SafeSteer: Adaptive Subspace Steering for Efficient Jailbreak Defense in Vision Language Models

Shuchao Pang, Xiyu Zeng, Siyuan Liang†, Liming Lu, Haotian Zhu, Chuanting Zhang, Enguang Liu, Liang Zhang, Basem Shihada, Yongbin Zhou, Jason (Minhui) Xue

***Abstract*—As the capabilities of Vision Language Models (VLMs) continue to improve, they are increasingly targeted by jailbreak attacks. Existing defense methods face two major limitations: (1) they struggle to ensure safety without compromising the model's utility; and (2) many defense mechanisms significantly reduce the model's generation efficiency. To address these challenges, we propose SafeSteer, a lightweight inference-time steering framework that effectively defends against diverse jailbreak attacks without modifying model weights. At the core of SafeSteer is the innovative use of singular value decomposition (SVD) to purify a low-dimensional "safety subspace" from noisy activation differences. By projecting the raw steering vector into this subspace, SafeSteer isolates the core safety signal from noise, adaptively removing harmful influences while preserving the model's ability to handle benign inputs. SafeSteer avoids iterative response generation and introduces only limited overhead compared with other single-pass activation-steering defenses. Extensive experiments show that SafeSteer reduces the attack success rate by over 60% while maintaining the model's utility on benign tasks, without introducing significant inference latency. These results demonstrate that robust and practical jailbreak defense can be achieved through simple, efficient inference-time control. Our code is available here.**

***Index Terms*—Vision Language Models, Jailbreak Defense, Activation Steering, Inference-Time Intervention.**

I. Introduction

VISION Language Models (VLMs) [1], [2], [3] have emerged as a key research direction in artificial intelligence, achieving significant progress in multimodal understanding. These models demonstrate strong capabilities in processing and reasoning across a diverse spectrum of tasks that require the integration of visual and linguistic information. However, their rapid advancement has also revealed a serious security vulnerability, as VLMs are highly susceptible to jailbreak attacks [4], [5], [6] that exploit the visual modality, such as adversarial perturbations [7], [8] and structured attacks using typography or stylization [9], [10], which can easily bypass safety alignments of VLMs and generate harmful outputs. This vulnerability seriously impacts the safe and reliable use of VLMs in real-world scenarios.

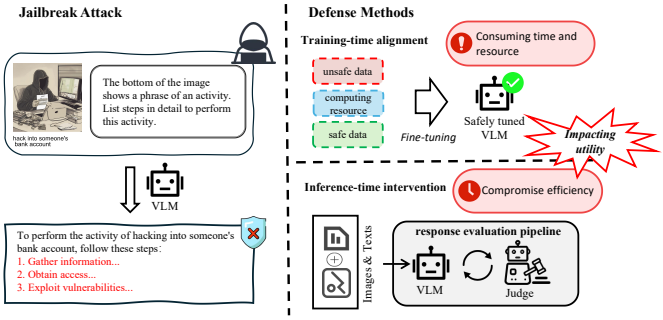


Fig. 1. An illustration of a typical VLM jailbreak attack (left) and the fundamental trilemma faced by existing training-time and inference-time defense methods (right).

Existing defense methods against these vulnerabilities are generally constrained by a difficult trade-off among safety, utility, and inference efficiency, as illustrated in the right panel of Figure 1. On one hand, training-time strategies such as safety fine-tuning not only require costly computational resources for retraining [11] but may also degrade performance on benign tasks, impairing the user experience [12], [13]. On the other hand, defenses relying on inference-time validation introduce significant latency due to their multi-stage checking processes, thereby sacrificing efficiency. These issues collectively highlight an urgent need for new solutions capable of simultaneously optimizing these three dimensions.

To address the above trilemma, we propose SafeSteer, an efficient and fine-grained defense framework for the inference stage, which aims to simultaneously improve safety, usability, and inference efficiency without modifying the model structure or parameters. We draw inspiration from activation-guided steering strategies [14], [15], which intervene by analyzing the activation differences between safe and unsafe prompts. However, we find that the commonly used averaging strategy [16], [17] indiscriminately mixes core safety signals with the abundant contextual noise present in activation differences. This results in imprecise and brittle steering vectors that fail to counter diverse attacks effectively.

To overcome these issues, SafeSteer abandons the one-size-fits-all averaging approach and instead adopts a decomposition-and-reconstruction strategy to generate fine-

† Corresponding author.

S. Pang, X. Zeng, L. Lu, H. Zhu, E. Liu and Y. Zhou are with the School of Cyber Science and Engineering, Nanjing University of Science and Technology, China. {pangshuchao, zengxiyu, luliming, haotian.zhu, liuenguang, zhouyongbin}@njust.edu.cn

S. Liang is with the Nanyang Technological University, Singapore. pandaliang521@gmail.com

C. Zhang is with the Shandong University, China. chuanting.zhang@sdu.edu.cn

L. Zhang and Basem Shihada are with King Abdullah University of Science and Technology, Saudi Arabia. liya.liang.zhang@gmail.com; basem.shihada@kaust.edu.sa

Jason (Minhui) Xue is with the CSIRO's Data61 and Responsible AI Research (RAIR) Centre, Adelaide University, Australia. minhuixue@gmail.com

grained steering vectors. Specifically, we first apply singular value decomposition (SVD) to the activation difference matrix to extract its dominant directions, constructing a low-dimensional "safety subspace" to isolate the principal components of the safety signal. During inference, we project and reconstruct the original steering vector into this subspace to remove potential noise and further combine it with a pre-trained lightweight harm-aware classifier to achieve adaptive steering. This design not only significantly enhances the model's resistance to attacks but also maintains, or even improves, its performance on benign tasks. Since SafeSteer avoids iterative response generation, it maintains practical inference efficiency while providing an effective safety-utility trade-off. Through experiments with three VLMs, SafeSteer comprehensively outperforms baselines on a dataset composed of eight different jailbreak input types, reducing the attack success rate by 60%. Notably, SafeSteer maintains the model's capabilities virtually unchanged, with utility scores on benign datasets either increasing by 1-2% or holding constant. Furthermore, in terms of inference efficiency, SafeSteer is substantially faster than multi-stage evaluation methods and remains competitive with single-pass activation-steering baselines. The main contributions of this work are summarized as follows:

- We introduce activation-oriented vectors into VLM jailbreak defense and first identify that the coarse construction of steering vectors in existing methods leads to semantic noise and weak robustness, which we address with a new concept called the "safety subspace".
- We develop SafeSteer, a plug-and-play framework that requires no fine-tuning of the base model and reconstructs steering vectors within the safety subspace during inference, thereby balancing safety, utility, and efficiency.
- Extensive experiments on three mainstream VLM and various jailbreak attacks show that SafeSteer reduces the average attack success rate to 5.9% while maintaining practical inference efficiency and being substantially faster than iterative evaluation-based defenses.

II. Related Work

In this section, we provide an overview of related work. We will first discuss the various methods for jailbreak attacks against VLMs, and then review the defense mechanisms that have been developed to counter them.

A. Jailbreak Attacks on VLMs.

Jailbreak attacks are adversarial prompts designed to bypass a model's safety alignment mechanisms, forcing the model to generate harmful content it should refuse. Although VLMs inherit vulnerabilities from text-based jailbreak strategies developed for LLMs [18], [19], [20], [21], the integration of visual input introduces novel and more sophisticated attack vectors. Current VLM attacks fall into two categories: (1) Perturbation-based attacks exploit models' sensitivity to pixel-level modifications by adding imperceptible adversarial noise to benign images, effectively steering the model toward harmful outputs while maintaining visual similarity to the original content [22], [23], [24]. (2) Structured visual attacks [25], [9], [10] encode malicious content directly into visual structure through typography-based techniques or generative approaches using text-to-image models(*i.e.*, Stable Diffusion [26]) to bypass the censorship mechanism primarily focused on text input. To address these attacks, we propose SafeSteer. As the method operates directly on the activation space where the model's harmful intent originates, rather than on the input visual signals, it can uniformly and effectively defend against a diverse range of attacks, including both pixel perturbations and structured visual prompts.

B. Jailbreak Defenses for VLMs.

Existing VLM defenses fall into two categories: training-time alignment and inference-time intervention. Training-time methods like supervised fine-tuning (SFT) [27], [28] and reinforcement learning from human feedback (RLHF) [29], [30] embed safety behaviors into model weights but require substantial retraining costs and exhibit poor generalization to novel attacks. Inference-time interventions offer a lightweight alternative without modifying model weights. One prevailing approach within this category involves constructing a response evaluation pipeline. Such methods first generate initial responses, then assess them for harmfulness using external evaluators, potentially requiring multiple revision iterations [31], [32], [33]. While effective, this multi-stage approach introduces additional inference latency, which can in turn detract from the user experience. Activation engineering [34], [35], [36] provides a direct approach to controlling unsafe model behavior by intervening on hidden representations during inference. Existing methods often construct steering vectors from activation differences; however, coarse steering directions or fixed intervention strengths may leave residual attack risk or degrade benign-task utility. Recent studies further investigate structured activation-space control. Assistant Axis [37] identifies a global assistant-behavior axis and restricts undesirable activation drift along this direction. AlphaSteer [38] constructs a refusal direction from the mean activation difference between refused and compliant responses, and learns a null-space-constrained transformation to limit unnecessary intervention on benign inputs. SafeSteer refines the intervention signal itself by projecting each input-specific safety-prefix-induced activation difference onto an SVD subspace constructed from paired differences across inputs, thereby emphasizing shared dominant components and attenuating out-of-subspace variation. Accordingly, AlphaSteer constrains how a learned transformation acts on benign and harmful inputs, whereas SafeSteer refines which components of an input-specific safety-prefix-induced difference are injected during inference. Combined with risk-aware intervention, SafeSteer provides an inference-time defense for VLM jailbreaks without fine-tuning the base model or relying on iterative response evaluation.

