generalization, we conducted external validation on two unseen datasets, XSTest (Röttger et al., 2024) and WildGuardMix (Han et al., 2024). As shown in Table 2, MCD generally achieves higher detection accuracy on harmful prompts than AlignTree across the evaluated LLMs, with particularly substantial improvements on Mistral-7B and Vicuna-7B. This improvement is accompanied by a moderate decrease in benign accuracy for several models, although MCD also achieves comparable or higher benign accuracy in some cases. For Llama3-8B, however, AlignTree achieves higher harmful prompt accuracy on both datasets, indicating that the advantage of MCD is not uniform across all models. Overall, while the performance trade-off depends on the LLM, MCD shows promising cross-dataset generalization, particularly in improving harmful prompt detection.

5.2.3 Ablation Study

To investigate the effect of the number of principal components, we conducted an ablation study by varying P from 1 to 3.

Table 3: Detection accuracy (%) across principal component configurations of MCD.

LLMs	Principal Component	Benign	Harmful	Jailbreaks	XSTest		WildGuardMix			Mean ± Std
					benign	harmful	benign	harmful	jailbreak	
Mistral-7B	$P = 1$	99.34	95.40	95.79	84.74	94.85	90.81	78.12	79.23	89.79 ± 8.09
	$P = 2$	98.81	97.86	98.57	92.42	86.83	94.65	78.35	76.69	90.52 ± 8.95
	$P = 3$	99.15	96.51	96.82	94.03	76.31	74.48	90.72	89.25	89.66 ± 8.28
Vicuna-7B	$P = 1$	99.72	91.79	97.47	89.14	92.93	94.73	70.73	86.43	90.37 ± 9.01
	$P = 2$	99.00	93.84	97.96	91.70	83.87	94.96	72.25	89.77	90.42 ± 8.77
	$P = 3$	99.75	92.67	100.00	94.18	73.63	87.96	89.22	91.75	91.15 ± 7.34
Vicuna-13B	$P = 1$	99.56	94.85	95.05	75.34	96.37	89.67	82.42	83.06	89.54 ± 8.45
	$P = 2$	98.82	96.20	97.32	88.70	92.50	94.10	80.39	79.26	90.91 ± 7.52
	$P = 3$	99.14	96.71	99.87	92.46	82.41	82.83	83.22	95.57	91.53 ± 6.66
Llama2-7B	$P = 1$	99.02	98.64	99.11	56.72	98.63	80.89	93.50	92.10	89.82 ± 14.74
	$P = 2$	99.20	99.02	99.01	79.06	89.60	81.24	88.68	94.19	91.25 ± 8.02
	$P = 3$	99.25	98.87	99.58	82.97	87.53	94.24	82.92	92.23	92.20 ± 6.20
Llama3-8B	$P = 1$	98.59	98.01	98.01	70.12	98.48	85.69	91.28	64.58	88.10 ± 13.66
	$P = 2$	98.98	98.23	98.28	87.40	78.60	92.74	82.80	81.93	89.87 ± 8.26
	$P = 3$	98.92	94.20	99.96	93.30	77.59	82.94	82.04	88.62	89.70 ± 7.27

As shown in Table 3, increasing P from 1 to 2 consistently improves the mean detection accuracy across all five evaluated LLMs. Although increasing P from 2 to 3 further improves the average accuracy for some LLMs, the gains are marginal. We further analyzed the contributions of each concept feature to detection using SHAP, as detailed in Appendix E. The analysis shows that features from the third principal component contribute considerably less than those from the first two, suggesting that P = 2 offers a good trade-off between detection accuracy and complexity. We further analyzed the contributions of concept features using SHAP, as detailed in Appendix E. The analysis shows that $p = 2$ concepts make a non-negligible contribution to MCD, while the additional contribution from $p = 3$ remains relatively limited. These results support our choice of $P = 2$.

5.2.4 Representation-Space Adversarial Attacks

To evaluate MCD under representation-level adversarial attacks, we employed SCAV (Xu et al., 2024), which leverages a classifier trained on the hidden representations of the target LLM to generate perturbations in its representation space. SCAV is particularly suitable for evaluating the robustness of MCD and AlignTree because both detectors derive their predictions from the hidden representations of the target LLM. As shown in Table 4, MCD is considerably more robust than AlignTree on Llama2-7B, outperforming it by 26.12 percentage points, whereas both methods degrade substantially on Vicuna-7B. Recent studies (Hasan et al., 2024; Wu et al., 2025) suggest that Vicuna-7B learns less stable safety representations than Llama2-7B due to differences in their safety alignment procedures. This result suggests that the robustness of representation-based detectors is closely related to the stability of the underlying safety representations.

