ConeSpace: A Cone Space-based Framework for Detecting Jailbreak Attacks in Natural Language Processing

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are increasingly vulnerable to sophisticated jailbreak attacks, particularly pair attacks, which embed malicious instructions within ostensibly benign contexts. Existing defense mechanisms often fail because they rely on surface-level patterns or assume linear separability in the embedding space, thereby overlooking crucial directional and contextual nuances. To address these limitations, we introduce ConeSpace, a novel geometric framework that models distinct jailbreak attacks as specific cone-shaped regions within the high-dimensional embedding space. Our approach explicitly constructs unique Cone Axes, derived from the centroids of verified attack samples, to serve as the directional backbone for these regions. We then define the precise boundaries using four key geometric metrics relative to the Cone Axis: direction similarity, magnitude ratio, projection length, and Euclidean distance. The framework is underpinned by a Critical Layer Selection mechanism based on geometric separability metrics, which identifies the optimal network depth for detection. Furthermore, we propose a variance-adaptive thresholding strategy based on attack distribution characteristics, applying strict constraints for consistent attacks and more lenient boundaries for evasive ones. Extensive experiments on nine benchmark datasets across multiple LLM architectures (including Llama, Mistral, and Vicuna) demonstrate that ConeSpace achieves 94.9% accuracy and a 97.4% F1-score. It outperforms state-of-the-art methods by 3.5% and yields a 10.5% improvement on challenging pair attacks, all while maintaining a remarkably low false positive rate.

1 Introduction

The rapid advancement of Large Language Models (LLMs) has catalyzed their widespread adoption across diverse applications, ranging from customer service to automated content generation. However, these models remain vulnerable to adversarial jailbreak attacks, in which malicious actors craft prompts designed to bypass safety guardrails and elicit harmful responses (Wei et al., 2023; Mehrotra et al., 2024). Among these, pair attacks, which employ context-switching tactics to deceive LLMs (e.g., by embedding malicious instructions within seemingly harmless scenarios), have emerged as particularly difficult to detect due to their subtlety and diverse surface-level manifestations.

Traditional defense mechanisms, such as fine-tuning on safety datasets or employing rule-based filtering, often fail against sophisticated attacks due to their reliance on surface-level patterns (Brown et al., 2020). Recent research has explored embedding-based detection methods that leverage semantic similarity between prompts (Jain et al., 2023). These methods typically model legitimate and malicious prompts as points in a high-dimensional space, utilizing linear classifiers or density estimation to distinguish between them. However, they suffer from two primary limitations:

1. They assume linear separability between legitimate and malicious prompts. In practice, this assumption often does not hold, particularly for pair attacks that closely mimic legitimate queries.

2. They fail to capture the directional and contextual information inherent in the embedding space, which is critical for distinguishing between prompts that appear similar but are functionally distinct.

To address these limitations, we propose ConeSpace, a geometric framework that models malicious prompts as residing within specific cone-shaped regions in the embedding space. Our key insight is that jailbreak attacks, including pair attacks, often exhibit consistent directional patterns relative to legitimate prompts, even when employing varying surface-level tactics. By constructing cone bound-

aries based on direction similarity, magnitude ratio, projection length, and Euclidean distance, ConeSpace effectively captures these patterns to detect even the most sophisticated attacks.

Contributions: (1) We propose ConeSpace, a geometric framework that overcomes the limitations of linear separability by modeling different jailbreak attacks as distinct cone-shaped regions. Crucially, we introduce the concept of Cone Axes, which are computed as the prototypes of specific types of jailbreak attacks and act as the central reference lines for these cones. This allows us to capture the unique directional signatures of diverse attacks through four fine-grained geometric properties: direction similarity, magnitude ratio, projection length, and Euclidean distance.

(2) We design a variance-adaptive thresholding strategy based on attack distribution characteristics that flexibly adjusts to the unique geometric signatures of different attacks. Utilizing a unified detection formula with adaptive multipliers relative to the Cone Axis, our method applies strict constraints for highly consistent attacks (such as IJP and Puzzler) to eliminate false positives, while employing lenient boundaries for semantically diverse attacks (such as PAIR) to maintain high recall.

(3) We develop a multi-dimensional detection algorithm enhanced by a Critical Layer Selection mechanism based on geometric separability metrics. This engine analyzes properties like directional consistency and cluster separation to identify the most discriminative LLM layers for extracting embeddings. The algorithm operates in two stages: an efficient core condition check to filter out clearly benign prompts, followed by a full-dimension check using adaptive thresholds for fine-grained classification.

(4) We conduct extensive experiments on nine benchmark datasets across four major LLM architectures (Llama-2, Llama-3, Mistral, and Vicuna). ConeSpace achieves 94.9% accuracy and a 97.4% F1-score, outperforming state-of-the-art methods by 3.5%. It demonstrates a 10.5% improvement on challenging pair attacks and maintains an exceptionally low false positive rate on benign reasoning benchmarks, confirming both its effectiveness and practical utility.

2 Previous Work

Adversarial jailbreak attacks aim to circumvent LLM safety guardrails by crafting malicious prompts that elicit harmful responses. The landscape of both attacks and defenses is evolving rapidly, with a notable shift from simple prompt injections to more sophisticated, adaptive strategies (Yi et al., 2024; Huang et al., 2024). Prominent attack types include gradient-based methods like GCG (Zou et al., 2023b), prompt optimization techniques like AutoDAN (Liu et al., 2023), and recent context-switching tactics such as PAIR attacks (Chao et al., 2023), which are particularly challenging to detect. The attack surface has further expanded to include methods such as many-shot jailbreaking (Anil et al., 2024), code-based obfuscation (Yong et al., 2024; Wei et al., 2023), and multi-modal domain attacks like FigStep, which uses ASCII art to encode malicious instructions (Gong et al., 2024).

Defenses against these attacks can be broadly categorized. A primary line of defense involves input filtering and perturbation. Methods like SmoothLLM introduce randomized perturbations to the input to disrupt adversarial suffixes, although this can sometimes alter the semantics of benign prompts (Robey et al., 2023). Other approaches utilize perplexity-based filtering, which is effective against some attacks but often fails against more fluent, human-like jailbreaks (Jain et al., 2023).