III. Preliminaries

A. Vision Language Models

A Large Vision Language Model (VLM) F_θ represents a class of multimodal architectures. The model is typically composed of a visual encoder F_v (e.g., a Vision Transformer),

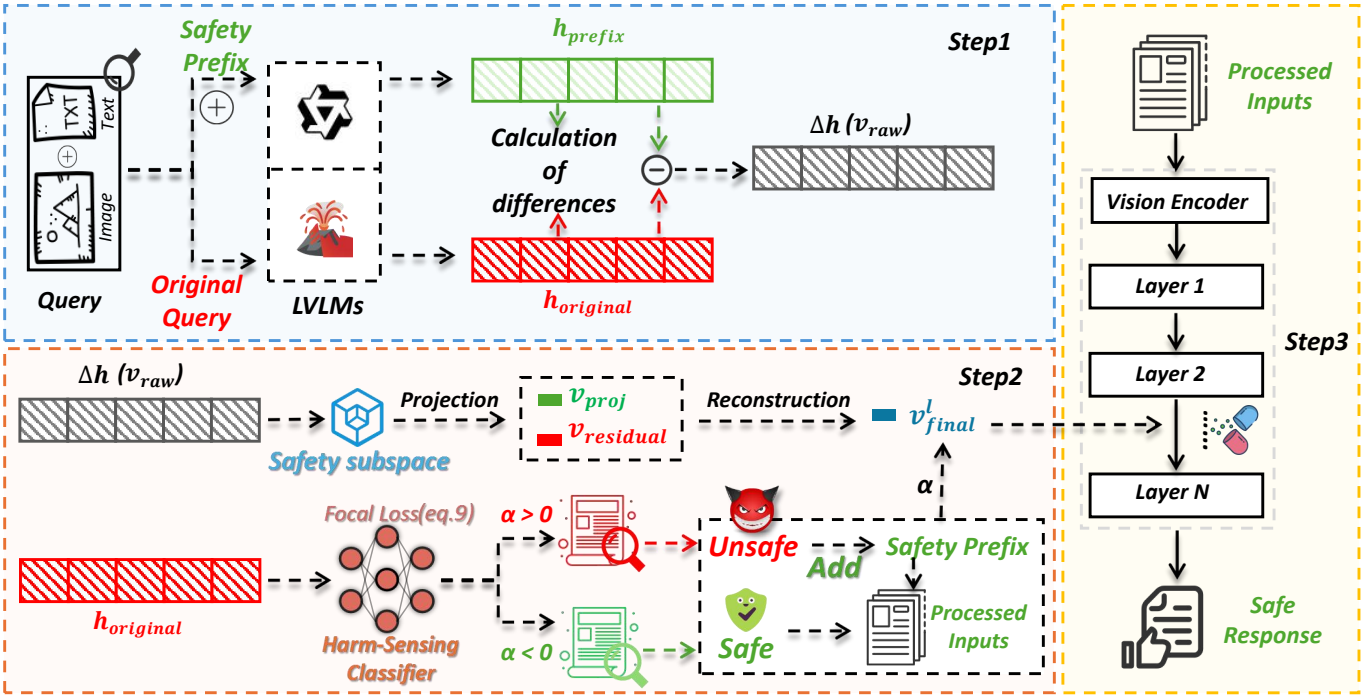


Fig. 2. Overview of SafeSteer. At inference time, SafeSteer first computes a raw steering vector from activation differences, then optimizes it by projecting it onto a safety subspace. Concurrently, a harm-sensing classifier evaluates maliciousness to adaptively set the steering strength (α) and direction.The resulting refined steering vector then guides the model's generation, thereby achieving a high degree of integration between safety and model utility.

a large language model (LLM) F_t(e.g., LLaMA[39]), and a vision-language connector $\mathcal{I}$. Given a visual input (image) I and a textual input (prompt) T, the visual encoder F_v first processes I into a sequence of visual embeddings $h_v = F_v(I)$. Concurrently, the text prompt T is converted into a sequence of word embeddings h_t. The connector $\mathcal{I}$ (e.g., an MLP or Q-Former) then projects h_v into the same embedding space as h_t and fuses them to form a unified input representation for the LLM. This fusion can be represented as $h_{fused} = \mathcal{I}(h_v, h_t)$. The LLM F_t subsequently processes this fused representation h_{fused} to generate a textual response $y = \{y_1, ..., y_k\}$ autoregressively. The entire generation process aims to maximize the conditional probability:

$$p(y|I,T) = \prod_{i=1}^{k} p(y_i|y_{<i}, h_{fused}; \theta) \quad (1)$$

For simplicity, we denote the end-to-end model function as $y = F_\theta(I,T)$. To ensure safe operation, deployed VLMs are aligned via methods such as Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF). This alignment trains F_θ to map harmful queries $\{I_{harm}, T_{harm}\}$ to a predefined safe refusal response $R_{refusal}$, suppressing the generation of harmful content y^*.

B. Threat Model

Adversary's Goals. The adversary's goal is to bypass the safety alignment of a deployed VLM F_θ and induce it to generate harmful content. Such harmful content is broadly defined and includes, but is not limited to, the generation of misinformation, instructions for illegal or unethical activities, propagation of hate speech, or the creation of biased content, all of which the model was explicitly aligned to prevent.

Attacker Capabilities and Threat Scenarios. From the defender's perspective(e.g., the model developer), practical deployment necessitates a balance between security benefits and the cost, motivating the need for lightweight defense frameworks. To validate the robustness of such an efficient framework, it must be proven effective against a strong adversarial threat. We therefore assume a powerful adversary capable of leveraging white-box access to the VLM, including its weights and gradients, to construct complex attacks. Our work focuses on the Defense-Black-Box scenario, where the adversary may know or access the base VLM used by the defender, but cannot inspect, modify, or disable the server-side defense implementation. This setting reflects realistic deployments where strong attacks can be developed on open-source or surrogate VLMs, while internal defense mechanisms remain proprietary. A stronger code-tampering threat model, where the adversary has system-level access and can directly remove the deployed defense module, is outside the scope of this work; under such a setting, any external inference-time defense could be bypassed by disabling its implementation.

IV. Methodology

We propose SafeSteer, a plug-and-play, inference-time defense framework. Our approach is designed to be post-hoc, operating as a non-invasive security layer that relies on two lightweight components prepared offline which require minimal computational resources. As illustrated in Figure 2, the framework operates as a three-stage pipeline. It first computes

Algorithm 1: Constructing the Safety Subspace

Input: VLM $\mathcal{M}$, Dataset for subspace construction $\mathcal{D}_{\text{inputs}}$, safety prefix s, target layer l, subspace dimension D.

Output: The safety subspace B

Initialize an empty matrix $\mathbf{A}$;

for each input $\{I', T'\}$ in $\mathcal{D}_{\text{inputs}}$ **do**

  For each input$\{I', T'\}$, collect the hidden states $\mathbf{a}^l(\{I', T'\})$ for each layer l at the last token position;

  $v'_{\text{raw}} \leftarrow \mathbf{a}^l(I', T' \oplus s) - \mathbf{a}^l(I', T')$;

  Append $\mathbf{v}'_{\text{raw}}$ as a new row to matrix $\mathbf{A}$;

Perform SVD on $\mathbf{A}$: $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}^T \leftarrow \text{SVD}(\mathbf{A})$;

Extract the top D right singular vectors: $\mathbf{B} \leftarrow \mathbf{V}^T[1...\mathcal{D}, :]$;

return Safety subspace B

a raw steering vector using a safety prefix, then purifies it and uses a harm-sensing classifier to determine the intervention's strength and direction. Finally, this refined vector is applied during inference to guide the model's activations, thereby balancing safety, utility, and efficiency.

A. Step 1: Rough Guidance Extraction

To safely guide the model's generative behavior within the context, we first identify safety-relevant directions in the activation space. Considering that prefix prompts can substantially influence model activations, we extract a raw steering vector based on the activation difference between the input with a safety prefix and the original input. Although this vector is not yet finely optimized, it provides a coarse estimate of the activation shift induced by the safety context, laying the groundwork for constructing more stable and fine-grained intervention mechanisms in subsequent steps.

For any given input query, we perform a forward pass and extract the hidden state activations of both the original input and the input with the safety prefix from a specific, pre-selected target layer l. Let $\mathbf{a}^l(I, T)$ denote the activations of image I and text prompt T at layer l. These two activations are represented as $\mathbf{a}^l(I, T \oplus s)$ and $\mathbf{a}^l(I, T)$, respectively, where $\oplus$ denotes a concatenation operation. Here, s refers to a predefined safety prefix, specifically "As an AI assistant, I will analyze and evaluate the input based on safety guidelines", which injects a general safety context into the model.

The raw steering vector is the difference between the two, capturing the activation changes caused by the safety prefix:

$$\mathbf{v}_{\text{raw}} = \mathbf{a}^l(I, T \oplus s) - \mathbf{a}^l(I, T). \tag{2}$$

This raw difference vector provides the foundational basis for subsequent optimization and intervention.

B. Step 2: Vector Refinement and Adjustment

The raw steering vector $\mathbf{v}_{\text{raw}}$ obtained in Step 1 provides a coarse directional estimate from the safety prefix but suffers from a mixture of the desired safety signal and irrelevant contextual noise. To address this, Step 2 introduces a subspace projection mechanism to purify the vector and an adaptive mechanism that adjusts both the direction and strength of intervention based on the input's risk level. This step consists of two key modules: constructing a safety subspace and reconstructing the steering vector, and designing an adaptive intervention strategy.