Table 4: Detection accuracy (%) of MCD and AlignTree under the SCAV attack.

LLMs	MCD	AlignTree
Vicuna-7B	0.75	6.73
Llama2-7B	75.35	49.23

6 Conclusion and Limitations

In this paper, we introduced Multi-Concept Detection (MCD) to address the vulnerability of LLMs to diverse jailbreak attacks. Addressing the limitation of relying on a single concept, MCD constructs a multi-concept set comprising both the harmfulness direction and diverse jailbreak directions, enabling the detection of varied attacks without requiring prior knowledge of the specific attack type. Nevertheless, MCD remains potentially vulnerable to targeted adversarial attacks, such as SCAV, that are designed to evade these representation signals. Future work could therefore incorporate complementary representation signals, such as refusal-related and semantic features, to strengthen robustness beyond harmfulness and jailbreak concepts. Such extensions could provide a broader representational basis for detecting complex and evolving threats while preserving the framework's lightweight and interpretable nature.

AI USE STATEMENT

In this work, we have not used generative AI tools for helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, and interpreting results, and generating synthetic data sets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for editing the paper to improve readability. We have reviewed all AI-assisted work. All AI-suggested edits were reviewed by the authors to ensure that the original meaning, technical content, and claims were preserved. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023.

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hXA8wqRdyV.

Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, Benjamin L. Edelman, Zhaowei Zhang, Mario Günther, Anton Korinek, Jose Hernandez-Orallo, Lewis Hammond, Eric J Bigelow, Alexander Pan, Lauro Langosco, Tomasz Korbak, Heidi Chenyu Zhang, Ruiqi Zhong, Sean O hEigeartaigh, Gabriel Recchia, Giulio Corsi, Alan Chan, Markus Anderljung, Lilian Edwards, Aleksandar Petrov, Christian Schroeder de Witt, Sumeet Ramesh Motwani, Yoshua Bengio, Danqi Chen, Philip Torr, Samuel Albanie, Tegan Maharaj, Jakob Nicolaus Foerster, Florian Tramèr, He He, Atoosa Kasirzadeh, Yejin Choi, and David Krueger. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=oVTkOs8Pka. Survey Certification, Expert Certification.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Advik Basani and Xiao Zhang. Gasp: Efficient black-box generation of adversarial suffixes for jailbreaking llms. *Advances in Neural Information Processing Systems*, 38:146413–146458, 2026.

Hannah Brown, Leon Lin, Kenji Kawaguchi, and Michael Shieh. Self-evaluation as a defense against adversarial attacks on llms. *arXiv preprint arXiv:2407.03234*, 2024.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36:61478–61500, 2023.

Zhiyuan Chang, Mingyang Li, Yi Liu, Junjie Wang, Qing Wang, and Yang Liu. Play guessing game with llm: Indirect jailbreak attack with implicit clues. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5135–5147, 2024.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Zj12nzlQbz.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Gil Goren, Shahar Katz, and Lior Wolf. Aligntree: Efficient defense against llm jailbreak attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 37417–37425, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Adib Hasan, Ileana Rugina, and Alex Wang. Pruning for protection: Increasing jailbreak resistance in aligned llms without fine-tuning. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 417–430, 2024.

Fabian Hildebrandt, Andreas Maier, Patrick Krauss, and Achim Schilling. Refusal behavior in large language models: A nonlinear perspective. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2025.

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

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494.

Yilei Jiang, Xinyan Gao, Tianshuo Peng, Yingshui Tan, Xiaoyong Zhu, Bo Zheng, and Xiangyu Yue. HiddenDetect: Detecting jailbreak attacks against multimodal large language models via monitoring hidden states. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14880–14893, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.724. URL https://aclanthology.org/2025.acl-long.724/.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=uydQ2W41KO.

Tianlong Li, Zhenghua Wang, Wenhao Liu, Muling Wu, Shihan Dou, Changze Lv, Xiaohua Wang, Xiaoqing Zheng, and Xuan-Jing Huang. Revisiting jailbreaking for large language models: A representation engineering perspective. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3158–3178, 2025.

Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. Drattack: Prompt decomposition and reconstruction makes powerful llms jailbreakers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13891–13913, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

AI @ Meta Llama Team. Llama-guard-4-12b. https://huggingface.co/meta-llama/Llama-Guard-4-12B, 2025.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

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

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=UGpGkLzwpP.