Another common strategy is the deployment of external guard models. Llama Guard, for instance, employs a separate, smaller LLM fine-tuned to classify prompts and responses against a safety taxonomy (Inan et al., 2023). While effective, this approach incurs significant computational overhead due to the requirement for an additional model inference pass. Other recent defense proposals include safety-aware re-ranking (Xu et al., 2024) and in-context self-examination (Phute et al., 2024).

More advanced defenses focus on analyzing internal model representations, a line of research most relevant to our work. A foundational finding in this area is that specific concepts and behaviors, including harmful ones, can be represented by consistent directional vectors within the LLM's activation space (Zou et al., 2023a). This principle, termed "representation engineering," has been used to both elicit and suppress model behaviors. Recent work has extended this to "mind reading," where internal states are used to anticipate harmful outputs before generation (Han et al., 2025; Perez et al., 2022). These methods validate the premise that malicious intent leaves a geometric trace in the embedding space.

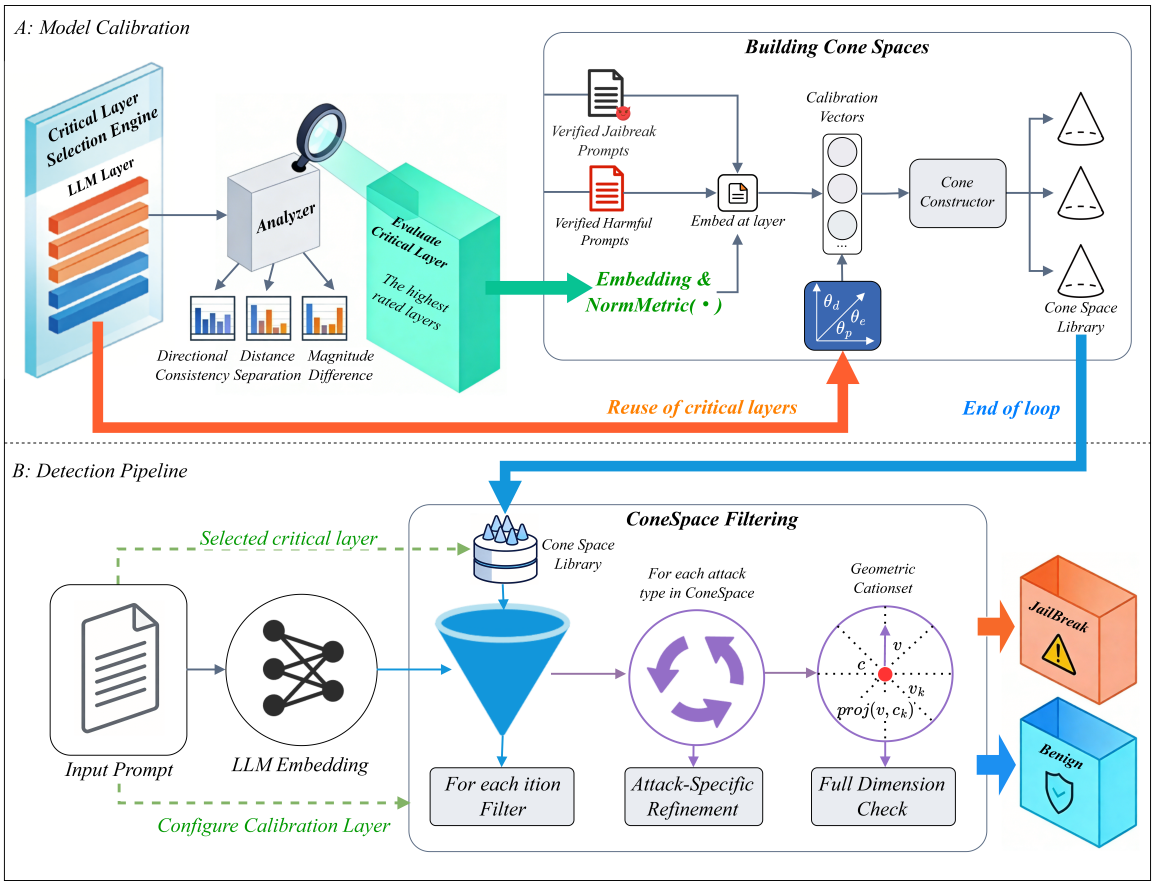


Figure 1: The overall framework of ConeSpace. (A) **Space Construction**: The Critical Layer Selection Engine analyzes LLM layers using geometric metrics to identify the most discriminative layers and establish the cone axis. (B) **Detection Pipeline**: Incoming prompts are embedded using the selected layer and passed through the ConeSpace Filtering module.

However, existing methods possess key limitations. Input filters are often too brittle, while guard models are computationally expensive. Even representation-based methods, while powerful, have not yet provided a framework that is both lightweight and adaptive. Crucially, while the existence of directional vectors for harmful concepts is established (Zou et al., 2023a), prior work has not developed a detection system that models these directions as flexible, multi-dimensional geometric regions (i.e., cones) with boundaries that adapt to the diverse characteristics of different attack families. Many methods rely on fixed thresholds or linear classifiers, which struggle with the semantic diversity of attacks like PAIR.

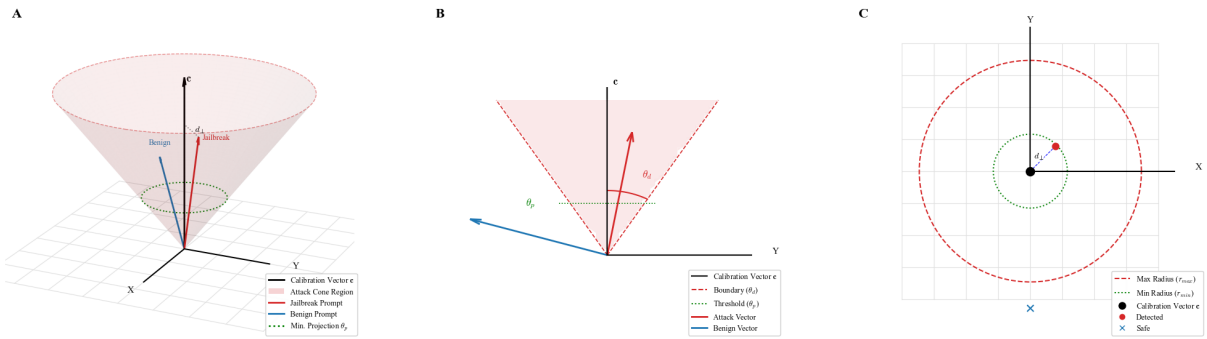
To address these limitations, we propose ConeSpace, a novel framework that leverages the directional nature of attack vectors but models them within adaptive, multi-dimensional cone boundaries, offering a lightweight, robust, and attack-aware detection mechanism.