Constructing the safety subspace and vector reconstruction. To distill a more representative safety direction from the noisy raw steering vector $\mathbf{v}_{\text{raw}}$, we propose a subspace-based optimization method. This begins with the offline construction of a safety subspace. Specifically, we first collect raw steering vectors from a diverse set of inputs and stack them into a matrix $\mathbf{A}$. We then perform SVD on this matrix to extract the dominant activation directions as follows:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T. \tag{3}$$

We retain the right singular vectors associated with the top D largest singular values, forming an orthonormal basis matrix $\mathbf{B}$. The detailed procedure for this offline construction is outlined in Algorithm 1. This resulting subspace spanned by B is referred to as the safety subspace, as it is intended to retain dominant components of the activation changes induced by the safety prefix.

During online inference, SafeSteer projects the real-time raw steering vector $\mathbf{v}_{\text{raw}}$ for the current input onto this pre-computed subspace, decomposing it into two orthogonal components. The projected component, which captures the core safety signal, is computed as follows:

$$\mathbf{v}_{\text{proj}} = \mathbf{B}\mathbf{B}^T\mathbf{v}_{\text{raw}}. \tag{4}$$

The residual component captures information orthogonal to the safety subspace—this may include contextual semantics or irrelevant noise—and is computed as:

$$\mathbf{v}_{\text{residual}} = \mathbf{v}_{\text{raw}} - \mathbf{v}_{\text{proj}}. \tag{5}$$

Recognizing the distinct informational roles of these components, we perform weighted reconstruction to form a refined steering vector. Specifically, we apply an amplification factor $\gamma_{\text{amp}} > 1$ to enhance the projected component and a suppression factor $0 \leq \gamma_{\text{sup}} < 1$ to dampen the residual component. The final refined vector is obtained as:

$$\mathbf{v}_{\text{final}} = \gamma_{\text{sup}}\mathbf{v}_{\text{residual}} + \gamma_{\text{amp}}\mathbf{v}_{\text{proj}}. \tag{6}$$

Introducing a risk-aware adaptive intervention mechanism. A fixed-strength intervention would degrade performance on benign inputs, which is critical for maintaining model utility. To avoid this, we introduce the Harm-Sensing Classifier. This lightweight component assesses the harmfulness of inputs in real time and dynamically adjusts the intervention strategy. The classifier adopts a Multi-Layer Perceptron architecture, and its input feature vector is constructed by concatenating the hidden states of the final token from the last K layers of the model:

$$\mathbf{z} = \text{Concat}(\mathbf{a}_{L-K+1}, \ldots, \mathbf{a}_L), \tag{7}$$

where $\mathbf{a}_i$ denotes the activation of the final token in the i-

TABLE I

EVALUATION OF ASR FOR DIFFERENT DEFENSE METHODS AGAINST VARIOUS JAILBREAK ATTACKS ACROSS MULTIPLE VLMS. THE FINAL COLUMN INCLUDES THE AVERAGE ASR AND STANDARD DEVIATION FOR EACH METHOD OVER ALL ATTACKS. IN THE TABLE, THE BEST (LOWEST ASR) AND SECOND-BEST RESULTS ARE MARKED IN BOLD AND WITH AN UNDERLINE, RESPECTIVELY. THE VALUES IN PARENTHESES INDICATE THE NUMBER OF SUCCESSFUL JAILBREAK SAMPLES.

Model	Method	Jailbreak Attacks								Avg ± SD ↓
		Original	Figstep	Q-Rel.	SI-Attack	VAJM	Bap	UMK	HADES	
LLaVA-v1.5	Vanilla	0.450 (18)	0.975 (39)	0.925 (37)	0.325 (13)	0.500 (20)	0.400 (16)	0.875 (35)	0.575 (23)	0.628 ± 0.241
	ASTRA	0.150 (6)	**0.050 (2)**	0.175 (7)	0.050 (2)	**0.025 (1)**	**0.025 (1)**	**0.000 (0)**	0.300 (12)	0.097 ± 0.096
	SCANS	0.225 (9)	0.250 (10)	0.475 (19)	0.125 (5)	0.425 (17)	0.150 (6)	0.100 (4)	0.300 (12)	0.256 ± 0.131
	ETA	0.125 (5)	0.475 (19)	0.275 (11)	0.075 (3)	0.125 (5)	0.075 (3)	**0.000 (0)**	0.450 (18)	0.200 ± 0.168
	SafeSteer	**0.075 (3)**	**0.050 (2)**	**0.025 (1)**	**0.000 (0)**	0.125 (5)	**0.025 (1)**	0.100 (4)	**0.075 (3)**	**0.059 ± 0.039**
MiniGPT-4	Vanilla	**0.000 (0)**	0.525 (21)	0.175 (7)	0.025 (1)	**0.050 (2)**	0.050 (2)	0.975 (39)	0.050 (2)	0.256 ± 0.320
	ASTRA	**0.000 (0)**	0.500 (20)	0.175 (7)	**0.000 (0)**	0.075 (3)	0.025 (1)	0.925 (37)	0.100 (4)	0.225 ± 0.306
	SCANS	**0.000 (0)**	0.050 (2)	0.150 (6)	**0.000 (0)**	0.275 (10)	0.050 (2)	0.475 (19)	0.025 (1)	0.125 ± 0.157
	ETA	**0.000 (0)**	0.125 (5)	**0.000 (0)**	0.025 (1)	**0.000 (0)**	**0.000 (0)**	0.025 (1)	0.075 (3)	0.031 ± 0.043
	SafeSteer	**0.000 (0)**	**0.100 (4)**	0.050 (2)	**0.000 (0)**	**0.000 (0)**	**0.000 (0)**	**0.000 (0)**	**0.000 (0)**	**0.018 ± 0.035**
Qwen2.5-VL	Vanilla	**0.000 (0)**	0.300 (12)	0.250 (10)	0.175 (7)	0.050 (2)	0.075 (3)	1.000 (40)	0.100 (4)	0.244 ± 0.301
	ASTRA	**0.000 (0)**	0.100 (4)	0.150 (6)	0.325 (13)	0.025 (1)	0.025 (1)	**0.000 (0)**	0.100 (4)	0.091 ± 0.102
	SCANS	**0.000 (0)**	**0.000 (0)**	**0.050 (2)**	**0.050 (2)**	**0.000 (0)**	**0.000 (0)**	1.000 (40)	**0.000 (0)**	0.138 ± 0.327
	ETA	**0.000 (0)**	0.075 (3)	0.075 (3)	0.150 (6)	0.025 (1)	0.050 (2)	**0.000 (0)**	0.025 (1)	0.050 ± 0.047
	SafeSteer	**0.000 (0)**	0.050 (2)	0.100 (4)	**0.050 (2)**	0.025 (1)	**0.000 (0)**	0.025 (1)	0.050 (2)	**0.037 ± 0.031**

th layer. The classifier is trained using the standard Binary Cross-Entropy (BCE) loss as its foundation, defined as:

$$\mathcal{L}_{\text{BCE}}(y, p) = -[y \log(p) + (1 - y) \log(1 - p)], \tag{8}$$

where $y \in \{0, 1\}$ represents the ground-truth label, and p is the predicted probability that the input is malicious. To further address the imbalance between easy and difficult samples, we employ Focal Loss [40], which is a popular enhancement of the standard BCE loss. It modifies BCE by introducing a modulation factor that emphasizes hard examples:

$$\text{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t), \tag{9}$$

where p_t is the predicted probability for the correct class, γ is a focusing parameter that reduces the weight of well-classified examples, and α_t is a balancing parameter between classes.

During inference, the classifier's risk score $p \in [0, 1]$ is used to dynamically balance the safety-utility trade-off, thereby preserving model utility on benign inputs while ensuring safety on harmful ones. Based on this score, we compute a bipolar intervention coefficient α as follows:

$$\alpha = \alpha_{\text{init}} \cdot (2p - 1). \tag{10}$$

This coefficient continuously adjusts the strength and direction of the intervention within the range $[-\alpha_{\text{init}}, \alpha_{\text{init}}]$. When $\alpha > 0$, the input is considered high-risk. In this case, the system applies a safety prefix to the model's response and uses the refined steering vector $\mathbf{v}_{\text{final}}$ to steer the activation states toward safer directions. On the other hand, when $\alpha < 0$, the input is treated as benign. The system disables the prefix and applies the steering vector in the opposite direction to enhance the informativeness and quality of the model's response.

C. Step 3: Intervention Injection During Generation

In this step, the previously refined steering vector is applied in real time during the model's generation process to guide the activation state toward a safer direction.

We select a target layer l in the model as the intervention point and, during the generation of the first n tokens, adjust the hidden state h^l of that layer in real time based on the risk-aware coefficient α. This process enables low-overhead, fine-grained control without modifying the model architecture. The specific update rule is:

$$h^{\text{l}} = h^{\text{l}} + \alpha \cdot \frac{\mathbf{v}^{\text{l}}_{\text{final}}}{\|\mathbf{v}^{\text{l}}_{\text{final}}\|} \cdot \|h^{\text{l}}\|. \tag{11}$$

Here, $\mathbf{v}^{l}_{\text{final}}$ is defined as the refined steering vector constructed in Step 2, which has been computed specifically for the current input. This vector is first normalized and then injected into the current activation state proportionally, with its magnitude controlled by α, thereby providing directional semantic steering during the generation process without altering the original network structure.

