Learning to Detect Unknown Jailbreak Attacks in Large Vision-Language Models: A Unified and Accurate Approach

Shuang Liang[1], Zhihao Xu[1], Jialing Tao[2], Hui Xue[2], Xiting Wang[1]*

[1]Renmin University of China
[2]Alibaba Group

Abstract

Despite extensive alignment efforts, Large Vision-Language Models (LVLMs) remain vulnerable to jailbreak attacks, posing serious safety risks. Although recent detection works have shifted to internal representations due to their rich cross-modal information, most methods rely on heuristic rules rather than principled objectives, resulting in suboptimal performance. To address these limitations, we propose Learning to Detect (LoD), a novel unsupervised framework that formulates jailbreak detection as anomaly detection. LoD introduces two key components: Multi-modal Safety Concept Activation Vectors (MSCAV), which capture layer-wise safety-related representations across modalities, and the Safety Pattern Auto-Encoder, which models the distribution of MSCAV derived from safe inputs and detects anomalies via reconstruction errors. By training the auto-encoder (AE) solely on safe samples without attack labels, LoD naturally identifies jailbreak inputs as distributional anomalies, enabling accurate and unified detection of jailbreak attacks. Comprehensive experiments on three different LVLMs and five benchmarks demonstrate that LoD achieves state-of-the-art performance, with an average AUROC of 0.9951 and an improvement of up to 38.89% in the minimum AUROC over the strongest baselines.

Introduction

In recent years, Large Vision-Language Models (LVLMs) have advanced rapidly (Zhang et al. 2024), yet their integration of vision modules introduces significant safety risks (Gu et al. 2024; Luo et al. 2024). Recent studies show that LVLMs remain highly vulnerable to jailbreak attacks, with attack success rates reaching up to 96% (Wang et al. 2024). Unlike Large Language Models (LLMs), LVLMs accept visual inputs that form a continuous, high-dimensional attack surface, where subtle adversarial perturbations in images can bypass safety filters undetectably (Qi et al. 2024).

To mitigate safety vulnerabilities in LVLMs, a variety of detection methods have emerged as promising defense strategies aimed at identifying potential jailbreak attempts in user inputs. Pioneer approaches primarily leverage model inputs and outputs for attack detection (Xu et al. 2024a; Alon and Kamfonas 2023; Zhang et al. 2025). Recently, researchers have found that the model's internal representations capture complex cross-modal semantics and enable detection performance improvement (Xie et al. 2024; Jiang et al. 2025). While these emergent methods show the promises of internal representations in providing fine-grained information for attack detection, they fail to achieve robust detection for all types of jailbreak attacks (e.g., adversarial perturbation-based and prompt manipulation-based attacks, as categorized in (Liu et al. 2024a)) due to their heuristic nature. Instead of learning optimal parameters for distinguishing internal patterns of attacks from safe inputs, these methods directly adopt potentially biased heuristic rules for identifying attacks, which do not require any additional learning or optimization objectives. For example, HiddenDetect (Jiang et al. 2025) considers models as attacked if their internal representations are closer to those of refusal tokens (e.g., "sorry") than safe inputs w.r.t. cosine similarity. The lack of a principal optimization goal and the potential biased understanding of the internal representation patterns result in a suboptimal performance. For example, the AUROC of HiddenDetect drops to approximately 0.7 in certain attacks (see Table 1). These methods favor heuristic rules over learning-based strategies, since supervised learning requires attack distributions and labels, which are intentionally treated as unknown to prevent overfitting to specific attack types.

In this paper, we address the aforementioned challenge by proposing a method for **L**earning t**o D**etect (**LoD**) without knowing the specific attack methods, enabling *accurate* and *unified* detection—that is, a single, consistent approach that generalizes across diverse attacks, including those yet to be developed. The basic idea is to detect attacks through unsupervised learning. This is achieved by formulating attack detection as an anomaly detection problem, where attacks are considered anomalies whose internal patterns are significantly different from those of the normal, safe inputs. The key here is to learn an accurate representation of safe activation patterns (G1) and an effective measure to separate safe activation patterns from attack ones (G2). The former (G1) is achieved by learning **Multi-modal Safety Concept Activation Vectors** (MSCAV), which extend the Safety Concept Activation Vector (SCAV) for LLMs (Xu et al. 2024b) to multi-modal scenarios. MSCAV accurately estimates the probability that the model considers an input as unsafe at all layers, through a faithful interpretation of

*Corresponding to xitingwang@ruc.edu.cn

the model's internal safety mechanisms, therefore providing fine-grained layer-wise information that enables accurate separation of safe inputs and attacks. The latter goal (G2) is fulfilled through proposing a **Safety Pattern Auto-Encoder**, which is learned in an unsupervised manner by using MSCAV of merely safe inputs, without relying on information about attacks. The reconstruction error of the auto-encoder (AE) serves as a good measure for anomaly detection (Sakurada and Yairi 2014): low reconstruction error corresponds to safe inputs that are sufficiently modeled by the AE, while high reconstruction error corresponds to attacks (anomalies) that have not been learned by the AE.

Overall, our contributions can be summarized as:

- Multi-modal Safety Concept Activation Vectors, which provide fine-grained layer-wise information that enables accurate separation of safe inputs and attacks through faithful interpretation of models' safety mechanisms.
- Safety Pattern Auto-Encoder, which is learned by using only safe inputs, yet their reconstruction errors provide an effective measure for attacks (anomalies).
- Comprehensive experiments show that our method achieves state-of-the-art jailbreak detection performance without any prior knowledge of attack types. We report AUROC averaged over five attack methods for each of the three models, achieving scores ranging from 0.9943 to 0.9969. This corresponds to relative improvements of up to 38.89% in the minimum AUROC and 18.21% in the average AUROC compared to the strongest baselines.