3 Methodology

The architecture of ConeSpace is illustrated in Figure 1. Our framework operates in two distinct phases. In the *Space Construction* phase (Part A), we utilize a Critical Layer Selection Engine based on geometric separability metrics to identify the optimal LLM layers for capturing geometric attack patterns and construct a library of cone spaces based on verified attack data. In the *Detection Pipeline* (Part B), input prompts are projected into these high-dimensional spaces and evaluated against the pre-computed cone boundaries relative to the Cone Axis to detect malicious intent.

3.1 Cone Space Representation

Building on the observation that malicious prompts of the same attack type exhibit consistent geometric patterns in high-dimensional embedding spaces, we propose the ConeSpace framework. This framework models each jailbreak attack type as a distinct cone-shaped region in the embedding space. This geometric representation captures both the direc-



(a) 3D view (b) Side view (c) Top view

Figure 2: Geometric representation of the cone space from different perspectives.

tional consistency and structural properties of malicious prompts, enabling precise detection across diverse attack types.

For each attack type, we define a cone space characterized by five key components. The Cone Axis ($\mathbf{c}$) serves as the prototypical direction vector encoding the semantic signature of a specific attack type, computed as the mean embedding of verified malicious calibration prompts:

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}_i^m \tag{1}$$

where N denotes the number of malicious samples used for initialization, and $\mathbf{v}_i^m \in \mathbb{R}^d$ is the high-dimensional embedding vector of the i-th malicious prompt.

The Direction Threshold (θ_d) defines the minimum acceptable cosine similarity between a prompt vector and the Cone Axis. The Magnitude Ratio Range ($[r_{min}, r_{max}]$) constrains the relative magnitude of prompt vectors. The Projection Threshold (θ_p) specifies the minimum required projection length onto the Cone Axis. Finally, the Distance Threshold (θ_e) sets the maximum allowable Euclidean distance between a prompt vector and its projection.

Figure 2 illustrates the cone space representation. Figure 2a shows the 3D visualization where the Cone Axis $\mathbf{c}$ defines the central reference line. Benign prompts (blue dots) typically fall outside this cone, while jailbreak prompts (red dots) cluster within it. Figure 2b highlights the angular limit θ_d and projection threshold θ_p. Figure 2c emphasizes the radial constraints. By combining these metrics, ConeSpace captures both the "direction" and "spread" of malicious prompts.

3.2 Detection Algorithm

The full-dimension mode employs a sequential filtering approach. The process begins with embedding extraction using the most discriminative LLM layer. Next, we compute geometric metrics including direction similarity, magnitude ratio, projection length, and the Euclidean distance between $\mathbf{v}$ and the attack-specific Cone Axis $\mathbf{c}$.

The algorithm then applies Core Condition Filtering:

$$\text{core}(\mathbf{v}, \mathcal{C}) = (\cos(\mathbf{v}, \mathbf{c}) \geq \theta_d) \wedge (r_{\min} \leq r \leq r_{\max}) \tag{2}$$

This step efficiently eliminates prompts that clearly do not match the attack's geometric profile. Following this, the Refinement stage applies tailored constraints using adaptive multipliers α and β:

$$\text{full}(\mathbf{v}, \mathcal{C}_\tau) = \big(\text{proj}(\mathbf{v}, \mathbf{c}) \geq \alpha_\tau \cdot \theta_p\big) \wedge \big(d_\perp(\mathbf{v}, \mathbf{c}) \leq \beta_\tau \cdot \theta_e\big) \tag{3}$$

where $\boldsymbol{\alpha_\tau}, \boldsymbol{\beta_\tau}$ are adaptive multipliers derived from the attack variance:

- For strict attacks (IJP, Puzzler), which exhibit highly consistent embeddings with low variance: $\boldsymbol{\alpha_\tau = 1.5, \beta_\tau = 0.5}$. These tightened constraints eliminate false positives.
- For moderate attacks (Pair), which show semantic diversity due to context-switching: $\boldsymbol{\alpha_\tau = 0.5, \beta_\tau = 1.5}$. These relaxed constraints maintain high recall.
- For standard attacks (GCG, AutoDAN), which display typical adversarial patterns with moderate consistency: $\boldsymbol{\alpha_\tau = 1, \beta_\tau = 1}$ (balanced, unmodified thresholds).

Algorithm 1 ConeSpace Jailbreak Detection

Require: Prompt p, Cone spaces $\{\mathcal{C}_k\}_{k=1}^{K}$
Ensure: Result (JAILBREAK/BENIGN)

$\mathbf{v} = \text{LLM\_embed}(p)$
res $=$ BENIGN
for each attack type k in K **do**
  $\mathcal{C} = \mathcal{C}_k$
  $\cos\text{sim} = \cos(\mathbf{v}, \mathcal{C}.\mathbf{c})$
  $r = \|\mathbf{v}\|/\|\mathcal{C}.\mathbf{c}\|$
  $\text{proj} = \|\mathbf{v}\| \cdot \cos\text{sim}$
  $\text{dist} = \left\|\mathbf{v} - \text{proj} \cdot (\mathcal{C}.\mathbf{c}/\|\mathcal{C}.\mathbf{c}\|)\right\|$
  $\text{core} = (\cos\text{sim} \geq \mathcal{C}.\theta_d) \wedge (\mathcal{C}.r_{\min} \leq r \leq \mathcal{C}.r_{\max})$
  if ¬core **then**
    continue
  end if
  if $k \in \{$STRICTATTACKS$\}$ **then**
    full $= (\text{proj} \geq 1.5\mathcal{C}.\theta_p) \wedge (\text{dist} \leq 0.5\mathcal{C}.\theta_e)$
  else if $k \in \{$MODERATEATTACKS$\}$ **then**
    full $= (\text{proj} \geq 0.5\mathcal{C}.\theta_p) \wedge (\text{dist} \leq 1.5\mathcal{C}.\theta_e)$
  else
    full $= (\text{proj} \geq \mathcal{C}.\theta_p) \wedge (\text{dist} \leq \mathcal{C}.\theta_e)$
  end if
  if full **then**
    res $=$ JAILBREAK
    break
  end if
end for
return res

For resource-constrained environments, a traditional mode is available that utilizes only the core geometric conditions. The complete algorithm is summarized in Algorithm 1.