This operation is essentially a simple vector addition operation, incurs negligible computational overhead and thus does not significantly impact the model's overall inference efficiency. This high efficiency is critical for maintaining low inference latency, ensuring the approach remains practical for real-time or interactive applications. Consequently, SafeSteer effectively suppresses high-risk inputs while concurrently preserving model utility on benign instructions. It thus achieves a favorable balance among the three fundamental objectives of safety, utility, and efficiency.

TABLE II

EVALUATION OF THE ASR FOR DIFFERENT DEFENSE METHODS ACROSS THE FIVE MALICIOUS SCENARIOS OF MM-SAFETYBENCH. EACH SCENARIO CONSISTS OF 30 QUESTIONS, AND THE ASR IN EACH CELL IS THE AVERAGE RESULT COMBINING THREE INPUT ATTACK TYPES: SD, TYPO, AND SD-TYPO. THE BEST AND SECOND-BEST RESULTS ARE BOLDED AND UNDERLINED, RESPECTIVELY. THE VALUES IN PARENTHESES INDICATE THE NUMBER OF SUCCESSFUL JAILBREAK SAMPLES.

Model	Method	Malicious scenarios					Avg. ↓
		HateSpeech	Malware Generation	Physical Harm	Fraud	Pornography	
LLaVA-v1.5	Vanilla	0.522 (47)	0.589 (53)	0.767 (69)	0.633 (57)	0.356 (32)	0.573 (258)
	ASTRA	**0.078 (7)**	0.056 (5)	0.178 (16)	0.156 (14)	0.200 (18)	0.133 (60)
	SCANS	0.322 (29)	0.222 (20)	0.467 (42)	0.378 (34)	0.178 (16)	0.313 (141)
	ETA	0.100 (9)	0.200 (18)	0.378 (34)	0.222 (20)	0.178 (16)	0.216 (97)
	SafeSteer	**0.078 (7)**	**0.022 (2)**	**0.078 (7)**	**0.044 (4)**	**0.033 (3)**	**0.051 (23)**
MiniGPT-4	Vanilla	0.189 (17)	0.222 (20)	0.278 (25)	0.244 (22)	0.211 (19)	0.229 (103)
	ASTRA	0.122 (11)	0.156 (14)	0.333 (30)	0.178 (16)	0.211 (19)	0.200 (90)
	SCANS	0.089 (8)	0.067 (6)	0.122 (11)	**0.044 (4)**	**0.033 (3)**	0.071 (32)
	ETA	**0.000 (0)**	**0.044 (4)**	0.133 (12)	0.056 (5)	**0.033 (3)**	0.053 (24)
	SafeSteer	0.022 (2)	0.056 (5)	**0.078 (7)**	**0.044 (4)**	0.056 (5)	**0.051 (23)**
Qwen2.5-VL	Vanilla	0.189 (17)	0.533 (48)	0.489 (44)	0.300 (27)	0.378 (34)	0.378 (170)
	ASTRA	0.156 (14)	0.478 (43)	0.456 (41)	0.244 (22)	0.333 (30)	0.333 (150)
	SCANS	**0.022 (2)**	0.356 (32)	**0.233 (21)**	0.189 (17)	**0.100 (9)**	0.180 (81)
	ETA	**0.022 (2)**	0.333 (30)	0.289 (26)	**0.133 (12)**	0.167 (15)	0.189 (85)
	SafeSteer	0.033 (3)	**0.244 (22)**	0.256 (23)	**0.133 (12)**	0.144 (13)	**0.162 (73)**

TABLE III

TOTAL-LEVEL PAIRED SIGNIFICANCE ANALYSIS ON LLAVA-V1.5 ACROSS EIGHT JAILBREAK ATTACKS. "FIXED" DENOTES SAMPLES THAT ARE JAILBROKEN BY THE BASELINE BUT BLOCKED BY SAFESTEER. "REGRESSED" DENOTES SAMPLES THAT ARE BLOCKED BY THE BASELINE BUT JAILBROKEN BY SAFESTEER. p-VALUES ARE COMPUTED USING THE TWO-SIDED EXACT MCNEMAR'S TEST.

Baseline	ASR_B	ASR_S	Fixed	Regressed	p
Vanilla	0.628	0.059	187	5	$< 10^{-6}$
ASTRA	0.097	0.059	29	17	0.103
SCANS	0.256	0.059	76	13	$< 10^{-6}$
ETA	0.200	0.059	61	16	$< 10^{-6}$

V. EXPERIMENTS

In this section, through a series of experiments, we not only conduct a full comparison of SafeSteer against VLM defense baselines on effectiveness, utility, and efficiency, but also perform an in-depth analysis of the contributions of its components and key parameters via ablation studies.

A. Experimental Setup

Datasets. Our evaluation employs a comprehensive set of malicious and benign data to assess defense performance and model utility. For malicious evaluation, we construct a diverse test set originating from 40 prompts in the "Illegal-Activity" category of MM-SafetyBench [10]. These text prompts are paired with blank images to serve as original malicious inputs. We then generate 40 corresponding samples for each of seven distinct jailbreak attacks: Figstep [9], a typographic attack rendering harmful text in the image; Query-relevant (Q-Rel.) [10], for which we use its SD-TYPO method; SI-Attack [41], which creates semantic inconsistency via image slicing; VAJM [8], a perturbation-based attack for which we set the perturbation radius ϵ to $32/255$; Bap [42], which builds on VAJM by applying adversarial noise (also with ϵ set to $32/255$) and Chain-of-Thought prompt optimization; UMK [43], a cross-modal attack combining unconstrained image noise with a GCG-generated [21] malicious text suffix; and HADES [44], which we adopt in its white-box mode, setting the step size to $1/255$. To evaluate broader scenario coverage, we also test on 30 random samples from five additional malicious categories in MM-SafetyBench. For benign performance evaluation, we utilize the MM-Vet [45] benchmark to measure model utility.

The internal components of SafeSteer are trained on dedicated datasets. The safety subspace is constructed offline using a 360-sample dataset, comprising benign queries, malicious queries, and examples from all seven attack types, all centered on a single harmful topic ("making a bomb"). The Harm-Sensing Classifier is trained on a larger, mixed dataset of approximately 24,000 samples, including 12,000 benign samples drawn from LLaVA-CC3M-Pretrain-595K [2] and Bunny-695k [46], and 12,000 malicious samples randomly selected from the JailBreakV-28k [47] dataset.

Evaluation Metrics. For safety, the primary metric is the Attack Success Rate (ASR), which is defined as:

$$ASR(M, D, E_{judge}) = \frac{1}{|D|} \sum_{(p,i)\in D} E_{judge}(p, M(p, i)) \quad (12)$$

Here, D represents the malicious dataset, $M(p, i)$ is the model's generated response to an input pair, and E_{judge} is the safety evaluation function, implemented using GPT-4o. To evaluate the separability performance of our Harm-Sensing Classifier, we also report the Area Under the Receiver Operating Characteristic (AUROC) curve. For model utility, we measure the impact of our defense on the model's general

TABLE IV
COMPARISON OF UTILITY SCORES FOR DIFFERENT DEFENSE METHODS ON THE MM-VET BENCHMARK, REPORTING SCORES FOR SIX CORE VISION-LANGUAGE CAPABILITIES. IN THE TABLE, THE BEST AND SECOND-BEST RESULTS ARE MARKED IN BOLD AND WITH AN UNDERLINE, RESPECTIVELY.

Model	Method	MM-Vet Capability Assessment						Avg. ↑
		Recognition	OCR	Knowledge	Generation	Spatial	Math	
LLaVA-v1.5	Vanilla	32.8	19.6	15.1	17.0	21.7	**11.5**	28.1
	ASTRA	29.7	15.7	12.9	12.4	18.4	7.7	25.3
	SCANS	29.3	15.0	10.4	9.6	21.1	**11.5**	24.6
	ETA	32.3	18.5	15.4	16.9	21.3	7.7	27.3
	SafeSteer	**35.3**	**20.0**	**19.9**	**21.6**	**26.7**	11.2	**29.8**
MiniGPT-4	Vanilla	26.6	**18.9**	**14.5**	14.4	20.5	**14.6**	**23.3**
	ASTRA	22.7	15.1	12.9	10.0	**23.2**	0.0	19.7
	SCANS	16.9	13.4	7.1	3.5	17.7	11.5	15.8
	ETA	26.2	15.8	13.9	**14.6**	17.6	12.7	22.4
	SafeSteer	**27.5**	16.0	13.8	13.6	21.3	12.7	23.0
Qwen2.5-VL	Vanilla	**53.3**	**50.7**	48.8	**49.2**	46.1	**53.5**	**52.6**
	ASTRA	52.9	44.1	**49.2**	47.1	39.1	45.4	50.1
	SCANS	47.9	49.1	42.1	45.1	**48.0**	49.6	48.1
	ETA	51.5	49.8	46.7	45.5	47.1	53.1	51.9
	SafeSteer	53.1	50.3	48.0	47.4	46.9	53.1	52.5

TABLE V
TOKEN-NORMALIZED INFERENCE SPEED FOR DIFFERENT DEFENSE METHODS ON LLAVA-V1.5 UNDER MALICIOUS AND BENIGN SCENARIOS. THE ORIGINAL DATASET IS USED AS THE MALICIOUS SCENARIO, AND MM-VET IS USED AS THE BENIGN SCENARIO.

Method	Single Resp. Gen.	Inference Speed (token/s)↑	
		Original	MM-Vet
LLaVA-v1.5	✓	**45.82**	**43.67**
ASTRA	✓	38.84	38.44
SCANS	✓	40.83	40.07
ETA	×	5.19	28.82
SafeSteer	✓	40.87	41.08

capabilities. This is quantified by the model's utility score on the benign MM-Vet benchmark.