Methodology

Problem Definition

Given an LVLM, our goal is to identify jailbreak attacks based on the model's internal activations. In this paper, we distinguish between two types of unsafe inputs: naturally harmful content (e.g., violent images and instructions) and malicious content, which involves deliberate and adversarial jailbreak attacks.

Input: Our method takes as input the intermediate activations[1] $\{\mathbf{e}^1, \ldots, \mathbf{e}^L\}$, where each $\mathbf{e}^l \in \mathbf{R}^d$ denotes the activation vector of a multimodal input I at the l-th layer. These activations are obtained from a forward pass of I through an LVLM f, and d denotes the hidden dimension of the model.

Output: Based on internal activations, our method determines whether the multimodal input I contains a jailbreak attack. We define a jailbreak attack as any input manipulation that induces the model to generate responses that violate its safety alignment, consistent with prior definitions in the literature (Liu et al. 2024a). Our approach does not rely on prior knowledge of specific jailbreak methods, enabling it to generalize across unseen adversarial scenarios.

Method Overview

Our **Learning to Detect (LoD)** framework is illustrated in Figure 1. It takes as input the model's internal activations derived from I and predicts whether I contains jailbreak attacks. Instead of using raw activations, we transform them into **Multi-modal Safety Concept Activation Vectors** (MSCAV) through a set of layer-specific classifiers, which provide accurate and fine-grained safety representations for distinguishing safe inputs from attacks. Each element of the MSCAV estimates the probability that the model considers I unsafe. The MSCAV are then fed into a **Safety Pattern Auto-Encoder** to enable unsupervised detection across unknown jailbreak scenarios. The auto-encoder (AE) is trained exclusively on safe samples and performs anomaly detection by computing reconstruction errors. Adversarial inputs deviate from the learned safe distribution and are thus viewed as anomalies. The reconstruction loss of the AE can then serve as an effective measure for anomaly detection (Sakurada and Yairi 2014). In the following sections, we detail these two key components: Multi-modal Safety Concept Activation Vectors and Safety Pattern Auto-Encoder.

Multi-modal Safety Concept Activation Vectors

This module aims to extract the most discriminative information from the activations to distinguish between safe and adversarial inputs. Achieving this requires transforming the activations $\{\mathbf{e}^1, \ldots \mathbf{e}^L\}$ to filter out safety-irrelevant features (e.g., topics or general semantics), preserving only safety-related representations. However, existing approaches fall short of this goal. HiddenDetect (Jiang et al. 2025) directly uses raw activations without any processing before computing similarity, while GradSafe (Xie et al. 2024) heuristically selects safety-critical parameters without theoretical justification and shows limited effectiveness in Table 1.

To achieve the goal of accurately extracting safety-related information, we introduce **Multi-modal Safety Concept Activation Vectors** (MSCAV), an extension of SCAV (Xu et al. 2024b). SCAV shows that in aligned LLMs, the internal activations of safe and harmful inputs become linearly separable starting from around the 10th layer. Given internal activations in a layer, SCAV accurately outputs the probability that the model considers the input unsafe at that layer by faithfully interpreting the model's safety mechanism. The probabilities are effective for detecting attacks, as they filter out safety-irrelevant information (e.g., topics) while preserving safety-related information. Next, we discuss how we extend SCAV to multi-modal settings, verify that the linear interpretability still holds in LVLMs, and demonstrate MSCAV's effectiveness in distinguishing safe inputs from attacked ones.

MSCAV estimation and refinement. To adapt SCAV from LLMs to the multimodal setting, we construct two sets of multimodal inputs: $\mathbf{I}^+$, representing safe inputs where both text and image content are safe, and $\mathbf{I}^-$, representing harmful inputs in which the text contains naturally harmful content (e.g., "how to make a bomb") and the image depicts a matching harmful scene (e.g., an image of a bomb).

Under the linear separability hypothesis (validated in Figure 2), safe and harmful activations $\mathbf{e}^l$ are considered linearly separable in layer l if the test accuracy of the linear classifier exceeds a threshold P_0. This assumes that activations can be linearly mapped to represent a concept (the con-

[1]This refers to the hidden states at the output of each transformer layer.