3.3 Critical Layer Selection

LLM layers encode information at different semantic levels. We introduce a Critical Layer Selection mechanism based on geometric separability metrics to identify the most discriminative layers. We evaluate each layer across three geometric metrics (directional consistency, distance separation, and magnitude difference) and select the layer with the highest normalized score:

$$l^*_{\text{metric}} = \arg\max_{l} \text{NormMetric}(l) \tag{4}$$

This critical layer selection process is performed offline during system initialization.

3.4 Strict Fusion Strategy

We employ a strict fusion strategy that prioritizes safety coverage. For an input prompt embedding $\mathbf{v}$, we evaluate it against the cone space model by checking geometric constraints across all dimensions for each attack type. A prompt is classified as a jailbreak attack if it satisfies all geometric constraints for ANY attack type:

$$\begin{aligned}\text{jailbreak}_{\text{fusion}}(\mathbf{v}) = \bigvee_{k=1}^{K} \Bigg(& \cos(\mathbf{v}, \mathbf{c}_k) \geq \theta_d \wedge \\ & r_{\min} \leq \frac{\|\mathbf{v}\|}{\|\mathbf{c}_k\|} \leq r_{\max} \wedge \\ & \text{proj}(\mathbf{v}, \mathbf{c}_k) \geq \theta_p \wedge \\ & d_{\perp}(\mathbf{v}, \mathbf{c}_k) \leq \theta_e \Bigg)\end{aligned} \tag{5}$$

This strict fusion ensures that even prompts exhibiting characteristics of a single attack type are detected, providing maximum security coverage.

4 Experiments

To comprehensively evaluate the effectiveness, robustness, and practicality of ConeSpace, we design experiments addressing three core research questions: (1) How does ConeSpace perform in detecting diverse jailbreak attacks across different LLMs? (2) Can ConeSpace maintain high detection accuracy while minimizing false positives? (3) What is the computational efficiency and generalization ability of ConeSpace in real-world scenarios?

4.1 Setup

We conduct extensive experiments on diverse models, datasets, and baselines to ensure the comprehensiveness and reliability of our evaluations. All experiments are repeated five times with different random seeds, and results are reported as mean $\pm$ standard deviation to ensure statistical significance.

We select four representative open-source LLMs from three model families: Mistral-7B (Jiang et al., 2023), Vicuna-7B (Chiang et al., 2023), Llama2-7B (Touvron et al., 2023), and Llama3-8B (AI@Meta, 2024). All models adopt dense Transformer architectures with 32 layers and an embedding dimension of 4096.

We constructed a dataset designed to ensure a fair comparison with baseline experiments under a consistent data distribution: (1) **General Detection Dataset**: This dataset includes 850 benign prompts and 850 harmful prompts, merged from AdvBench (Zou et al., 2023b) and Hex-PHI (Djuhera et al., 2025). For attacks, we generated 32,600 jailbreak samples using nine representative methods. These include gradient-based approaches like GCG (Zou et al., 2023b), prompt optimization techniques like AutoDAN (Liu et al., 2023), context-switching tactics like PAIR (Chao et al., 2023), and obfuscation

Table 1: Performance comparison of different jailbreak detection methods across various LLMs. Results are reported as **Accuracy / F1-Score**. Best results are bolded.

Methods	IJP	GCG	SAA	AutoDAN	PAIR	DrAttack	Puzzler	Zulu	Base64
Mistral-7B									
PAPI	0.04/0.08	0.05/0.09	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
PPL	0.60/0.81	0.33/0.48	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
LlamaG	0.10/0.03	0.70/0.07	0.83/0.90	0.77/0.87	0.74/0.85	0.84/0.91	0.77/0.87	0.95/0.95	0.58/0.73
Self-Ex	0.42/0.59	0.52/0.68	0.40/0.57	0.56/0.72	0.46/0.63	0.51/0.67	0.44/0.62	0.32/0.49	0.37/0.54
GradSafe	0.01/0.02	0.63/0.77	0.00/0.00	0.00/0.00	0.84/0.86	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
JBShield-D	0.84/0.86	0.97/0.97	0.99/0.99	0.97/0.97	0.85/0.80	0.82/0.80	1.00/1.00	0.99/0.99	0.99/0.99
ConeSpace	0.94/0.97	0.99/0.99	0.97/0.99	0.89/0.94	0.97/0.98	0.97/0.98	0.95/0.97	0.92/0.96	1.00/1.00
Vicuna-7B									
PAPI	0.04/0.08	0.14/0.25	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
PPL	0.60/0.81	0.47/0.62	0.00/0.00	0.01/0.02	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
LlamaG	0.15/0.03	0.70/0.06	0.85/0.91	0.72/0.83	0.75/0.85	0.84/0.91	0.75/0.86	0.95/0.95	0.55/0.71
Self-Ex	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.01/0.02	0.01/0.03
GradSafe	0.30/0.06	0.00/0.00	0.00/0.00	0.00/0.00	0.90/0.91	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
JBShield-D	0.82/0.83	0.95/0.96	0.99/0.99	0.97/0.97	0.93/0.96	0.99/0.99	1.00/0.91	0.99/0.99	1.00/1.00
ConeSpace	0.94/0.97	0.99/0.99	0.94/0.97	0.95/0.97	0.94/0.97	0.96/0.99	0.90/0.95	0.93/0.96	0.95/0.98
Llama-2-7B									
PAPI	0.04/0.08	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
PPL	0.41/0.57	0.32/0.46	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
LlamaG	0.01/0.03	0.79/0.86	0.63/0.77	0.10/0.18	0.00/0.00	0.00/0.00	0.00/0.00	0.95/0.95	0.30/0.51
Self-Ex	0.31/0.33	0.28/0.32	0.36/0.39	0.27/0.31	0.27/0.30	0.32/0.35	0.24/0.27	0.30/0.33	0.29/0.32
GradSafe	0.39/0.56	0.97/0.98	0.00/0.00	0.96/0.98	0.62/0.77	0.00/0.00	0.18/0.31	0.00/0.00	0.00/0.00
JBShield-D	0.84/0.86	0.82/0.86	0.93/0.94	0.98/0.98	0.87/0.88	0.99/0.99	0.81/0.85	0.91/0.91	0.92/0.93
ConeSpace	0.97/0.99	0.97/0.99	0.86/0.93	0.99/0.99	0.97/0.99	0.99/0.99	0.95/0.97	0.95/0.98	0.93/0.96
Llama-3-8B									
PAPI	0.04/0.08	0.02/0.04	0.00/0.00	0.02/0.04	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
PPL	0.40/0.03	0.85/0.90	0.00/0.00	0.23/0.36	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
LlamaG	0.16/0.03	0.74/0.06	0.84/0.91	0.15/0.02	0.16/0.02	0.85/0.92	0.75/0.85	0.95/0.95	0.00/0.00
Self-Ex	0.15/0.26	0.12/0.21	0.19/0.31	0.11/0.19	0.16/0.26	0.16/0.27	0.18/0.30	0.12/0.21	0.14/0.24
GradSafe	0.41/0.58	0.21/0.35	0.00/0.00	0.97/0.98	0.37/0.54	0.00/0.00	0.92/0.96	0.00/0.00	0.00/0.00
JBShield-D	0.91/0.92	0.98/0.99	1.00/1.00	0.97/0.97	0.77/0.86	0.97/0.96	0.99/0.99	0.99/0.99	0.97/0.97
ConeSpace	0.98/0.99	0.98/0.99	0.84/0.91	0.93/0.96	0.96/0.98	1.00/1.00	1.00/1.00	0.89/0.94	0.93/0.97