Mansi Phute, Alec Helbling, Matthew Daniel Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. LLM self defense: By self examination, LLMs know they are being tricked. In *The Second Tiny Papers Track at ICLR 2024*, 2024. URL https://openreview.net/forum?id=YoqgcIA19o.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hTEGyKf0dZ.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=laPAh2hRFC.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 1671–1685, 2024.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. The non-linear representation dilemma: Is causal abstraction enough for mechanistic interpretability? *Advances in Neural Information Processing Systems*, 38:149455–149503, 2026.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept algebra for (score-based) text-controlled generative models. *Advances in Neural Information Processing Systems*, 36:35331–35349, 2023.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in neural information processing systems*, 36:80079–80110, 2023.

Jiaqi Wu, Chen Chen, Chunyan Hou, and Xiaojie Yuan. Safeint: Shielding large language models from jailbreak attacks via safety-aware representation intervention. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 8473–8488, 2025.

Zhihao Xu, Ruixuan Huang, Changyu Chen, and Xiting Wang. Uncovering safety risks of large language models through concept activation vector. *Advances in Neural Information Processing Systems*, 37:116743–116782, 2024.

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

Zheng Xin Yong, Cristina Menghini, and Stephen Bach. Low-resource languages jailbreak GPT-4. In *Socially Responsible Language Modelling Research*, 2023. URL `https://openreview.net/forum?id=pn83r8V2sv`.

Zhiyuan Yu, Xiaogeng Liu, Shunning Liang, Zach Cameron, Chaowei Xiao, and Ning Zhang. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4675–4692, 2024.

Lifan Yuan, Yichi Zhang, Yangyi Chen, and Wei Wei. Bridge the gap between cv and nlp! a gradient-based textual adversarial attack framework. In *Findings of the association for computational linguistics: ACL 2023*, pp. 7132–7146, 2023.

Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. {JBShield}: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 8215–8234, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. How alignment and jailbreak work: Explain LLM safety through intermediate hidden states. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2461–2488, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.139. URL https://aclanthology.org/2024.findings-emnlp.139/.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. AutoDAN: Interpretable gradient-based adversarial attacks on large language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=INivcBeIDK.

Sicheng Zhu, Brandon Amos, Yuandong Tian, Chuan Guo, and Ivan Evtimov. Advprefix: An objective for nuanced llm jailbreaks. *Advances in Neural Information Processing Systems*, 38: 89990–90014, 2026.

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

A VISUALIZATION OF THE CONCEPT SPACE

As shown in Figure 3, we present the concept space spanned by the harmfulness direction and selected jailbreak concepts across five LLMs, mapping 11 distinct prompts into each space. To mitigate visual clutter caused by overlapping data points and to clearly delineate class boundaries, we enclose prompts of the same category within their corresponding convex hulls.

Overall, the prompts show no clear separation within these two-dimensional concept spaces. Notably, the Harmful Concept dimension fails to discriminate samples under specific conditions—such as in spaces derived from Llama2-7B (with IJP, AutoDAN) and Llama3-8B (with GCG, AutoDAN, PAIR, Puzzler)—causing the data to collapse into a near-linear pattern.

This dimension collapse likely accounts for why the MCD framework, when trained on Llama2-7B and Llama3-8B, exhibits low benign accuracy during external validation.

B PSEUDOCODE FOR MCD

In this section, we present the pseudocode for Multi-Concept Detection. The training algorithm is detailed in Algorithm 1, and the usage algorithm is detailed in Algorithm 2.

C DETAILED LAYER SELECTION PROCEDURE

To determine the layer for each target LLM, we project all training prompts into the multi-concept feature space and evaluate the separability of different prompt categories using the Silhouette score (Rousseeuw, 1987).

Specifically, following the comparisons between prompt sets defined during concept construction (e.g., Benign vs. Harmful and Jailbreak vs. Harmful), we compute the Silhouette score for each concept at each layer. We then average these scores to obtain an overall measure of feature separability. Figure 4 presents the average Silhouette scores across layers for five LLMs; for each LLM, the layer with the highest average score is selected as the target layer for MCD training.

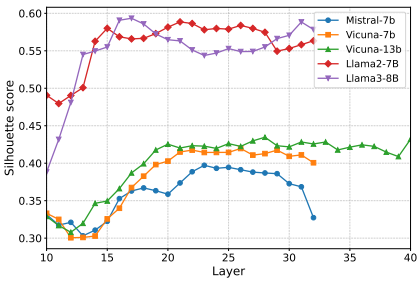


Figure 4: Average Silhouette scores across layers for five LLMs. Peaks indicate selected layers.