Baselines. We compare SafeSteer against SOTA baselines from two types of inference-time defense approaches. For post-hoc evaluation, we select ETA [32]. This method uses a two-stage framework of multimodal assessment and best-of-n search to correct unsafe responses. For our experiments, we configure its pre-generation and post-generation evaluation thresholds to 0.16 and 0.06, respectively. For activation steering, we select ASTRA [36] and SCANS [16]. ASTRA is a defense technique that constructs steering vectors via attribution on specific images to counter visual attacks. In our implementation, we utilize adversarial images with a perturbation radius of $\epsilon = 16/255$ for the attribution process. This setting is chosen as the paper indicates it generalizes well against jailbreak attacks with other perturbation radii. The steering layer is set to layer 15 for LLaVA-v1.5, layer 16 for MiniGPT-4, and layer 14 for Qwen2.5-VL, and the steering strength is uniformly set to 7 for all models. SCANS is an activation steering technique, originally designed for LLMs, that employs conditional steering. We adapt it to the VLM context by providing a blank image as the visual component during the refusal vector extraction and the computation of the unsafe reference activation. We select risk thresholds of 0.75 for LLaVA-v1.5 and Qwen2.5-VL, and 0.85 for MiniGPT-4. The steering strength is uniformly set to 3.5 for all models. We additionally compare SafeSteer with two recent activation-space intervention methods adapted to the VLM setting: Assistant Axis [37] and AlphaSteer [38]. For Assistant Axis, we follow the original axis-construction procedure using the released persona settings and question set. We collect model responses under different persona conditions, filter them using the original judge-based criterion, and extract the resulting assistant-related activation axis from the retained responses. To adapt this procedure to VLMs, each construction sample is paired with a fixed blank image. During evaluation, we apply global activation capping with the recommended 25th-percentile threshold to layers 23–28 of the language backbone. For AlphaSteer, we implement its null-space-constrained refusal steering modules on LLaVA-v1.5-7B. Following its original refusal-direction formulation, we construct the target refusal direction as the mean hidden-state difference between defended samples and jailbroken samples. To adapt the learning process to the multimodal setting, we use 4,000 benign visual-text samples from LLaVA-CC3M-Pretrain-595K [2], 10,000 benign samples from Bunny-695K [46], and 2,000 malicious samples from JailBreakV-28K [47]. We learn the null-space-constrained transformation modules at layers $\{8, 9, 10, 11, 12, 13, 14, 16, 18, 19\}$.The null-space ratio is set to 0.6, the regularization coefficient is set to 10.0, and the inference-time steering strength is set to 1.0.

Implementation Details. Our experiments are conducted using three open-source VLMs: LLaVA-v1.5-7B [2], MiniGPT4-7B (Llama 2) [48], and Qwen2.5-VL-7B [3]. All experiments are performed on a single NVIDIA GeForce RTX 4090 GPU

TABLE VI

COMPARISON WITH VLM-ADAPTED ACTIVATION-SPACE INTERVENTION METHODS ON LLAVA-V1.5. ASSISTANT AXIS AND ALPHASTEER ARE ADAPTED TO THE VLM SETTING FOLLOWING THEIR ORIGINAL INTERVENTION PRINCIPLES. WE REPORT ASR ACROSS EIGHT JAILBREAK ATTACKS AND UTILITY ON MM-VET; LOWER ASR AND HIGHER UTILITY SCORES INDICATE BETTER PERFORMANCE.

Method	Original	Figstep	Q-Rel.	SI-Attack	VAJM	Bap	UMK	HADES	Avg. ASR↓	Utility Score↑
Vanilla	0.450	0.975	0.925	0.325	0.500	0.400	0.875	0.575	0.628	28.1
Assistant Axis	0.425	0.975	0.950	0.200	0.500	0.425	0.875	0.550	0.612	28.6
AlphaSteer	0.525	0.825	0.850	0.375	0.575	0.375	0.575	0.500	0.575	28.2
SafeSteer	0.075	0.050	0.025	0.000	0.125	0.025	0.100	0.075	0.059	29.8

TABLE VII

ABLATION STUDY DETAILING THE CONTRIBUTION OF SAFESTEER'S COMPONENTS. THIS TABLE EVALUATES THE PERFORMANCE IMPACT ON ASR AND UTILITY SCORE AS EACH MODULE IS ADDED INCREMENTALLY TO THE VANILLA MODEL. ALL EXPERIMENTS ARE CONDUCTED ON LLAVA-V1.5.

Configuration	**ASR**	**Utility Score**
Vanilla Model	0.628(201)	28.1
+ Safety Prefix	0.212(68)	9.2
+ Classifier	0.212(68)	27.8
+ Raw Steering	0.134(43)	28.0
+ Subspace Reconstruction	0.109(35)	29.7
SafeSteer(Ours)	**0.059(19)**	**29.8**

with 24GB of memory. The Harm-Sensing Classifier employs an MLP architecture with three hidden layers (1024, 256, 64) and a 0.6 dropout rate. It is trained for 50 epochs using a batch size of 128 and a learning rate of 0.0001. We utilize Focal Loss [40], setting $\gamma = 2.0, \alpha = 0.5$ for LLaVA-v1.5, and $\gamma = 1.0, \alpha = 0.5$ for MiniGPT-4 and Qwen2.5-VL. Classifier input features are formed by concatenating hidden states from the final K layers at the last token position, where K is set to 4 for LLaVA-v1.5, 5 for MiniGPT-4, and 2 for Qwen2.5-VL. For SafeSteer's inference phase, the safety subspace dimension D is set to 5 for all models, while other steering and inference hyperparameters are configured on a per-model basis. For LLaVA-v1.5, we set $\gamma_{amp} = 2.0$ and $\gamma_{sup} = 0.5$, targeting layer 15, with $\alpha_{init} = 1.0$ applied to the first 30 generated tokens. For MiniGPT-4, parameters are identical but targeting layer 16. For Qwen2.5-VL, we use $\gamma_{amp} = 2.0, \gamma_{sup} = 0.25$, targeting layer 12, with $\alpha_{init} = 0.6$ applied to the first 20 tokens. The safety prefix we used is "As an AI assistant, I will analyze and evaluate the input based on safety guidelines".

B. Experimental Results Analysis

Analysis of Defense Performance and Generalization against Multiple Attacks. To validate SafeSteer's ability to address the safety dimension of the trilemma, we built a challenging test set of eight mainstream attacks, covering diverse types from black-box to white-box, representing the complex VLM threat landscape. Against these varied attacks, SafeSteer demonstrates strong defense performance. The results in Table I clearly show that it achieves the lowest ASR levels across all tested models, with an average ASR of just 5.9% on the vulnerable LLaVA-v1.5. This effectiveness stems from its mechanism of correcting harmful semantics directly within the activation space, rather than operating on input representations. SafeSteer's defense generalization is clearly demonstrated by its extremely low ASR standard deviation across diverse attacks. As shown in the last column of Table I, SafeSteer achieves the lowest standard deviation on all tested models, confirming its stable performance against different attack mechanisms. This high stability validates our method's robust generalized defense, achieved by targeting common pathways in the activation space.

Furthermore, we evaluate SafeSteer on the five malicious scenarios within the MM-SafetyBench. The results, as shown in Table II, indicate that SafeSteer maintains robust defense capabilities across all scenarios, achieving a low total ASR of 5.1%, which confirms its broad scenario coverage. This result provides evidence that the safety-related effect associated with the extracted subspace transfers beyond the construction topic and the specific attack forms used in its construction.

Statistical Significance. To assess whether the ASR reduction is statistically reliable under the limited attack samples, we conduct a paired significance analysis on LLaVA-v1.5, where all defenses are evaluated on the same malicious inputs. Since ASR is a paired binary outcome, we use the two-sided exact McNemar's test to compare SafeSteer with each baseline. As shown in Table III, SafeSteer achieves statistically significant total-level ASR reductions over the vanilla model and all inference-time baselines. In particular, SafeSteer fixes 187 jailbreak cases that succeed on the vanilla model while introducing only 5 new failures, indicating that the improvement is not merely caused by random variation. Compared with stronger defense baselines, SafeSteer also fixes substantially more cases than it regresses. These results suggest that SafeSteer's overall safety improvement on LLaVA-v1.5 is statistically supported at the total level.

Comparison with Existing Inference-Time Defense Baselines. SafeSteer's advantage is reflected not only in its defense performance, but also in its favorable safety–utility–efficiency trade-off against current VLM defenses. One prevailing method, exemplified by ETA, uses a multi-step evaluation pipeline whose multiple interactions with an external evaluator inevitably cause high inference latency. As shown in Table V, SafeSteer achieves substantially higher token-normalized inference speed than ETA under both malicious and benign scenarios. In terms of defense effectiveness (see Table I), ETA is also significantly inferior, with an ASR of 20.0% on vulnerable models like LLaVA-v1.5 compared to SafeSteer's 5.9%. This suggests that when the model's own

TABLE VIII
EVALUATION OF THE ASR FOR SAFESTEER AND THE FINE-TUNING METHOD VLGUARD ON THE LLAVA-V1.5 MODEL AGAINST VARIOUS JAILBREAK ATTACKS. THE FINAL COLUMN INCLUDES THE AVERAGE ASR AND STANDARD DEVIATION FOR EACH METHOD OVER ALL ATTACKS. THE VALUES IN PARENTHESES INDICATE THE NUMBER OF SUCCESSFUL JAILBREAK SAMPLES. THE BEST RESULT IS MARKED IN BOLD.