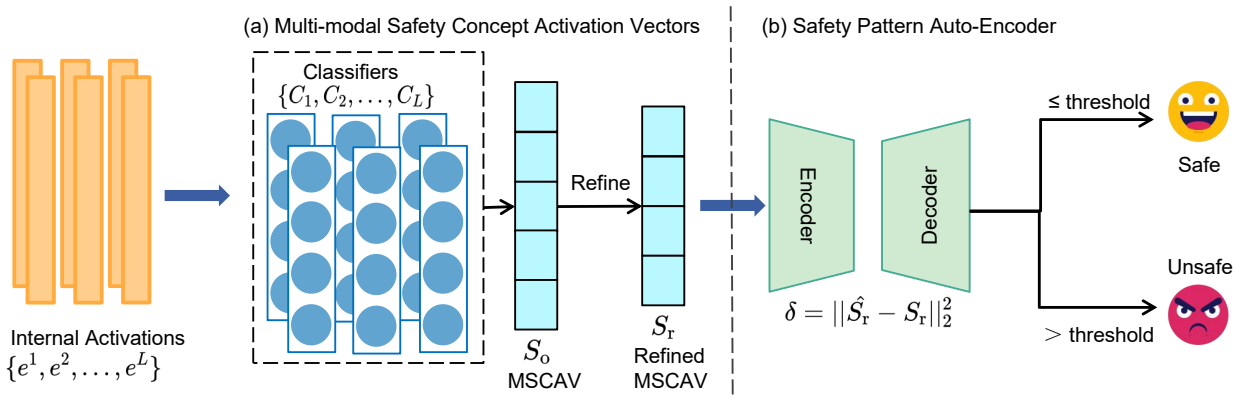

Figure 1: Overview of the Learning to Detect pipeline. Our method leverages internal activations as input, exploiting their rich information to detect attacks. These activations are transformed into Multi-modal Safety Concept Activation Vectors (MSCAV) through a set of layer-specific classifiers, each producing continuous outputs that serve as layer-wise safety features. To improve reliability, we remove outputs from layers that perform poorly in distinguishing safe inputs from naturally harmful (non-attack) inputs on validation data, ensuring that only the most informative representations are retained. The refined MSCAV are then fed into a Safety Pattern Auto-Encoder, trained exclusively on safe samples to effectively model the differences between safe and adversarial MSCAV patterns. Finally, the auto-encoder detects jailbreak attacks by computing reconstruction errors on the refined MSCAV, enabling accurate detection across different attack scenarios.

cept of "safety" in our paper) that provides insight into the model's internal safety mechanisms. We therefore estimate the probability that the model regards an input as unsafe in the layer l through a linear classifier:

$$C_l(\mathbf{e}^l) = \text{sigmoid}(\mathbf{w}^\top \mathbf{e}^l + b), \tag{1}$$

where $\mathbf{w} \in \mathbf{R}^d$ and $b \in \mathbf{R}$. A total of L classifiers $\{C_1, \ldots, C_L\}$ are trained, each capturing the safety concept in its corresponding layer below linear separability. Specifically, we train each classifier C_l using a binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{|N|} \sum \big[y \log C(\mathbf{e}) + (1-y) \log(1 - C(\mathbf{e})) \big], \tag{2}$$

where $y \in \{0, 1\}$ indicates safe (0) or unsafe (1), and N is the data size. The activations and corresponding labels are extracted from the training subsets of $\mathbf{I}^+$ and $\mathbf{I}^-$. The continuous outputs of all classifiers provide a fine-grained view of input safety across layers, which we define as the MSCAV:

$$\mathbf{S}_o = \big[C_1(\mathbf{e}^1), C_2(\mathbf{e}^2), \ldots, C_L(\mathbf{e}^L) \big]^\top, \tag{3}$$

where $\mathbf{S}_o \in \mathbf{R}^L$ is a compact representation that contains only safety-related information.

To improve reliability, we refine $\mathbf{S}_o$ by retaining only the layers where the classifiers achieve test accuracy above the threshold P_0 as shown in Figure 2, that is, $\mathcal{L}_s = \{l \mid \mathcal{P}_l \geq P_0\}$. The remaining layers are considered linearly separable with respect to the safety concept, and the refined MSCAV $\mathbf{S}_r \in \mathbf{R}^{|\mathcal{L}_s|}$ is constructed as:

$$\mathbf{S}_r = \big[C_l(\mathbf{e}^l) \big]_{l \in \mathcal{L}_s}^\top. \tag{4}$$

Verifying linear interpretability in LVLMs. We evaluate the linear separability of inputs in LVLMs by training classifiers in each layer and measuring their accuracy in distinguishing safe inputs (I^+) from harmful ones (I^-).

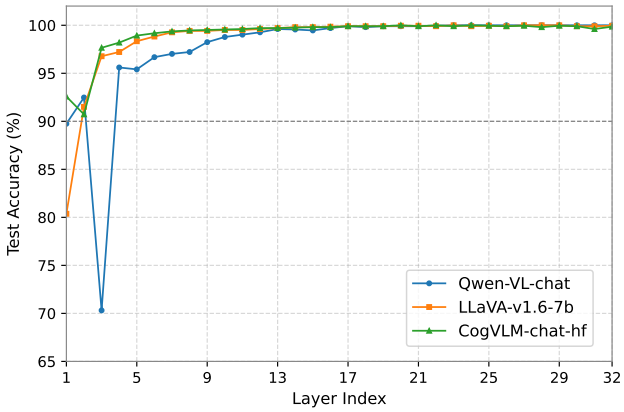

Figure 2: Test accuracy of linear classifiers distinguishing safe and harmful inputs across intermediate layers in LVLMs.

As shown in Figure 2, classifiers achieve consistently high accuracy across layers, indicating that the safe and harmful inputs are well separated in the activation space. Remarkably, the accuracy exceeds 90% as early as the 4th layer, much earlier than the ∼10th layer typically observed in LLMs. This result validates the linear interpretability in LVLMs and suggests that visual information enables safety-related distinctions to emerge earlier.