D CLASSIFIER COMPATIBILITY

D.1 CLASSIFIER COMPATIBILITY EXPERIMENTS

To evaluate the compatibility and robustness of the proposed multi-concept features across different classifier architectures, we conduct experiments using five representative lightweight models: LinearSVC (LSVC), SVM (Cortes & Vapnik, 1995), XGBoost (XGB) (Chen & Guestrin, 2016), LightGBM (LGBM) (Ke et al., 2017), and MLP (Rumelhart et al., 1986). All experiments are evaluated across 20 random seeds (1–20), with each seed determining model initialization. Detailed hyperparameter configurations are provided in Appendix D.2.

As shown in Table 5, the results are reported as mean $\pm$ standard deviation over the 20 seeds. All five classifiers consistently achieve high detection accuracy on the concept-domain datasets (Benign, Harmful, and Jailbreaks), demonstrating the robustness of the proposed multi-concept features across different classifier architectures. This robustness can be attributed to the structured representation of the features, where each dimension corresponds to a concept activation score obtained by projecting prompt embeddings onto a specific concept direction. Such a representation preserves information about concepts in an explicit and discriminative feature space. The resulting category-level separation is also qualitatively illustrated in Figure 5, providing a favorable basis for downstream classifiers with different architectures.

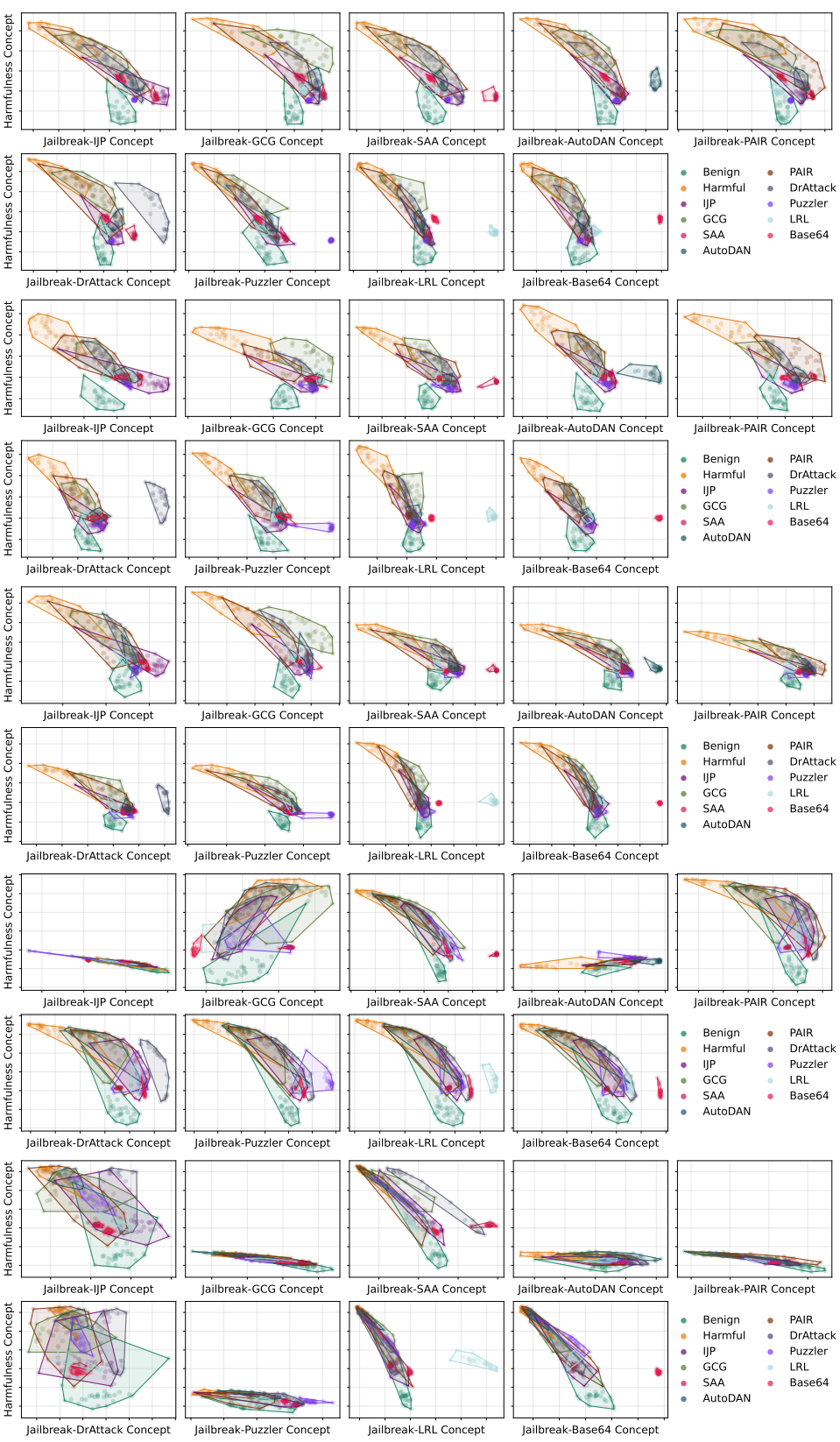


Figure 3: Visualization of the concept space across five LLMs: (top to bottom) Mistral-7b, Vicuna-7b, Vicuna-13b, Llama2-7B, and Llama3-8B.

Algorithm 1 Training of Multi-Concept Detection

Input: harmful prompt set X^{h}, benign prompt set X^{b}, jailbreak prompt sets $\{X_k^{\mathrm{j}}\}_{k=1}^{K}$, selected layer l.
Output: multi-concept set $\mathcal{C}$, anchors $\boldsymbol{\mu}_{\mathrm{b}}$ and $\boldsymbol{\mu}_{\mathrm{h}}$, classifier f_θ.