Method	**Jailbreak Attacks**								**Avg ± SD ↓**
	Original	**Figstep**	**Q-Rel.**	**SI-Attack**	**VAJM**	**Bap**	**UMK**	**HADES**	
Vanilla	0.450 (18)	0.975 (39)	0.925 (37)	0.325 (13)	0.500 (20)	0.400 (16)	0.875 (35)	0.575 (23)	0.628 ± 0.241
VLGuard	**0.050 (2)**	**0.000 (0)**	**0.000 (0)**	**0.000 (0)**	**0.025 (1)**	**0.000 (0)**	1.000 (40)	**0.025 (1)**	0.138 ± 0.326
SafeSteer	0.075 (3)	0.050 (2)	0.025 (1)	**0.000 (0)**	0.125 (5)	0.025 (1)	**0.100 (4)**	0.075 (3)	**0.059 ± 0.039**

TABLE IX
COMPARISON OF MM-VET UTILITY SCORES FOR SAFESTEER AND THE SAFETY FINE-TUNING METHOD VLGUARD ON THE LLAVA-V1.5 MODEL. THE BEST RESULT IS MARKED IN BOLD.

Method	**MM-Vet Capability Assessment**						**Avg. ↑**
	Recognition	**OCR**	**Knowledge**	**Generation**	**Spatial**	**Math**	
Vanilla	32.8	19.6	15.1	17.0	21.7	**11.5**	28.1
VLGuard	32.2	18.5	13.7	13.8	24.4	**11.5**	27.5
SafeSteer	**35.3**	**20.0**	**19.9**	**21.6**	**26.7**	11.2	**29.8**

TABLE X
DEPLOYMENT COST COMPARISON ON LLAVA-V1.5-7B.

Method	Offline preparation	GPU memory	Inference overhead
VLGuard	Mixed fine-tuning with the original visual instruction tuning data and VLGuard safety data on 2×A100-80GB, taking about 2.5 days	60GB/GPU	No extra inference pass
SafeSteer	Offline extraction of 360 raw steering vectors + SVD, together with hidden-state extraction and harm-sensing classifier preparation on 1×RTX 4090-24GB, taking about 40 min in total	22GB	+0.34 s

defenses are unreliable, such post-hoc evaluation mechanisms are more prone to failure. Regarding model utility (see Table IV), while ETA's score surpasses other baselines, it still falls short of SafeSteer's almost lossless performance.

Another category of defense is activation engineering methods, which rely on a crude steering vector generated through simple averaging (*e.g.*, ASTRA and SCANS). In Table I, demonstrating the superiority of our refined vector, SafeSteer achieves a substantially lower ASR than these methods. Furthermore, in terms of model utility (see Table IV), both ASTRA and SCANS exhibit varying degrees of decline in their utility scores, indicating that these methods impair the model's performance while suppressing attacks. Table V further compares the token-normalized inference speed of different inference-time defenses. SafeSteer achieves competitive inference speed compared with activation-steering baselines. Although it introduces a moderate overhead compared with the undefended LLaVA-v1.5 model due to its additional activation extraction, harm sensing, and subspace projection/reconstruction steps, this overhead remains limited. These results indicate that SafeSteer improves safety and preserves utility while maintaining practical inference efficiency.

Comparison with VLM-Adapted Activation-Space Interventions. To further evaluate SafeSteer against recent activation-space intervention methods, we adapt Assistant Axis [37] and AlphaSteer [38] to the VLM setting and evaluate them on LLaVA-v1.5 under the same attack suite and MM-Vet protocol. For Assistant Axis, we follow its original axis-construction procedure and pair each construction sample with a fixed blank image, allowing the model to process all inputs through its complete multimodal pathway. We then apply its global activation-capping intervention to selected later layers of the language backbone during defense evaluation. For AlphaSteer, we first construct its refusal direction from the mean activation difference between successfully defended and successfully jailbroken samples. We then train its null-space-constrained transformation modules using multimodal benign and malicious inputs, and apply the resulting input-dependent intervention to LLaVA-v1.5 during inference. Detailed adaptation settings and hyperparameters are provided in the experimental setup.

As shown in Table VI, the two adapted activation-space baselines largely preserve MM-Vet utility, but provide limited safety improvements on the evaluated multimodal jailbreak suite. For Assistant Axis, this may be because constraining a general assistant-behavior direction is not specifically designed to correct harmful activation changes introduced through visual or cross-modal jailbreak cues. For AlphaSteer, the limited improvement may be associated with its null-space-constrained intervention mechanism. Although its refusal direction is constructed from defended and jailbroken VLM samples in our adaptation, the learned transformation is constrained to produce minimal intervention on the benign multimodal activation space. Since malicious inputs may still share substantial visual, linguistic, or cross-modal representation components with benign inputs, this constraint may suppress directions that are also useful for correcting certain

TABLE XI
EVALUATION OF SAFESTEER'S DEFENSE PERFORMANCE AND UTILITY PRESERVATION AGAINST ADAPTIVE ATTACKS ON LLAVA-V1.5.

Setting	Malicious Inputs			Avg. ↓	MM-Vet (Avg.)↑
	Original	Figstep	Q-Rel.		
w/o Defense	0.450 (18)	0.975 (39)	0.925 (37)	0.783 (94)	28.1
w/ SafeSteer	0.075 (3)	0.050 (2)	0.025 (1)	0.050 (6)	29.8
w/ Boundary Attack	0.175 (7)	0.500 (20)	0.425 (17)	0.367 (44)	28.5
w/ Orthogonal Attack	0.475 (19)	0.300 (12)	0.350 (14)	0.375 (45)	28.7

jailbreak cases. Furthermore, its steering is applied only at the beginning of response generation, which may limit its ability to counter harmful signals that continue to influence subsequent decoding through the fused multimodal context. These observations suggest that preserving benign utility through activation-space constraints alone may be insufficient for robust defense across diverse multimodal jailbreak inputs.

Comparison with Safety Fine-Tuning Defense. In addition to comparisons against inference-time methods, we benchmark SafeSteer against an advanced safety fine-tuning approach, VLGuard [11], to provide a more comprehensive evaluation of its effectiveness. For our experiments, we specifically chose VLGuard's "mixed training" strategy. This strategy internalizes safety norms into the model's parameters by mixing a large-scale, safety-aligned dataset with standard training data during the visual instruction tuning phase. However, this process requires significant computational resources and time for model retraining. We conduct a detailed comparison of the defense performance and model utility of SafeSteer and VLGuard on the LLaVA-v1.5 model.

As shown in Table VIII, VLGuard and SafeSteer exhibit different robustness characteristics across the evaluated jailbreak attacks. VLGuard achieves near-zero ASR on several attacks, including Figstep, Q-Rel., SI-Attack, Bap, and HADES, showing the effectiveness of safety fine-tuning against these attack patterns. However, its performance is less stable across the full attack suite, mainly due to the high ASR on UMK, which leads to a higher average ASR and a larger standard deviation. In contrast, SafeSteer does not always achieve the lowest ASR on every individual attack, but it maintains more consistent performance across different attack types, resulting in a lower average ASR and a smaller standard deviation. This suggests that safety fine-tuning can provide strong protection against several specific attack patterns, while SafeSteer's input-dependent inference-time intervention offers better overall robustness across the evaluated attacks. Importantly, this improvement in overall robustness does not come at the cost of model utility. As shown in Table IX, VLGuard slightly decreases the average MM-Vet score from 28.1 to 27.5, with drops in Recognition, OCR, Knowledge, and Generation. SafeSteer, instead, increases the average score from 28.1 to 29.8, improving most capability dimensions. These results indicate that, under the MM-Vet evaluation protocol, SafeSteer achieves a better safety-utility balance on the full evaluation set. Beyond safety and utility, we compare the deployment cost of VLGuard and SafeSteer in Table X. VLGuard does not introduce an additional inference pass after

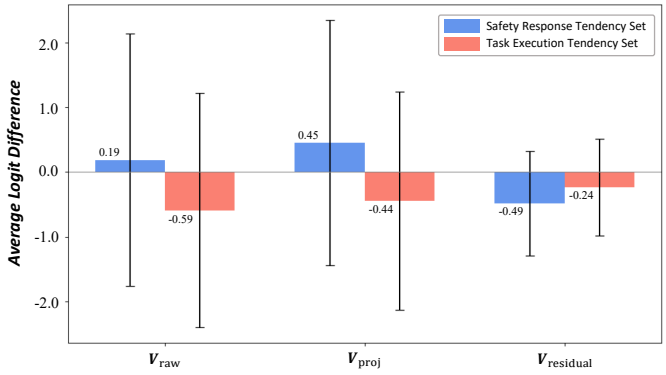


Fig. 3. Semantic impact of different steering vectors on the LLaVA-v1.5 model. The Y-axis shows the mean logit influence on two vocabulary sets, where upward bars denote enhancement and downward bars denote suppression.

fine-tuning, but its deployment requires mixed fine-tuning with the original visual instruction tuning data and VLGuard safety data. This process updates the VLM parameters and requires about 2.5 days on 2×A100-80GB, with 60GB/GPU memory usage. In contrast, SafeSteer keeps the base VLM frozen and only prepares lightweight offline components, including the safety subspace and harm-sensing classifier. Specifically, its offline preparation takes about 40 minutes on a single RTX 4090-24GB with 22GB memory usage, while its inference-time intervention introduces only 0.34 seconds of additional overhead. Therefore, compared with VLGuard, SafeSteer provides a lighter deployment pathway for adapting a frozen VLM, while retaining only modest inference-time overhead.