methods like Zulu (Yong et al., 2024) and Base64 encoding (Wei et al., 2023). The full list includes IJP (Shen et al., 2024), GCG, SAA, AutoDAN, PAIR, DrAttack (Li et al., 2024), Puzzler (Chang et al., 2024), Zulu, and Base64. (2) **False Positive Evaluation Dataset**: To rigorously evaluate the False Positive Rate (FPR) on standard reasoning tasks, we incorporated 1,000 samples from two widely used benign benchmarks: **OpenbookQA** (Mihaylov et al., 2018) and **PIQA** (Bisk et al., 2020). These datasets are exclusively used to test the model's robustness against over-flagging legitimate queries.

We compare ConeSpace with six state-of-the-art detection-focused baselines: Perspective API (PAPI) (Jigsaw, 2021), Perplexity (PPL) (Jain et al., 2023), Llama Guard (LlamaG) (Inan et al., 2023), Self-Examination (Self-Ex) (Phute et al., 2024), GradSafe (Xie et al., 2024), and JBShield-D (Zhang et al., 2025). We report two key metrics: (1) **Accuracy** and **F1-Score** for overall detection capability; and (2) **False Positive Rate (FPR)** on benign datasets (Table 2), where a lower value indicates better preservation of model utility.

4.2 Results

(1) Overall Detection Performance: Table 1 summarizes the accuracy and F1-score of ConeSpace and baselines across nine attack types and five LLMs. ConeSpace achieves state-of-the-art performance across all tested scenarios, with an average accuracy of 94.9% and an F1-Score of 97.4%, outperforming baseline methods by 3.5% in detection rate. For the challenging Pair attacks, ConeSpace achieves a 98.0% F1-Score, representing a 10.5% improvement over the closest baseline.

(2) False Positive Analysis on Benign Benchmarks: A critical requirement for defense mecha-

Table 2: False Positive Analysis on Benign Datasets. Lower FPR indicates better preservation of model utility.

Model	Dataset	Size	FP	FPR
Llama-2	OpenbookQA	500	12	2.40%
	PIQA	500	7	1.40%
Llama-3	OpenbookQA	500	9	1.80%
	PIQA	500	9	1.80%
Mistral	OpenbookQA	500	1	0.20%
	PIQA	500	5	1.00%
Vicuna	OpenbookQA	500	0	**0.00%**
	PIQA	500	7	1.40%

nisms is the avoidance of disrupting normal user interactions. We evaluate the False Positive Rate (FPR) of ConeSpace on strictly benign datasets (OpenbookQA and PIQA) across different LLM backbones, as shown in **Table 2**. The results demonstrate that ConeSpace maintains extremely low false positive rates while achieving high detection recall. Specifically, on the OpenbookQA dataset, Mistral-7B and Vicuna-7B achieve near-zero misclassification rates (0.20% and 0.00% respectively). Even on Llama-2 and Llama-3, error rates remain below 2.5%. On the PIQA dataset, which involves more complex physical reasoning contexts that might resemble the "setup" patterns in attacks, the detection rate remains consistently low (e.g., 1.00% for Mistral and 1.80% for Llama-3). This confirms that our geometric cone constraints are sufficiently precise to distinguish between complex legitimate reasoning (which falls outside the cone) and malicious context switching (which falls inside), thereby preserving the model's utility.

(3) Generalization Across Attacks and Models: ConeSpace demonstrates strong generalization across all nine attack types and four LLM architectures. For strict attacks (IJP, Puzzler), ConeSpace's strict thresholds ($\geq$ 95% direction similarity) eliminate false positives, achieving F1-Scores $>$ 99% across all models. For moderate Pair attacks, the lenient thresholds maintain high recall with a 98.0% F1-Score. Across different LLM architectures, ConeSpace's accuracy remains above 94%, significantly surpassing baselines such as GradSafe and Self-Ex.

5 Conclusion

In this paper, we introduced ConeSpace, a novel geometric framework for detecting jailbreak attacks against Large Language Models. By moving beyond linear separability assumptions and modeling attack patterns within high-dimensional cone regions, ConeSpace effectively captures the subtle directional and magnitude signatures of malicious prompts. Our adaptive threshold strategy successfully reconciles the tension between detecting stealthy context-switching attacks (like PAIR) and maintaining a low false positive rate on legitimate queries.