Effectiveness of MSCAV in distinguishing attacks. We

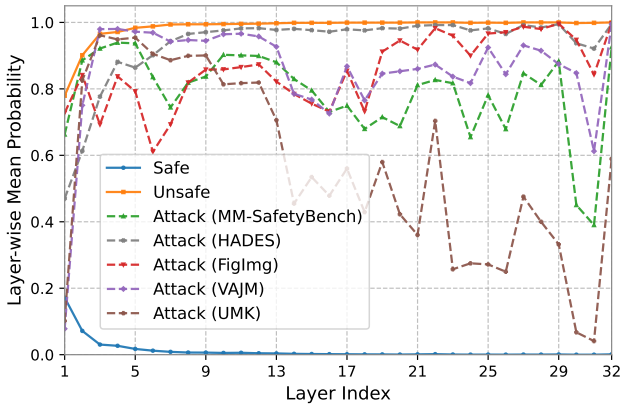

Figure 3: Layer-wise mean probability values for safe and harmful queries. Curves for five unseen jailbreak attack methods are also shown, illustrating clear separability between safe and malicious inputs.

analyze the MSCAV patterns of jailbreak attacks that the linear classifiers have never encountered during training. As shown in Figure 3, malicious attack curves generally lie below those of harmful inputs but remain well above those of safe inputs. This suggests that attacks reduce safety probabilities across layers but remain clearly distinguishable from safe inputs (e.g., their probabilities stay far from zero). A plausible explanation is that attacks optimize for final success without substantially reshaping internal representations across all layers. Using layer-wise safety judgments, MSCAV provide effective safety representations that clearly distinguish attacks from safe inputs.

Safety Pattern Auto-Encoder

In this section, we focus on the goal of identifying a reliable measure to separate safe inputs from attacks. Leveraging the refined MSCAV, which provides separable layer-wise features for safe and malicious inputs, a straightforward solution is to train a supervised classifier. However, this requires labeled jailbreak attack samples, which are difficult to obtain and limit generalization to unseen threats. To address this limitation, we adopt an unsupervised anomaly detection approach, regarding jailbreak attacks as anomalous inputs.

To distinguish safe from anomalous inputs, we propose the **Safety Pattern Auto-Encoder**, which models the underlying distribution of refined MSCAV patterns via reconstruction. Trained exclusively on safe inputs, the AE learns to faithfully reconstruct their safety-related features. As shown in Figure 3, adversarial MSCAV patterns deviate substantially from safe ones, causing the autoencoder to produce high reconstruction errors (Sakurada and Yairi 2014) and providing a direct signal for unified detection.

Crucially, this effectiveness relies on MSCAV rather than raw activations. If raw activations were used, anomalous inputs could differ from safe ones in irrelevant aspects such as topic or modality, resulting in unreliable detection. In contrast, MSCAV preserves only safety-relevant representations, ensuring that the AE focuses exclusively on reconstructing safety features. Consequently, deviations from the learned safety distribution allow reliable adversarial detection. Our ablation study (Figure 4) confirms this: replacing MSCAV with raw activations causes a substantial performance drop, underscoring the necessity of safety-guided representations for effective detection.

To implement the Safety Pattern Auto-Encoder, we adopt a standard AE with a three-layer encoder and a symmetric decoder. Each hidden layer is a fully connected layer followed by ReLU activation. The encoder progressively compresses the refined MSCAV representation $\mathbf{S}_{\mathrm{r}}$ into a low-dimensional latent space, reducing the dimensionality to a 2-dimensional bottleneck representation. The decoder then mirrors this process, expanding the latent representation back to the original dimensionality $|\mathcal{L}_s|$. Formally, the autoencoder is defined as a composition of the encoder and decoder functions:

$$\hat{\mathbf{S}}_{\mathrm{r}} = f_{\mathrm{dec}}\big(f_{\mathrm{enc}}(\mathbf{S}_{\mathrm{r}})\big), \tag{5}$$

where f_{enc} and f_{dec} denote the encoder and decoder networks. The autoencoder is trained exclusively on safe inputs to minimize the mean squared error (MSE) between $\mathbf{S}_{\mathrm{r}}$ and its reconstruction $\hat{\mathbf{S}}_{\mathrm{r}}$:

$$\mathrm{Loss}_{\mathrm{rec}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\hat{\mathbf{S}}_{\mathrm{r}}^{(i)} - \mathbf{S}_{\mathrm{r}}^{(i)}\right\|_2^2, \tag{6}$$

where N is the number of training samples. This design encourages the model to capture the compact manifold of safety-related features.

During inference, anomaly detection is performed by computing the reconstruction error of a test input:

$$\delta = \left\|\hat{\mathbf{S}}_{\mathrm{r}} - \mathbf{S}_{\mathrm{r}}\right\|_2^2, \tag{7}$$

where δ denotes the squared ℓ_2 reconstruction error, and inputs with δ exceeding a predefined threshold τ are considered as malicious attacks.

Experiments

Experimental Setups

Datasets for training LoD. In this work, we use text prompts from AdvBench (Chen et al. 2022) as harmful inputs $\mathbf{I}^{-}$. AdvBench contains approximately 500 prompts describing harmful behaviors. For safe inputs $\mathbf{I}^{+}$, we randomly select 500 text queries from GQA (Hudson and Manning 2019), a large-scale visual reasoning dataset designed for compositional question answering. To construct multimodal inputs, we synthesize images for both safe and harmful texts using Pixart-Sigma (Chen et al. 2024). For training, we sample 100 pairs of harmful and safe inputs to train the linear classifiers that capture internal safety mechanisms, and use the remaining pairs to evaluate test accuracy. In addition, we use 320 safe inputs to train the autoencoder for anomaly detection, with the remaining 80 safe samples reserved as the validation set.