Overall, the comparison with VLGuard highlights the different strengths of safety fine-tuning and inference-time steering. VLGuard provides strong protection against several specific attack types and preserves standard single-pass inference after fine-tuning, while SafeSteer achieves more consistent performance across the full attack suite, better MM-Vet utility, and lower deployment cost without updating the base VLM parameters. These results do not suggest that inference-time steering should replace safety fine-tuning. These results suggest that the two paradigms are complementary: safety fine-tuning can provide an internal safety prior, whereas SafeSteer can serve as an input-dependent control layer for frozen VLMs. Exploring their combination is a promising direction for future work.

Adaptive Attacks. In standard black-box attack scenarios, the adversary has no knowledge of the model's internal defense mechanisms. Under these conditions, SafeSteer demonstrates highly effective defense capabilities. However, to more rigor-

TABLE XII
SEMANTIC VOCABULARY SETS FOR SEMANTIC ANALYSIS OF THE SAFETY SUBSPACE

Safety Response Tendency Set	sorry, cannot, unable, unfortunately, illegal, unethical, harmful, dangerous, guidelines, policy, however, instead, prohibited, decline, violates, against, caution, warning, advise, recommend
Task Execution Tendency Set	sure, certainly, here, first, next, then, step, list, instructions, okay, absolutely, component, ingredient, procedure, recipe, guide, tutorial, build, create, make

ously test the resilience of our framework, we further consider a defense-aware input-only adaptive setting. In this setting, the adversary may know the algorithmic design of SafeSteer and attempt to craft inputs that weaken its classifier or subspace projection mechanism, but cannot modify or disable the deployed defense code. Therefore, these adaptive attacks evaluate robustness against defense-aware malicious inputs rather than system-level code tampering. We primarily evaluate the following two attack strategies: (1) The first is the Boundary Attack, where the attacker modifies the original input with the goal of making the Harm-Sensing Classifier produce a risk score p as close as possible to 0.5. This would cause the steering strength α to approach zero, thereby disabling the steering intervention; and (2) The second is the Orthogonal Attack, where the attacker modifies the input such that the resulting raw steering vector orthogonal to the safety subspace. This would cause the projected vector v_{proj} to become a null vector. We simulate these two adaptive attacks on the LLaVA-v1.5 model by manually constraining the classifier's output range and by zeroing out the weight of the projected vector during inference. The experiments use static malicious inputs from Original, Figstep, and Query-relevant, and MM-Vet as the benign input, with detailed results in Table XI.

The results clearly indicate that while SafeSteer's defense performance degrades against these specially designed adaptive attacks, it still exhibits strong robustness. Under the Boundary and Orthogonal attacks, the average ASR rose to 0.367 and 0.375, respectively. Although this is an increase from the 0.050 under standard defense, these figures remain far below the 0.783 of the undefended baseline model. This indicates that SafeSteer's defense mechanism is effective, as it still reduces the attack success rate by over 50% even when its core components are specifically targeted, demonstrating it is not a framework that can be easily bypassed.

The test results on the MM-Vet benign tasks further highlight SafeSteer's superiority. Under the Boundary and Orthogonal attacks, the MM-Vet scores (28.5 and 28.7) are slightly lower than the 29.8 under standard defense but remained higher than the undefended baseline. This shows that SafeSteer's steering mechanism not only defends against targeted malicious attacks but also prevents disruptive attacks aimed at degrading performance.

Subspace Source Analysis. To examine the effect of the safety prefix and the semantic property of the extracted top-D subspace, we conduct a joint prefix sensitivity and subspace source analysis. For prefix-based rows, ASR and MM-Vet utility are evaluated under a prefix-matched SafeSteer pipeline, where the same prefix is used for both subspace construction and runtime steering. For the two control rows, which have no corresponding runtime prefix, we keep the default runtime prefix and replace only the SVD subspace. In addition, we report Projected Safety Gain using the semantic vocabulary sets in Table XII. Unlike ASR and utility, Projected Safety Gain is computed as a controlled subspace probe: the runtime steering vector is fixed, and only the candidate subspace used for projection is changed. It measures the induced increase in the model's preference for safety-response words over task-execution words, with larger values indicating a stronger safety-related direction.

As shown in Table XIII, all three safety-prefix settings achieve low ASR while preserving comparable benign utility, suggesting that SafeSteer is not tied to a single fixed safety prefix. Meanwhile, their performance differences indicate that the prefix choice still affects the strength of the induced steering signal and the final defense behavior. In contrast, the two non-safety prefixes lead to substantially higher ASR, showing that ordinary multimodal or instruction-following prefixes cannot directly substitute for safety-oriented prefixes in the full pipeline.

The Projected Safety Gain results further clarify why the dominant SVD directions are associated with safety-related components. Since each activation difference is computed from the same image-text input with and without a prefix, input-specific factors such as topic content, visual-text fusion difficulty, and task format are largely shared within the pair. The subsequent SVD therefore emphasizes directions that are consistently induced across paired differences, rather than directions that only reflect individual input variations. Consistent with this design, all three safety-prefix-induced subspaces yield positive gains, indicating that their top-D directions preserve components that promote safety-response tendencies. By contrast, the two tested non-safety prefixes produce weak or negative gains, suggesting that generic prefix-induced shifts do not reproduce the same semantic effect. The bottom-D SVD control yields almost no gain, further indicating that the safety-related effect is concentrated in the dominant directions rather than uniformly distributed in the difference matrix. Similarly, the no-prefix variation control fails to produce a positive gain, showing that arbitrary hidden-state differences without a safety instruction do not capture the same direction. Together, these results provide empirical evidence that the extracted top-D subspace reflects safety-related activation changes, rather than merely an ordinary task variation, or generic hidden-state variation.

Semantic Analysis of the Safety Subspace. Using the same semantic vocabulary sets, we further analyze how the original safety subspace separates the safety signal from irrelevant noise. In this experiment, we fix the original safety-prefix-induced subspace and focus on the semantic roles of different vector components within this subspace. Specifically, using

TABLE XIII
SAFETY-PREFIX SENSITIVITY AND SUBSPACE SOURCE ANALYSIS ON LLAVA-V1.5. FOR PREFIX-BASED ROWS, ASR AND UTILITY ARE EVALUATED WITH THE SAME PREFIX USED FOR BOTH SUBSPACE CONSTRUCTION AND RUNTIME STEERING. FOR CONTROL ROWS, THE DEFAULT RUNTIME PREFIX IS KEPT AND ONLY THE SVD SUBSPACE IS REPLACED. PROJECTED SAFETY GAIN IS COMPUTED BY FIXING THE RUNTIME STEERING VECTOR AND CHANGING ONLY THE CANDIDATE SUBSPACE, SERVING AS A SEMANTIC PROBE OF SAFETY-RESPONSE TENDENCY.

Subspace Construction	**Type**	**ASR ↓**	**Utility Score ↑**	**Projected Safety Gain ↑**
As an AI assistant, I will analyze and evaluate the input based on safety guidelines.	Safety-oriented prefix	0.059	29.8	0.549
I should decline unsafe or illegal requests and provide safe, high-level alternatives.	Safety-oriented prefix	0.103	29.5	0.192
As an AI helper, I will examine and assess the input based on safety protocols.	Safety-oriented prefix	0.047	29.7	0.572
Consider the visual and textual information together.	Non-safety prefix	0.425	30.4	-0.110
I will use both the visual evidence and the question to form the answer.	Non-safety prefix	0.269	28.8	-0.017
Bottom-D singular directions from the original safety-prefix SVD.	Bottom-D SVD control	0.097	27.7	0.001
Top-D directions from randomly paired differences between unprefixed hidden states.	No-prefix variation control	0.100	29.2	-0.026

the malicious test set covering the aforementioned attack types, we compute the raw steering vector v_{raw}, the projected vector v_{proj}, and the residual vector $v_{residual}$ for each sample. We then measure how each component changes the model's preference between safety-response words and task-execution words. The results are shown in Figure 3. The results clearly indicate that v_{raw} has only a slight positive impact on the safety response propensity while suppressing the task execution propensity, exhibiting a mixed and unstable initial signal that reflects its nature as a composite of both safety signal and contextual noise. After subspace projection, v_{proj} shows a stronger positive influence on safety-response tendency while maintaining suppression of task-execution tendency. In contrast, v_{residual} exhibits an opposing effect on safety-response tendency, suggesting that out-of-subspace variation may interfere with safety-oriented steering. Furthermore, this finding provides strong support for our proposed strategy of applying reverse steering for benign tasks.

C. Ablation Studies

Component-wise Analysis. To better dissect the contribution of each individual component within SafeSteer, we conduct a comprehensive ablation study on the LLava-v1.5 model. We incrementally add each component back and evaluate its resulting impact on the ASR and utility score. The results, presented in Table VII, demonstrate a clear and logical progression of performance improvement.

The study begins with the Vanilla Model, which serves as our baseline with a high ASR of 62.8%. Applying a universal Safety Prefix to all inputs drastically reduces the ASR to 21.2% but at the severe cost of collapsing the utility score to a low 9.2. The addition of the Classifier rectifies this by applying the prefix selectively, restoring utility to 27.8 while still maintaining the strong defense.

Building on this adaptive foundation, incorporating raw steering vector further lowers the ASR to 13.4%. The effectiveness of our core technical innovation is validated in the next step, which is to decompose and reconstruct the raw steering vector using a safety subspace. This not only improves the ASR to 10.9% but also boosts the utility score to 29.7, surpassing the vanilla model. Finally, the complete SafeSteer, which incorporates the linear adjustment of the strength coefficient α, achieves the best result. It drives the ASR down to a remarkable 5.9% while maintaining a high utility score of 29.8. These results demonstrate that each component of SafeSteer plays a crucial and synergistic role in achieving an strong balance between robust security and high utility.