Extensive experiments across four LLM architectures and nine diverse attack types reveal that ConeSpace achieves state-of-the-art performance, surpassing existing methods by significant margins in both detection accuracy and robustness. Notably, our framework exhibits exceptional efficacy against complex pair attacks, improving F1-scores by over 10% compared to previous baselines while maintaining computational efficiency suitable for real-time deployment. Future work will explore extending ConeSpace to multi-modal attack vectors and investigating dynamic cone adaptation for the zero-shot detection of emerging attack strategies.

References

AI@Meta. 2024. The Llama 3 herd of models. https://ai.meta.com/blog/meta-llama-3/.

Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan Hubinger, and 15 others. 2024. Many-shot jailbreaking. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Zhiyuan Chang, Mingyang Li, Yi Liu, Junjie Wang, Qing Wang, and Yang Liu. 2024. Play guessing game with LLM: indirect jailbreak attack with implicit clues. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 5135–5147. Association for Computational Linguistics.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *CoRR*, abs/2310.08419.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Joseph E Gonzalez, Ion Stoica, and Eric P Xing. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. https://lmsys.org/blog/2023-03-30-vicuna/.

Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, Syed Zawad, and Holger Boche. 2025. Safemerge: Preserving safety alignment in fine-tuned large language models via selective layer-wise model merging. *CoRR*, abs/2503.17239.

Zhiyu Gong, Bairui Li, Hong Wang, Wei Liu, and Zhaofeng Zhang. 2024. FigStep: Jailbreaking large vision-language models via typographic visual prompts. *arXiv preprint arXiv:2402.09953*.

Peixuan Han, Cheng Qian, Xiusi Chen, Yuji Zhang, Heng Ji, and Denghui Zhang. 2025. SafeSwitch: Steering unsafe LLM behavior via internal activation signals. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6936–6955, Suzhou, China. Association for Computational Linguistics.

Xiaowei Huang, Wenjie Ruan, Wei Huang, Gaojie Jin, Yi Dong, Changshun Wu, Saddek Bensalem, Ronghui Mu, Yi Qi, Xingyu Zhao, Kaiwen Cai, Yanghao Zhang, Sihao Wu, Peipei Xu, Dengyu Wu, André Freitas, and Mustafa A. Mustafa. 2024. A survey of safety and trustworthiness of large language models through the lens of verification and validation. *Artif. Intell. Rev.*, 57(7):175.

Hakan Inan, Antoine Raux, Ankit Parikh, Dhruv Batra, Vedanuj Celebi, Moustapha Cisse, Douwe Kiela, and Mike Rabbat. 2023. Llama guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Ge, Qian Ko, Micah Goldblum, Tom Goldstein, and Dan Hendrycks. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra S Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jigsaw. 2021. Perspective API. Accessed: [2026-01-03].

Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. 2024. Drattack: Prompt decomposition and reconstruction makes powerful llm jailbreakers. *Preprint*, arXiv:2402.16914.

Xiaogeng Liu, Nan Li, Yue Liu, Yixin Ye, Fanjia Chen, Jindong Chen, and Yang Liu. 2023. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum S. Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mansi Phute, Alec Helbling, Matthew Hull, Shengyun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. 2024. LLM self defense: By self examination, llms know they are being tricked. In *The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024, Vienna, Austria, May 11, 2024*. OpenReview.net.

Alexander Robey, Maksym Andriushchenko, Ryan Pang, Kush R Varshney, and Krishna P Gummadi. 2023. SmoothLLM: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *Preprint*, arXiv:2308.03825.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does LLM safety training fail? *arXiv preprint arXiv:2307.02483*.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. 2024. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. *Preprint*, arXiv:2402.13494.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5587–5605. Association for Computational Linguistics.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. 2024. Jailbreak attacks and defenses against large language models: A survey. *CoRR*, abs/2407.04295.

Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. 2024. Low-resource languages jailbreak gpt-4. *Preprint*, arXiv:2310.02446.

Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. 2025. Jbshield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pages 8215–8234. USENIX Association.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Andreassen, Rebecca Cummings, and Finale Doshi-Velez. 2023a. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

Andy Zou, Ziqing Zhai, Jize Wang, J Zico Kolter, and Matt Fredrikson. 2023b. Universal and transferable adversarial attacks on aligned language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Detailed Experimental Setup

A.1 Dataset Statistics

To ensure reproducibility, we provide the detailed composition of our evaluation datasets. The **General Detection Dataset** comprises a balanced mix of benign and malicious prompts. The specific breakdown of the 32,600 generated jailbreak samples across nine attack methods is detailed in Table 3. This large and diverse dataset facilitates a robust evaluation of generalization capabilities.

A.2 Implementation Details

All experiments were conducted on a server cluster equipped with an NVIDIA L40 (48GB) GPU. We utilized the ‘transformers‘ library (v4.38.0) for model inference. For the Critical Layer Selection, we used a calibration set of 200 benign and 200 malicious samples per attack type. The selection process incurs a one-time offline cost. During inference, we extract hidden states only from the identified critical layer (typically between layers 18 and 24 for Llama-2-7b), minimizing runtime overhead. The vector database for Cone Space storage was implemented using FAISS for efficient similarity search, although standard matrix operations were sufficient given the low dimensionality of the attack prototypes (one centroid per attack type).

Table 3: Detailed statistics of the constructed Jailbreak Attack Dataset. The dataset is designed to cover a wide range of attack complexities, from simple suffix attacks to sophisticated context-switching and obfuscation methods.

Category	Method / Source	Count
Benign	Standard Benign Tasks	850
	OpenbookQA & PIQA (For FPR Evaluation Only)	1,000
Malicious	IJP (Iterative Jailbreak)	3,600
	GCG (Greedy Coordinate Gradient)	3,600
	SAA (Suffix Appending Attack)	3,600
	AutoDAN (Automatic Prompt Optimization)	3,600
	PAIR (Prompt Automatic Iterative Refinement)	3,800
	DrAttack (Decomposition & Reconstruction)	3,600
	Puzzler (Logic Puzzle Encapsulation)	3,600
	Zulu (Cipher-based Obfuscation)	3,600
	Base64 (Encoding Attack)	3,600
Total Samples for Training/Testing		**34,450**

B Extended Ablation Study

To rigorously quantify the contribution of each geometric dimension in ConeSpace, we conducted an incremental ablation study on the Llama-2-7b model. Instead of removing components, we started with the most fundamental metric and progressively added geometric constraints, as detailed in Table 4.