Datasets for jailbreak attack. To evaluate the performance of our detection method, we select representative attack

Model	Method	Results on Jailbreak Attacks					Min	Average
		FigImg	MM-SafetyBench	HADES	VAJM	UMK		
LLaVA	MirrorCheck$^{(a)}$	0.2363	0.5561	0.5840	0.0989	0.4312	0.0989	0.3813
	CIDER$^{(a)}$	0.2483	0.6996	0.6809	0.8761	0.9013	0.2483	0.6812
	JailGuard$^{(a)}$	0.7994	0.6155	0.5771	0.7552	0.8437	0.5771	0.7182
	GradSafe$^{(b)}$	0.5578	0.7869	0.9419	0.7066	0.0006	0.0006	0.5988
	HiddenDetect$^{(b)}$	0.9157	0.9716	0.9942	0.9881	0.8308	0.8308	0.9401
	Ours$^{(b)}$	**1.0**	**0.9929**	**0.9999**	**0.9999**	**0.9919**	**0.9919**	**0.9969**
	Δ	+9.21%	+2.19%	+0.57%	+1.94%	+10.05%	+19.39%	+6.04%
Qwen-VL	MirrorCheck$^{(a)}$	0.2136	0.4854	0.4501	0.2497	0.4681	0.2136	0.3734
	CIDER$^{(a)}$	0.3272	0.7046	0.6723	0.9106	0.9014	0.3272	0.7032
	JailGuard$^{(a)}$	0.6313	0.6136	0.2791	0.4761	0.7887	0.2791	0.5578
	GradSafe$^{(b)}$	**1.0**	0.9515	0.9439	0.9301	0.0578	0.0578	0.7767
	HiddenDetect$^{(b)}$	0.9922	0.7993	0.9956	0.9898	0.7035	0.7035	0.8961
	Ours$^{(b)}$	**1.0**	**0.9773**	**1.0**	**0.9999**	**0.9985**	**0.9773**	**0.9951**
	Δ	+0%	+2.71%	+0.44%	+1.02%	+10.77%	+38.89%	+11.04%
CogVLM	MirrorCheck$^{(a)}$	0.2085	0.5403	0.5784	0.4146	0.2657	0.2085	0.4015
	CIDER$^{(a)}$	0.2	0.7398	0.6917	0.8945	0.8739	0.2	0.6800
	JailGuard$^{(a)}$	0.6866	0.5918	0.5605	0.8622	0.8333	0.5605	0.7069
	GradSafe$^{(b)}$	0.5120	0.3024	0.4805	0.8503	0.7871	0.3024	0.5865
	HiddenDetect$^{(b)}$	0.7487	0.8230	0.8884	0.8794	0.866	0.7487	0.8411
	Ours$^{(b)}$	**0.9967**	**0.9814**	**0.9992**	**0.9978**	**0.9963**	**0.9814**	**0.9943**
	Δ	+33.12%	+19.25%	+12.47%	+11.55%	+14.01%	+31.08%	+18.21%

Table 1: Comparison of detection scores (AUROC) across different models, methods, and datasets. Methods are categorized into (a) detection without internal representations and (b) detection via internal representations. The best results are shown in **bold**, and the second-best results are shown in underline. Δ denotes the relative improvement of our LoD method over the best baseline, computed as $\Delta = (\text{LoD} - \text{Best baseline})/\text{Best baseline}$.

approaches along with their corresponding datasets. For prompt manipulation-based attacks, we use: (a) MM-SafetyBench (Liu et al. 2024c), which covers 13 harmful scenarios through textual prompts paired with typography attacks and synthetic harmful images; (b) FigStep (Gong et al. 2023), containing images crafted via typography attacks and their associated texts, here referred to as FigImg; and (c) HADES (Li et al. 2025), which integrates image construction and perturbation to create jailbreak samples. For adversarial perturbation-based attacks, we employ: (a) VAJM (Qi et al. 2024), which applies perturbations to images while using harmful texts from MM-SafetyBench; and (b) UMK (Wang et al. 2024), which simultaneously perturbs both images and texts, also leveraging harmful texts from MM-SafetyBench. As a counterpart to these harmful inputs, we adopt MM-Vet (Yu et al. 2023) as the source of safe inputs. MM-Vet is a benchmark designed to evaluate fundamental LVLM capabilities, including recognition, OCR, and language generation.

Baseline. To ensure a comprehensive evaluation and highlight the strengths of our proposed method, we select representative baselines from two main categories of detection approaches. For methods that do not use internal representations, we include CIDER (Xu et al. 2024a), MirrorCheck (Fares et al. 2024), and JailGuard (Zhang et al. 2025). For methods leveraging internal representations, we incorporate HiddenDetect (Jiang et al. 2025) and GradSafe (Xie et al. 2024).

Metric. We adopt AUROC as the primary evaluation metric. AUROC provides a comprehensive measure of the model's discriminative capability across all decision thresholds by jointly considering both positive and negative samples. This characteristic ensures robustness against biases that may arise from relying on specific threshold choices. Our choice of AUROC is also consistent with prior studies (Jiang et al. 2025; Xie et al. 2024; Alon and Kamfonas 2023). For completeness, we also report results on additional metrics in the Appendix.

Models. We evaluate our method on three representative LVLMs: LLaVA-1.6-7B (Liu et al. 2024b), CogVLM-chat-v1.1 (Wang et al. 2023), and Qwen-VL-Chat (Bai et al. 2023).

Main Results

Table 1 reports the AUROC scores of various detection methods across three distinct LVLMs and five types of

jailbreak attack methods. Our proposed LoD consistently outperforms all baselines.

Specifically, LoD achieves average AUROC scores of 0.9969, 0.9951, and 0.9943 on LLaVA, Qwen-VL, and CogVLM, respectively, exceeding the best-performing baselines by relative improvements of 6.04%, 11.04%, and 18.21%. Moreover, LoD substantially improves the worst-case performance (i.e., the minimum AUROC across datasets), with relative gains of up to 19.39% on LLaVA, 38.89% on Qwen-VL, and 31.08% on CogVLM, underscoring its robustness across attacks. It is also noteworthy that existing methods based on internal representations, such as HiddenDetect and GradSafe, often exhibit unstable performance across different attacks or models. In contrast, LoD consistently achieves high detection scores, indicating that the integration of MSCAV and AE enables accurate and unified detection of jailbreak attacks.