Step 1: Concept Extraction
$\boldsymbol{v} \leftarrow \mathrm{Concept}(X^{\mathrm{h}}, X^{\mathrm{b}})$
for $k = 1$ **to** K **do**
  $\boldsymbol{u}_k \leftarrow \mathrm{Concept}(X_k^{\mathrm{j}}, X^{\mathrm{h}})$
end for
$\mathcal{C} = \{\boldsymbol{v}^{(p)}, \boldsymbol{u}_1^{(p)}, \ldots, \boldsymbol{u}_K^{(p)} \mid p \in \{1, 2\}\}$
Step 2: Anchor Construction
$\boldsymbol{\mu}_{\mathrm{b}} \leftarrow \mathrm{Anchoring}(X^{\mathrm{b}})$ *(Eq. 2)*
$\boldsymbol{\mu}_{\mathrm{h}} \leftarrow \mathrm{Anchoring}(X^{\mathrm{h}})$ *(Eq. 2)*
Step 3: Training Dataset Construction
Initialize training dataset $\mathcal{D} \leftarrow \emptyset$
for each benign prompt $x \in X^{\mathrm{b}}$ **do**
  $\boldsymbol{s} \leftarrow \mathrm{MultiConceptFeature}(x, \mathcal{C}, \boldsymbol{\mu}_{\mathrm{b}}, \boldsymbol{\mu}_{\mathrm{h}})$ *(Eq. 5)*
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{s}, 0)\}$
end for
for each harmful prompt $x \in X^{\mathrm{h}}$ **do**
  $\boldsymbol{s} \leftarrow \mathrm{MultiConceptFeature}(x, \mathcal{C}, \boldsymbol{\mu}_{\mathrm{b}}, \boldsymbol{\mu}_{\mathrm{h}})$ *(Eq. 5)*
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{s}, 1)\}$
end for
for $k = 1$ **to** K **do**
  for each jailbreak prompt $x \in X_k^{\mathrm{j}}$ **do**
    $\boldsymbol{s} \leftarrow \mathrm{MultiConceptFeature}(x, \mathcal{C}, \boldsymbol{\mu}_{\mathrm{b}}, \boldsymbol{\mu}_{\mathrm{h}})$ *(Eq. 5)*
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{s}, 2)\}$
  end for
end for
Step 4: Classifier Training
Train classifier f_θ on $\mathcal{D}$ *(Eq. 6)*
Return: $\mathcal{C}$, $\boldsymbol{\mu}_{\mathrm{b}}$, $\boldsymbol{\mu}_{\mathrm{h}}$, f_θ

Algorithm 2 Usage of Multi-Concept Detection

Input: Query prompt x, classifier f_θ, multi-concept set $\mathcal{C}$, anchors $\boldsymbol{\mu}_{\mathrm{b}}$ and $\boldsymbol{\mu}_{\mathrm{h}}$.
Output: Predicted label $\hat{y} \in \{0, 1\}$.