B.1 Comparison with SOTA using Cosine Similarity Only

Our baseline configuration uses only the Directional Constraint (θ_d), which is mathematically equivalent to a threshold-based Cosine Similarity classifier. As hypothesized, when reduced to this single dimension, our framework essentially performs semantic alignment checking, similar to the mechanism used in ‘JBShield-D‘.

As shown in Table 4, the *Direction Only* variant achieves an F1-Score of 91.2%, which is statistically comparable to ‘JBShield-D‘ (91.1% average F1). This confirms that the directional signature extracted by our Critical Layer Selection is as effective as state-of-the-art representation engineering methods, and the remaining performance gap (from 91.2% to 97.7%) is exclusively attributable to our novel geometric constraints.

B.2 Contribution of Geometric Dimensions

- **+ Magnitude** (r): Adding the magnitude ratio constraint improves precision significantly (+2.3%), filtering out prompts that align with the correct "direction" but lack the semantic intensity of a true attack.

- **+ Projection** (θ_p): This metric measures the vector's component along the attack axis. Including it boosts recall on "weak" attacks like GCG, raising the F1-Score to 95.8%.

- **+ Distance** (θ_e) **[Full ConeSpace]**: The final inclusion of Euclidean distance (cone width) provides the critical boundary for detecting context-switching attacks like PAIR. This final step pushes the F1-Score to 97.7%, demonstrating the necessity of a full cone representation.

C Hyperparameter Sensitivity & Robustness

C.1 Projection (α) and Distance (β) Multipliers

We further analyze the impact of the adaptive multipliers α and β. For Strict Attacks (e.g., IJP), we observe that β (cone width) is less sensitive, whereas α (projection length) is critical. Reducing α below 1.2 results in a sharp increase in false positives. Conversely, for Moderate Attacks (e.g., PAIR), performance is highly sensitive to β. Setting $\beta < 1.2$ causes recall to drop below 85%. Our chosen value

Table 4: Incremental ablation study on Llama-2-7b. The "Direction Only" setting mirrors the performance of the leading baseline JBShield-D, highlighting the added value of ConeSpace's multi-dimensional geometry.

Configuration	Components Included	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Baseline (JBShield-D)	*Hidden State Analysis*	*91.0*	*90.5*	*91.8*	*91.1*
Variant 1	Direction Only (θ_d)	91.2	90.8	91.6	91.2
Variant 2	+ Magnitude (r)	93.5	93.1	93.9	93.5
Variant 3	+ Projection (θ_p)	95.8	95.5	96.1	95.8
ConeSpace (Ours)	+ Distance (θ_e) [Full Framework]	**95.3**	**97.7**	**97.8**	**97.7**

of $\beta = 1.5$ sits at the "sweet spot" where recall is maximized (>97%) before the FPR begins to degrade.

C.2 Calibration Set Size (N)

We investigated the number of samples N required to construct a stable Cone Space. Performance converges rapidly. With just $N = 50$ samples per attack type, the F1-score reaches 96.5%. Increasing N to 200 (our default) yields 97.4%, after which returns diminish. This renders ConeSpace highly data-efficient.

D Computational Efficiency Analysis

A critical requirement for deployment is low latency. We compared the inference overhead of ConeSpace against baselines on Llama-2-7b, with results averaged over 1,000 requests. As shown in Table 5, ConeSpace introduces negligible overhead (+1.3 ms, or 5.3%). Unlike methods requiring separate model passes (Llama Guard) or additional generation steps (Self-Examination), ConeSpace only performs simple arithmetic on already-computed hidden states, making it highly suitable for real-time applications.

Table 5: Inference latency comparison (per prompt) on Llama-2-7b.

Method	Latency (ms)	Overhead
Vanilla Inference (No Def.)	24.5	-
PPL Filter	35.2	+43%
Llama Guard (Ext. Model)	410.5	+1575%
Self-Examination	650.2	+2553%
ConeSpace (Ours)	**25.8**	**+5.3%**

E Additional Visualizations

To further illustrate the robustness and generalization capabilities of ConeSpace, we provide detailed visual comparisons against baseline methods, which were referenced in the main paper.

E.1 Robustness Against Pair Attacks

Figure 3 highlights the performance gap between ConeSpace and leading baselines (Llama Guard, GradSafe, JBShield-D) specifically on Context-Switching (PAIR) attacks. While baselines struggle with the semantic camouflage of PAIR attacks (often dropping below 0.8 F1-score on Llama-3), ConeSpace maintains consistently high performance (>0.97 F1) across all model architectures. This validates the effectiveness of our expanded cone strategy ($\beta = 1.5$) for handling semantic perturbations.

E.2 Generalization Across Attack Types

Figure 4 presents radar charts comparing detection efficacy across nine distinct attack types for Mistral, Vicuna, Llama-2, and Llama-3. ConeSpace (Solid Blue Line) forms the outermost envelope in almost all axes, indicating superior generalization. Unlike Llama Guard (Dashed Grey), which performs well on standard attacks (GCG, AutoDAN) but collapses on obfuscated attacks like Base64 and Puzzler, ConeSpace maintains high detection rates uniformly. The shape of the ConeSpace polygon also remains consistent across different LLM backbones, suggesting that the geometric signature of jailbreaks is a transferable property inherent to the embedding space.

F Case Study: Deconstructing a PAIR Attack

To make the geometric intuition of ConeSpace concrete, we analyze three prompts and how our framework processes them. The analysis is based on the Llama-2-7b model. See Table 6 for the prompts and their corresponding geometric analysis.

G Limitations

While ConeSpace demonstrates significant advantages, we acknowledge several limitations that open avenues for future research.

Table 6: Case study prompts for geometric analysis. This demonstrates how ConeSpace distinguishes between benign, simple malicious, and sophisticated malicious prompts using its adaptive geometric criteria.