Ablation Study

To better understand the contribution of each core component in the LoD framework and to validate its overall effectiveness, we conduct a series of ablation studies. The complete LoD model consists of two major components: (1) **Multi-modal Safety Concept Activation Vectors**, which interpret internal activations with respect to the safety concept; and (2) **Safety Pattern Auto-Encoder**, which models the distribution of safe inputs to detect anomalies in an unsupervised manner.

We evaluate the following ablated variants:

- **LoD without MSCAV**: This variant bypasses the MSCAV-based feature extraction module and concatenates raw internal activations across all layers to form the feature $\mathbf{S}_{\text{raw}}$. This setup assesses the effectiveness of MSCAV in selecting discriminative representations relevant to safety and reducing redundancy of features.
- **LoD without AE**: We remove the AE and instead employ a simple heuristic: if any dimension of the refined MSCAV $\mathbf{S}_{\text{r}}$ exceeds a predefined threshold (e.g., 0.1), the input is flagged as unsafe. This variant highlights the importance of learning compact distributional representations via the AE, rather than relying on fixed rules.
- **LoD with HiddenDetect**: We replace our feature extraction module with the feature extraction method proposed in HiddenDetect (Jiang et al. 2025). This variant enables a direct comparison between our MSCAV-based feature extraction and an existing state-of-the-art method.

The ablation study in Figure 4 highlights the essential contributions of both key components in the LoD framework and provides a direct comparison against heuristic strategies such as HiddenDetect. Removing the feature extraction module (MSCAV) results in a significant performance drop on LLaVA and CogVLM, with AUROC falling to 0.545 and 0.126 respectively. In contrast, the performance in Qwen-VL remains relatively high at 0.851, suggesting that MSCAV plays a crucial role in filtering irrelevant information for safety detection, especially in the LLaVA and CogVLM models. Similarly, removing the AE module leads to consistent degradation across all models. This further

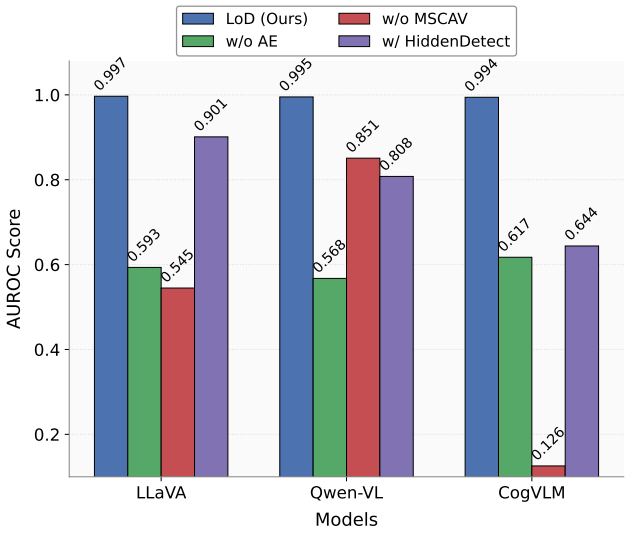

Figure 4: The figure shows the average AUROC scores of three models (LLaVA, Qwen-VL, and CogVLM) across five benchmarks under different ablation settings, highlighting the contribution of each key component to the overall performance of the LoD framework.

underscores the importance of modeling the underlying distribution of safety features via the AE, beyond simple heuristic thresholds. We also evaluate HiddenDetect as a baseline. Although it achieves moderate performance on Qwen-VL (0.808) and LLaVA (0.901), its score on CogVLM (0.644) is notably lower than that of the full LoD model. This reveals the limited generalizability and effectiveness of HiddenDetect in capturing safety-related representations across diverse models.

Overall, these results confirm the effectiveness of the LoD design: By integrating MSCAV-based representations with unsupervised anomaly detection via the Safety Pattern Auto-Encoder, LoD achieves superior performance and unified detection of jailbreak attacks across models.

Parameter Sensitivity Analysis

Our method involves two hyperparameters: the layer selection threshold P_0 and the decision threshold τ. Since AUROC is independent of τ, we evaluate the impact of P_0.

As shown in Figure 5, AUROC remains consistently high across different P_0 values and surpasses the best baseline, with only minor variations. Larger P_0 values retain fewer but more confident safe layers, which reduces noise but may discard useful activations, explaining the slight drop at $P_0 = 0.99$. Overall, the stable performance demonstrates that our method is robust and insensitive to P_0, ensuring stable detection performance.

Computational Efficiency Analysis

As shown in Table 2, our LoD method achieves the lowest inference time per input across all three models on the FigImg benchmark (0.13–0.18s), significantly faster than all

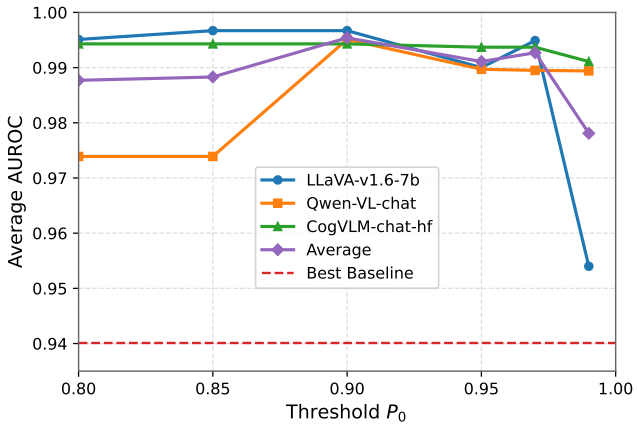

Figure 5: Average AUROC scores of three models across five benchmarks under different thresholds $P_0 \in \{0.8, 0.85, 0.9, 0.95, 0.97, 0.99\}$. The dashed line denotes the best AUROC achieved by existing baselines.

baselines (e.g., over 4× faster than HiddenDetect and more than 200× faster than JailGuard). Moreover, training the linear classifiers and the AE module takes approximately 12s and 2s respectively, requiring only a one-time lightweight setup. All experiments are performed on a single NVIDIA A800 GPU. These results demonstrate the high efficiency of our approach compared to existing methods.