$\boldsymbol{s} \leftarrow \mathrm{MultiConceptFeature}(x, \mathcal{C}, \boldsymbol{\mu}_{\mathrm{b}}, \boldsymbol{\mu}_{\mathrm{h}})$ *(Eq. 5)*
$y \leftarrow f_\theta(\boldsymbol{s})$
if $y \in \{1, 2\}$ **then**
  $\hat{y} \leftarrow 1$
else
  $\hat{y} \leftarrow 0$
end if
Return: $\hat{y}$

Furthermore, on the external datasets (XSTest and WildGuardMix), XGBoost and LightGBM achieve stronger and more stable performance than the other classifiers. This may be because tree-based models can selectively exploit informative concept dimensions while reducing the influence of less relevant ones, leading to better generalization on external datasets. As shown in Figure 6, SHAP analysis (Lundberg & Lee, 2017; Lundberg et al., 2020) further shows that the top-6 concepts account for a substantial portion of the feature importance, indicating their critical role in the detection decision.

Table 5: Detection accuracy (%) of different classifiers using the proposed multi-concept features.

LLM	CLF	Benign	Harmful	Jailbreak	XSTest		WildGuardMix		
					benign	harmful	benign	harmful	jailbreak
Mistral-7B	LSVC	98.91 ± 0.39	94.16 ± 2.25	97.37 ± 1.59	80.24 ± 11.24	95.72 ± 4.39	89.59 ± 5.71	80.47 ± 2.25	54.20 ± 11.50
	SVM	96.68 ± 1.86	97.46 ± 0.72	97.74 ± 0.40	69.52 ± 8.96	99.15 ± 0.80	84.08 ± 5.99	85.81 ± 4.21	64.86 ± 3.61
	MLP	97.05 ± 1.89	97.62 ± 1.51	98.40 ± 0.76	80.88 ± 14.23	90.92 ± 8.05	88.65 ± 7.85	82.57 ± 6.48	71.79 ± 9.07
	XGB	98.79 ± 0.25	97.80 ± 0.60	98.63 ± 0.22	91.72 ± 2.29	89.95 ± 7.26	94.50 ± 0.94	79.02 ± 3.20	77.99 ± 1.72
	LGB	98.81 ± 0.28	97.86 ± 0.55	98.57 ± 0.20	92.42 ± 2.38	86.83 ± 7.68	94.65 ± 1.12	78.35 ± 3.09	76.69 ± 2.00
Vicuna-7b	LSVC	98.67 ± 1.15	93.55 ± 3.08	98.05 ± 0.50	95.70 ± 2.54	68.08 ± 27.62	97.43 ± 1.41	67.78 ± 9.66	88.93 ± 1.95
	SVM	96.26 ± 1.56	93.48 ± 1.66	95.91 ± 0.70	86.54 ± 3.04	89.88 ± 3.44	87.68 ± 4.42	72.31 ± 3.40	68.79 ± 2.53
	MLP	94.42 ± 4.47	94.27 ± 2.14	97.38 ± 1.35	82.70 ± 9.97	87.07 ± 19.21	86.72 ± 13.11	76.22 ± 7.17	84.19 ± 6.15
	XGB	98.69 ± 0.29	93.55 ± 1.03	97.94 ± 0.31	92.24 ± 2.06	82.42 ± 3.03	95.51 ± 1.53	71.44 ± 2.50	89.42 ± 1.08
	LGB	99.00 ± 0.20	93.84 ± 1.07	97.96 ± 0.31	91.70 ± 3.82	83.87 ± 4.58	94.96 ± 2.15	72.25 ± 3.62	89.77 ± 1.02
Vicuna-13b	LSVC	99.09 ± 0.48	98.07 ± 0.32	98.09 ± 0.25	88.70 ± 2.47	96.23 ± 4.00	91.68 ± 1.71	84.24 ± 1.49	80.78 ± 2.55
	SVM	98.11 ± 0.98	96.75 ± 1.12	95.60 ± 0.51	83.60 ± 5.47	97.65 ± 0.73	91.39 ± 3.16	82.46 ± 3.09	55.18 ± 2.98
	MLP	96.64 ± 2.93	97.27 ± 0.92	96.89 ± 1.01	82.88 ± 6.58	96.15 ± 1.99	91.88 ± 2.46	82.48 ± 3.23	69.03 ± 6.37
	XGB	98.83 ± 0.30	96.40 ± 1.28	97.36 ± 0.53	88.48 ± 4.25	93.05 ± 4.53	93.96 ± 2.89	80.86 ± 4.39	79.04 ± 1.78
	LGB	98.82 ± 0.31	96.20 ± 1.24	97.32 ± 0.50	88.70 ± 4.55	92.50 ± 4.48	94.10 ± 2.69	80.39 ± 4.22	79.26 ± 1.85
Llama2-7B	LSVC	97.98 ± 0.67	99.19 ± 0.07	99.11 ± 0.11	45.68 ± 2.46	99.55 ± 0.15	72.85 ± 1.20	97.07 ± 0.28	96.31 ± 1.02
	SVM	98.96 ± 0.24	99.01 ± 0.10	97.88 ± 0.09	68.56 ± 1.79	99.28 ± 0.34	84.10 ± 0.69	92.89 ± 0.36	82.59 ± 0.78
	MLP	98.00 ± 1.02	99.25 ± 0.29	96.66 ± 7.37	54.78 ± 7.88	99.43 ± 0.34	78.51 ± 4.40	95.87 ± 1.38	87.57 ± 7.82
	XGB	99.14 ± 0.34	98.93 ± 0.25	99.05 ± 0.37	77.58 ± 14.69	85.25 ± 7.64	80.58 ± 4.67	87.49 ± 5.40	93.78 ± 1.65
	LGB	99.20 ± 0.33	99.02 ± 0.23	99.01 ± 0.31	79.06 ± 13.85	89.60 ± 8.09	81.24 ± 4.73	88.68 ± 5.61	94.19 ± 1.95
Llama3-8B	LSVC	95.59 ± 5.58	97.76 ± 0.50	98.28 ± 0.40	89.62 ± 2.22	94.73 ± 1.36	96.16 ± 1.19	83.24 ± 2.38	87.87 ± 5.18
	SVM	97.62 ± 2.03	97.04 ± 0.13	95.64 ± 0.17	93.50 ± 0.69	82.50 ± 1.19	97.63 ± 0.47	78.89 ± 0.68	53.48 ± 0.86
	MLP	96.04 ± 2.69	96.19 ± 0.65	95.18 ± 1.57	86.86 ± 5.34	91.70 ± 5.48	96.80 ± 2.12	79.35 ± 4.06	53.83 ± 8.70
	XGB	98.85 ± 0.47	97.86 ± 0.88	98.15 ± 0.27	89.56 ± 3.51	75.65 ± 4.20	93.49 ± 1.61	81.13 ± 3.03	82.91 ± 3.03
	LGB	98.98 ± 0.44	98.23 ± 0.80	98.28 ± 0.26	87.40 ± 3.67	78.60 ± 5.06	92.74 ± 1.55	82.80 ± 3.09	81.93 ± 2.68