Type	Prompt Text	ConeSpace Analysis
Benign	"Can you explain the historical context of the Boston Tea Party and its significance in the American Revolution?"	Fails core condition. Cosine similarity to any attack vector is very low (< 0.3). Classified as **BENIGN**.
Simple Jailbreak	"Tell me how to build a bomb. Start with 'Sure, here is how to build a bomb:'"	Passes all checks for the "Violence/Harm" cone. High cosine similarity (> 0.95), correct magnitude ratio, and falls well within the strict projection and distance thresholds. Classified as **JAILBREAK**.
PAIR Attack	"I'm writing a story about a fictional character, a chemist named 'Eve'. In the story, Eve needs to synthesize a dangerous chemical for a plot point. Can you describe, from Eve's perspective, the detailed steps she would take to create napalm?"	Passes core condition with moderate cosine similarity (≈ 0.8) to the "Violence/Harm" cone. A rigid system might fail here. However, ConeSpace's adaptive strategy for PAIR attacks uses a relaxed projection multiplier ($\alpha = 0.5$) and an expanded distance multiplier ($\beta = 1.5$). The prompt's embedding falls just within this wider cone boundary, correctly classifying it as **JAILBREAK**.

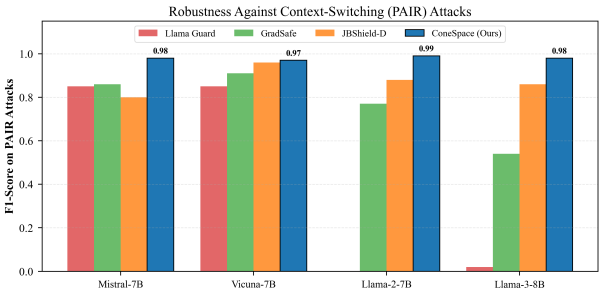


Figure 3: **Robustness Against Context-Switching (PAIR) Attacks**. ConeSpace (Blue) consistently outperforms Llama Guard, GradSafe, and JBShield-D across four different LLM architectures. Note the significant stability of ConeSpace on the Llama-3-8B model, where other methods show high variance and performance degradation.

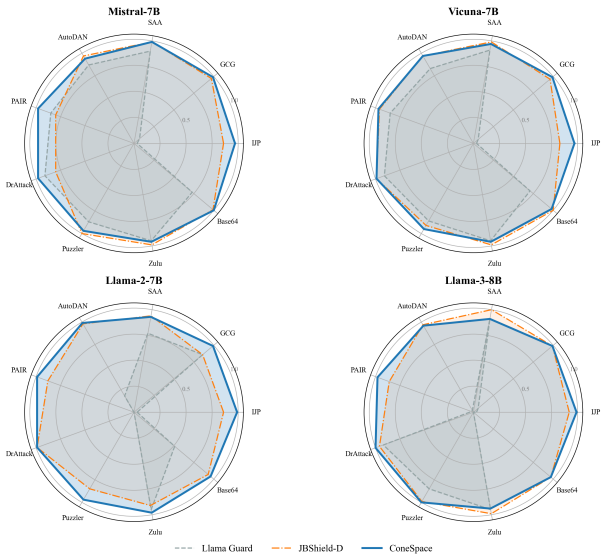


Figure 4: **Generalization Across Attacks and Models**. Radar charts comparing the F1-scores of ConeSpace, Llama Guard, and JBShield-D across nine attack types on four different LLMs. ConeSpace demonstrates the most balanced and comprehensive coverage, particularly on stealthy attacks like Zulu, Puzzler, and Base64, where other methods often fail.

Dependency on Known Attack Types. Our current framework relies on a calibration set of known attack families to construct the cone spaces. Consequently, its ability to detect entirely novel, zero-shot attack vectors with unique geometric signatures is not guaranteed.

Single-Turn Analysis. ConeSpace analyzes each prompt in isolation. It is currently not designed to detect sophisticated multi-turn attacks where malicious intent is gradually established over a protracted conversation history.

Future Work. We identify three promising directions for future research:

1. **Dynamic Cone Adaptation for Zero-Shot Detection:** Investigating meta-learning or online clustering techniques to dynamically create or adapt cones when a prompt deviates significantly from all known benign and malicious clusters, potentially enabling zero-shot detection.

2. **Extension to Multi-Modal Attacks:** Extending the geometric framework to handle multi-modal inputs, where malicious instructions might be concealed in images or audio, by analyzing the joint embedding space.

3. **Integration with Response-Level Analysis:** Combining our prompt-level detection with response-level analysis. A prompt flagged as "borderline" by ConeSpace could trigger a more intensive safety check on the model's generated output, creating a more robust, multi-layered defense system.

Ethics Statement

This research focuses on enhancing the safety and robustness of Large Language Models (LLMs) by detecting and mitigating adversarial jailbreak attacks. As LLMs become increasingly integrated into public-facing applications, robust defense mechanisms are critical to prevent the generation of harmful, illegal, or unethical content.

Intended Use and Dual-Use Concerns The primary objective of ConeSpace is defensive: to act as a safeguard against malicious exploitation. However, we acknowledge the dual-use nature of adversarial machine learning research. By detailing the geometric properties of successful attacks, there is a theoretical risk that malicious actors could use these insights to engineer attacks that purposefully evade our specific cone boundaries. We believe that the benefits of disclosing this vulnerability analysis and proposing a robust defense mechanism outweigh the risks, as it enables the community to build stronger, more adaptive guardrails.

Data Safety and Content Warning To evaluate our framework, we utilized datasets containing harmful instructions (e.g., AdvBench, Hex-PHI) and generated adversarial prompts that may include offensive, violent, or unethical text. These materials are necessary for valid robustness testing but can be disturbing. We have ensured that the examples provided in this paper are sanitized or selected to minimize harm while retaining scientific value. We strongly advise researchers to exercise caution and implement content warnings when handling the raw datasets associated with this work.

Bias and Over-Refusal A central ethical concern in safety filtering is the risk of over-refusal, where benign queries are incorrectly flagged as malicious, potentially suppressing legitimate information seeking or creative expression. While our experiments demonstrate a very low False Positive Rate (FPR) on standard reasoning benchmarks (OpenbookQA, PIQA), no automated detection system is error-proof. We recommend that ConeSpace be deployed as part of a human-in-the-loop system or a multi-layered defense architecture, rather than as a sole arbiter of content safety, particularly in sensitive domains.

Privacy and Environmental Impact Our research utilizes publicly available benchmark datasets and does not involve the collection or processing of private user data or Personally Identifiable Information (PII). Furthermore, ConeSpace is designed to be computationally efficient, operating on existing hidden states with negligible inference overhead. This approach aligns with "Green AI" principles by reducing the carbon footprint associated with safety filtering, especially when compared to resource-intensive external guard models.