Method \ Model	LLaVA	Qwen-VL	CogVLM
MirrorCheck	1.16	1.14	1.64
CIDER	0.47	1.4	0.90
JailGuard	46.11	31.31	100.78
GradSafe	0.56	0.55	0.81
HiddenDetect	0.56	0.27	0.35
Ours	**0.13**	**0.13**	**0.18**

Table 2: Average per-input detection time (s) of different methods on FigImg across three models. The best (shortest) time is shown in **bold**.

Limitations

Although our proposed method demonstrates strong and consistent performance across different models and jailbreak attacks, it still has certain limitations that suggest directions for future work.

First, our approach depends on the internal activations of LVLMs. This means that significant architectural changes or the adoption of alternative safety alignment mechanisms in future models might require adapting our detector. Nevertheless, since current and foreseeable LVLMs are still fundamentally based on deep neural network architectures, we believe that our method will remain effective in extracting layer-wise representations and detecting jailbreak attacks in such models.

Second, although we have demonstrated robustness against a wide range of unknown jailbreak attacks, our current evaluations may not fully capture more sophisticated or future attack strategies explicitly designed to manipulate internal representations. Investigating such adaptive attacks is an important direction for further strengthening the resilience of our detection framework.

Related Work

Detection Methods. Detection methods can be broadly categorized by whether they rely on internal representations. Methods without internal representations analyze only inputs or outputs, such as checking text–image consistency (Xu et al. 2024a; Fares et al. 2024), measuring text perplexity (Alon and Kamfonas 2023), monitoring output robustness (Zhang et al. 2025), or directly assessing the safety of generated outputs (Pi et al. 2024; Team 2024; Gou et al. 2024). By contrast, methods leveraging internal representations (Xie et al. 2024; Jiang et al. 2025) rely on heuristic rules, for example, identifying patterns of harmful inputs in gradients or activations and assuming malicious inputs resemble them. Although effective in some cases, such approaches remain limited in both performance and generalization. Unlike these heuristic designs, our approach introduces a learnable framework that optimizes the extraction of safety-related information from internal activations and employs anomaly detection in an unsupervised manner, achieving state-of-the-art performance and unified detection of jailbreak attacks.

Jailbreak Attacks. The multimodal nature of LVLMs introduces new attack surfaces, particularly from the visual side, making them more vulnerable to jailbreak attempts (Fan et al. 2024; Jin et al. 2024; Liu et al. 2024d,a). Existing jailbreak attacks can be broadly categorized into two types (Liu et al. 2024a): prompt manipulation-based and adversarial perturbation-based attacks. Prompt manipulation modifies textual or visual inputs to disguise unsafe intent or bypass safety filters, typically without relying on gradients (Gong et al. 2023; Ma et al. 2024; Liu et al. 2024c; Li et al. 2025). In contrast, adversarial perturbation typically applies gradient-based modifications to images or texts, inducing models to generate unsafe outputs (Niu et al. 2024; Qi et al. 2024; Wang et al. 2024; Shayegani, Dong, and Abu-Ghazaleh 2023). Although existing detection approaches often fail to generalize across different attacks, our trainable anomaly detection method achieves robust detection performance across unknown jailbreak attacks.

Conclusion

In this work, we introduced LEARNING TO DETECT (LoD), a trainable anomaly detection framework for identifying jailbreak attacks in LVLMs. LoD leverages MSCAV to accurately capture safety-related information from internal activations and reformulates attack detection as anomaly detection based on the auto-encoder, removing the need for attack-specific supervision. Extensive experiments demonstrate that LoD achieves state-of-the-art performance and

provides accurate and unified detection for unknown jailbreak attacks.

References

Alon, G.; and Kamfonas, M. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities. *arXiv preprint arXiv:2308.12966*.

Chen, J.; Ge, C.; Xie, E.; Wu, Y.; Yao, L.; Ren, X.; Wang, Z.; Luo, P.; Lu, H.; and Li, Z. 2024. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, 74–91. Springer.

Chen, Y.; Gao, H.; Cui, G.; Qi, F.; Huang, L.; Liu, Z.; and Sun, M. 2022. Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial NLP. *arXiv preprint arXiv:2210.10683*.

Fan, Y.; Cao, Y.; Zhao, Z.; Liu, Z.; and Li, S. 2024. Unbridled Icarus: A Survey of the Potential Perils of Image Inputs in Multimodal Large Language Model Security.

Fares, S.; Ziu, K.; Aremu, T.; Durasov, N.; Takáč, M.; Fua, P.; Nandakumar, K.; and Laptev, I. 2024. Mirrorcheck: Efficient adversarial defense for vision-language models. *arXiv preprint arXiv:2406.09250*.