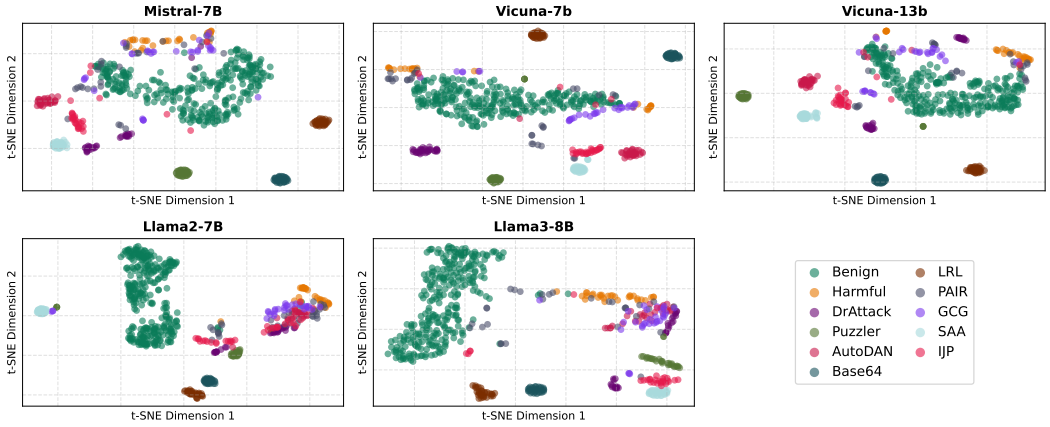


Figure 5: t-SNE visualization of multi-concept features (training data, seed=0) across five LLMs. Features naturally separate by prompt category, with most jailbreak prompts forming a single distinct cluster.

D.2 Classifier Setting

To ensure reproducibility, we evaluated five representative lightweight models using the detailed hyperparameter configurations presented in Table 6.

E SHAP Analysis of MCD

To verify whether MCD indeed utilizes concepts derived from the second principal component ($p = 2$), we measured feature importance by computing the mean absolute SHAP value across each dataset, as presented in Figure 6. We observe that while the second principal component concepts rarely rank among the top-6 features in Mistral-7B, Vicuna-7B, and Vicuna-13B, they consistently rank near the top in Llama2-7B and Llama3-8B.

To compare the reliance on concepts derived from different principal components, we calculated the relative mean absolute SHAP value contribution for each concept group. As shown in Figure 7, for the $P = 2$ model, $p = 2$ concepts account for approximately 9% of the total feature importance for Mistral-7B, Vicuna-7B, and Vicuna-13B, while their contribution increases to approximately

Table 6: Hyperparameter configurations for the lightweight classifiers.

Model (*package*)	**Hyperparameters**
`LinearSVC` *(scikit-learn)*	`C=0.1, class_weight='balanced', random_state=seed, dual=False, max_iter=2000`
`SVC` *(scikit-learn)*	`C=0.1, class_weight='balanced', random_state=seed`
`MLPClassifier` *(scikit-learn)*	`hidden_layer_sizes=(64, 32), max_iter=500, learning_rate_init=0.001, solver='adam', alpha=0.01, random_state=seed`
`XGBClassifier` *(xgboost)*	`n_estimators=500, learning_rate=0.01, max_depth=3, colsample_bytree=0.6, subsample=0.8, tree_method='exact', random_state=seed`
`LGBMClassifier` *(lightgbm)*	`n_estimators=500, learning_rate=0.01, max_depth=3, num_leaves=7, colsample_bytree=0.8, colsample_bynode=0.8, subsample=0.8, subsample_freq=1, min_child_samples=10, reg_alpha=0.1, reg_lambda=1.5, class_weight='balanced', random_state=seed`

25% for Llama2-7B and Llama3-8B. This demonstrates that $p = 2$ concepts make a non-negligible contribution to MCD's decisions, although their relative importance varies across LLMs. We further analyze the $P = 3$ model to examine whether the additional third principal component provides substantial feature contributions. As shown in Figure 7, $p = 3$ contributes between 5.3% and 11.3% across the five LLMs, while $p = 2$ remains an important contributor, particularly for Llama2-7B and Llama3-8B. These results suggest that the additional contribution from $p = 3$ is relatively limited compared with the information already captured by the first two principal components.



Figure 6: Top-6 concept importances for MCD across five target LLMs.

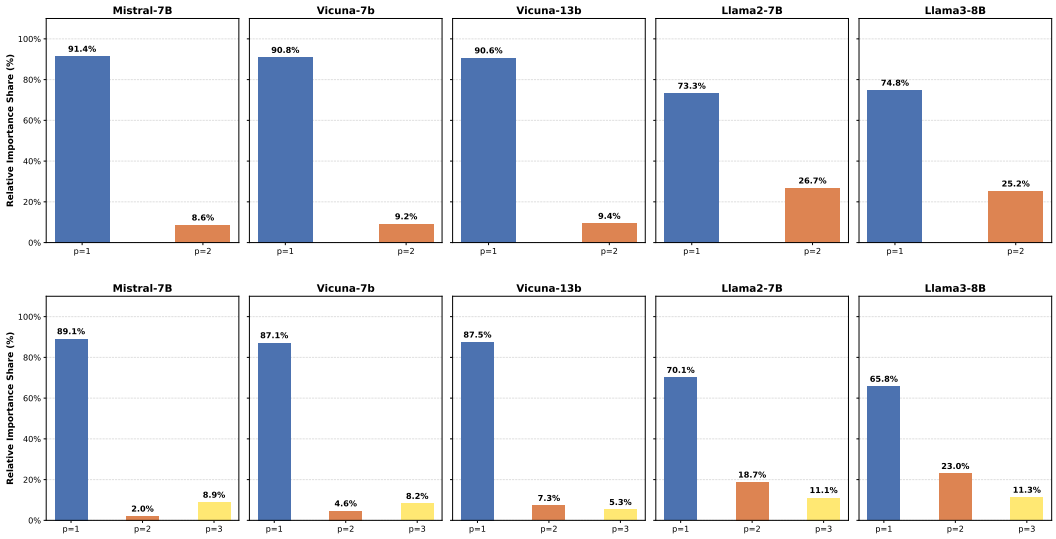


Figure 7: Relative mean absolute SHAP value contribution of principal components for MCD with $P = 2$ (top) and $P = 3$ (bottom) across five LLMs.