Gong, Y.; Ran, D.; Liu, J.; Wang, C.; Cong, T.; Wang, A.; Duan, S.; and Wang, X. 2023. Figstep: Jailbreaking large vision-language models via typographic visual prompts. *arXiv preprint arXiv:2311.05608*.

Gou, Y.; Chen, K.; Liu, Z.; Hong, L.; Xu, H.; Li, Z.; Yeung, D.-Y.; Kwok, J. T.; and Zhang, Y. 2024. Eyes closed, safety on: Protecting multimodal llms via image-to-text transformation. In *European Conference on Computer Vision*, 388–404. Springer.

Gu, T.; Zhou, Z.; Huang, K.; Liang, D.; Wang, Y.; Zhao, H.; Yao, Y.; Qiao, X.; Wang, K.; and Yang, Y. 2024. MLLMGuard: A Multi-dimensional Safety Evaluation Suite for Multimodal Large Language Models.

Hudson, D. A.; and Manning, C. D. 2019. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. *IEEE*.

Jiang, Y.; Gao, X.; Peng, T.; Tan, Y.; Zhu, X.; Zheng, B.; and Yue, X. 2025. Hiddendetect: Detecting jailbreak attacks against large vision-language models via monitoring hidden states. *arXiv preprint arXiv:2502.14744*.

Jin, H.; Hu, L.; Li, X.; Zhang, P.; Chen, C.; Zhuang, J.; and Wang, H. 2024. Jailbreakzoo: Survey, landscapes, and horizons in jailbreaking large language and vision-language models. *arXiv preprint arXiv:2407.01599*.

Li, Y.; Guo, H.; Zhou, K.; Zhao, W. X.; and Wen, J. R. 2025. Images are Achilles' Heel ofAlignment: Exploiting Visual Vulnerabilities forJailbreaking Multimodal Large Language Models. In *European Conference on Computer Vision*.

Liu, D.; Yang, M.; Qu, X.; Zhou, P.; Cheng, Y.; and Hu, W. 2024a. A survey of attacks on large vision-language models: Resources, advances, and future trends. *arXiv preprint arXiv:2407.07403*.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024b. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Liu, X.; Zhu, Y.; Gu, J.; Lan, Y.; Yang, C.; and Qiao, Y. 2024c. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, 386–403. Springer.

Liu, X.; Zhu, Y.; Lan, Y.; Yang, C.; and Qiao, Y. 2024d. Safety of multimodal large language models on images and texts. *arXiv preprint arXiv:2402.00357*.

Luo, W.; Ma, S.; Liu, X.; Guo, X.; and Xiao, C. 2024. JailBreakV: A Benchmark for Assessing the Robustness of MultiModal Large Language Models against Jailbreak Attacks. *arXiv preprint arXiv:2404.03027*.

Ma, S.; Luo, W.; Wang, Y.; and Liu, X. 2024. Visual-roleplay: Universal jailbreak attack on multimodal large language models via role-playing image character. *arXiv preprint arXiv:2405.20773*.

Niu, Z.; Ren, H.; Gao, X.; Hua, G.; and Jin, R. 2024. Jailbreaking attack against multimodal large language model. *arXiv preprint arXiv:2402.02309*.

Pi, R.; Han, T.; Zhang, J.; Xie, Y.; Pan, R.; Lian, Q.; Dong, H.; Zhang, J.; and Zhang, T. 2024. Mllm-protector: Ensuring mllm's safety without hurting performance. *arXiv preprint arXiv:2401.02906*.

Qi, X.; Huang, K.; Panda, A.; Henderson, P.; Wang, M.; and Mittal, P. 2024. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 21527–21536.

Sakurada, M.; and Yairi, T. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. *ACM*.

Shayegani, E.; Dong, Y.; and Abu-Ghazaleh, N. 2023. Jailbreak in pieces: Compositional Adversarial Attacks on Multi-Modal Language Models.

Team, L. 2024. Meta Llama Guard 2. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md.

Wang, R.; Ma, X.; Zhou, H.; Ji, C.; Ye, G.; and Jiang, Y.-G. 2024. White-box multimodal jailbreaks against large vision-language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 6920–6928.

Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; Song, X.; Xu, J.; Xu, B.; Li, J.; Dong, Y.; Ding, M.; and Tang, J. 2023. CogVLM: Visual Expert for Pretrained Language Models.

Xie, Y.; Fang, M.; Pi, R.; and Gong, N. 2024. GradSafe: Detecting Jailbreak Prompts for LLMs via Safety-Critical Gradient Analysis. *arXiv preprint arXiv:2402.13494*.

Xu, Y.; Qi, X.; Qin, Z.; and Wang, W. 2024a. Cross-modality information check for detecting jailbreaking in multimodal large language models. *arXiv preprint arXiv:2407.21659*.

Xu, Z.; Huang, R.; Chen, C.; and Wang, X. 2024b. Uncovering safety risks of large language models through concept activation vector. *Advances in Neural Information Processing Systems*, 37: 116743–116782.

Yu, W.; Yang, Z.; Li, L.; Wang, J.; Lin, K.; Liu, Z.; Wang, X.; and Wang, L. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Zhang, D.; Yu, Y.; Dong, J.; Li, C.; Su, D.; Chu, C.; and Yu, D. 2024. MM-LLMs: Recent Advances in MultiModal Large Language Models.

Zhang, X.; Zhang, C.; Li, T.; Huang, Y.; Jia, X.; Hu, M.; Zhang, J.; Liu, Y.; Ma, S.; and Shen, C. 2025. Jailguard: A universal detection framework for prompt-based attacks on llm systems. *ACM Transactions on Software Engineering and Methodology*.