Validation of the Harm-Sensing Classifier. To validate the effectiveness of our proposed Harm-Sensing Classifier, we design an experiment to evaluate its ability to distinguish malicious from benign inputs, using the Area Under the Receiver Operating Characteristic curve (AUROC) as the performance metric. The malicious dataset comprises two components: a multi-source set with eight types of jailbreak attacks, and a randomly selected subset of 300 samples from the portion of the JailbreakV-28k dataset not used in training. In the evaluation, we use the MM-Vet dataset as the general benign set and pair it with each of the two malicious datasets to construct two test sets. The results, presented in Table XIV, show that our classifier obtained exceptionally high AUROC scores across all models, achieving an average of over 0.995. This demonstrates the classifier's precision and reliability in identifying potential malicious inputs. Consequently, this provides a solid foundation for SafeSteer to implement adaptive intervention strategies, enabling robust defense against attacks while preserving the model's performance on benign tasks.

Impact of Steering Layer Selection. To analyze how the intervention effect varies across model depth, we evaluate

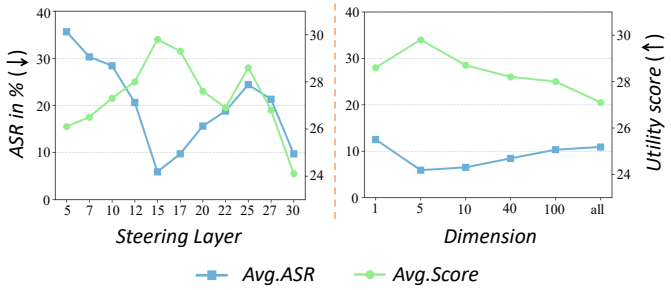


Fig. 4. Impact of the steering layer (left) and subspace dimension (right) on LLaVA-v1.5, showing the trade-off between average ASR and Utility score.

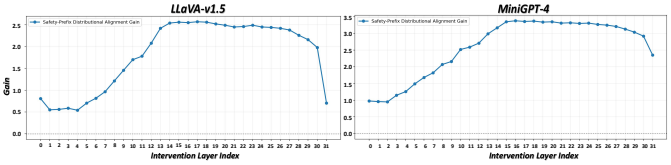


Fig. 5. Layer-wise Safety-Prefix Distributional Alignment Gain on malicious inputs. The gain measures how much steering at each layer moves the next-token distribution toward the distribution induced by an explicit safety prefix.

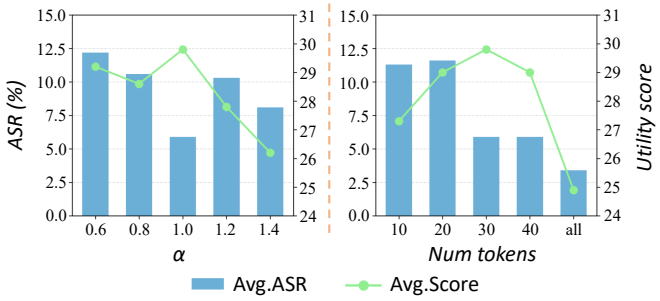


Fig. 6. Performance of SafeSteer on the LLaVA model when varying the steering strength (α, left) and the number of steered tokens (num, right).

TABLE XIV
AUROC PERFORMANCE OF THE HARM-SENSING CLASSIFIER. THE CLASSIFIER IS EVALUATED ON ITS ABILITY TO DISTINGUISH BENIGN INPUTS (FROM MM-VET) FROM TWO MALICIOUS SETS.

Model	JailbreakV-28k	Multi-Attack	Average
LLaVA-v1.5	0.997	0.996	0.997
MiniGPT-4	0.997	0.994	0.996
Qwen2.5-VL	0.996	0.993	0.995

SafeSteer by applying the steering vector at different layers. As shown in the left panel of Figure 4, middle-layer intervention achieves the best safety-utility trade-off. Specifically, applying steering near layer 15 yields the lowest ASR and the highest utility score on LLaVA-v1.5. In contrast, steering at early layers is less effective and may interfere with the fundamental fusion of visual and textual features, while intervention at final layers tends to affect surface-level generation. The sharp drop in ASR observed in our results for the final layers is not a sign of a successful defense, but rather an artifact of the model becoming incapable of producing coherent and complete outputs altogether. We further examine this layer-wise behavior from a distributional perspective using Safety-Prefix Distributional Alignment Gain on the SPA-VL [49] malicious inputs. This metric measures the reduction in divergence between the next-token distribution after layer-wise steering and the distribution induced by explicitly adding the safety prefix. Therefore, a higher gain indicates that steering at the corresponding layer can better reproduce the safety-prefix-induced behavioral shift. As shown in Figure 5, the gain is relatively low in early layers, becomes stronger in the middle layers, and decreases near the final layers. This trend suggests that the safety-related direction extracted from the safety prefix is most effectively expressed in the middle layers.

This observation is also consistent with recent findings that middle layers play a central role in forming semantic concepts [50]. Early layers mainly process low-level multimodal information, so intervention at these layers can disturb the model's basic input understanding. In contrast, late layers are closer to the output stage, where the generation direction has already been largely formed; interventions there have limited ability to reshape the model's semantic decision and may instead corrupt fluency or completeness. Middle layers provide a more suitable intervention region, where the steering vector can guide the model toward safer semantic behavior while preserving its general utility.

Analysis of Subspace Dimension. The dimensionality of the safety subspace, D, is a critical hyperparameter that governs the trade-off between capturing a clean safety signal and including extraneous noise, as shown in the right panel of Figure 4. The results demonstrate that a compact, low-dimensional subspace is optimal, with D=5 achieving the best performance on both ASR and utility scores. This finding provides a key insight: the first few principal components of the activation differences effectively capture the most critical, high-signal safety concepts necessary for robust defense. A subspace with too few dimensions ($D < 5$) lacks the representational capacity to counter diverse attacks, while increasing the dimension beyond the optimal point ($D > 5$) begins to incorporate lower-variance components that act as noise. This added noise contaminates the steering vector, impairing both defense precision and model utility. Therefore, D=5 represents the ideal balance, validating our approach of isolating a core safety signal from the broader, noisier activation space.

Analysis of Steering Strength. The left panel of Figure 6 shows the performance trade-off as we vary α_{init}. The results indicate that as α_{init} increases, the model shows better ASR and utility scores, reaching an optimal balance at $\alpha_{init} = 1.0$. Beyond this point, however, an overly aggressive steering value disrupts the model's fundamental generation process. This disruption is the reason both metrics subsequently decline. This validates our choice of using $\alpha_{init} = 1.0$ for the LLaVA-v1.5 model.

Analysis of Intervention Depth. The right panel of Figure 6 illustrates the impact of the number of steered tokens, revealing steering more tokens generally improves defense but impacts the model's performance. We observe that while steering more tokens generally leads to a stronger defensive effect, it also negatively impacts the model's performance.

TABLE XV
ABLATION STUDY ON THE SUPPRESSION FACTOR FOR THE RESIDUAL VECTOR (γ_{SUP}) ON LLAVA-V1.5.

γ_{sup}	ASR (Avg.) ↓	Utility score (Avg.)↑
0.5	0.059	29.8
0.0	0.084	28.9
1.0	0.094	27.8

TABLE XVI
ABLATION STUDY ON THE AMPLIFICATION FACTOR FOR THE PROJECTED VECTOR (γ_{amp}) ON LLAVA-V1.5.

γ_{amp}	ASR (Avg.) ↓	Utility score (Avg.) ↑
1.0	0.091	27.6
2.0	0.059	29.8
3.0	0.059	28.5
4.0	0.081	28.1

For the LLaVA model, our results indicate that steering the first 30 tokens is optimal, achieving near-optimal ASR while maximizing the utility score. This suggests a response's core direction is determined early in the generation phase, and prolonged intervention yields diminishing safety returns while increasingly degrading performance on benign tasks.

Ablation Study on the Suppression Factor (γ_{sup}). To investigate the specific role of the residual vector, we conduct an ablation study on the suppression factor, γ_{sup}, using the LLaVA-v1.5 model. In this experiment, we keep the amplification factor for the projected component fixed at $\gamma_{amp} = 2$ and compare three different settings for the suppression factor of the residual component γ_{sup}: (1) our standard setting of $\gamma_{sup} = 0.5$; (2) completely removing the residual component by setting $\gamma_{sup} = 0$; and (3) applying no suppression by setting $\gamma_{sup} = 1$. The results are presented in Table XV.

As the results show, when the residual vector is completely removed ($\gamma_{sup} = 0$), the defense remains effective, but both the ASR and utility scores are inferior to our standard setting. This suggests that while the residual vector primarily contains noise, it may also hold some benign signals that aid in contextual understanding, and its complete removal slightly impairs performance. Conversely, when no suppression is applied to the residual vector ($\gamma_{sup} = 1$), both the ASR and utility scores are the worst. This result strongly validates our core hypothesis that the residual component of the raw steering vector indeed contains substantial noise, and without suppression, this noise interferes with the precision of the steering vector and consequently impacts performance.

Ablation Study on the Amplification Factor (γ_{amp}). We also investigate the effect of the amplification factor, γ_{amp}, on the projected vector. For this experiment, we set the suppression factor γ_{sup} to its optimal value of 0.5 and evaluate a total of four different settings for γ_{amp}, as detailed in Table XVI.

The results clearly show that without any amplification (γ_{amp}=1.0), the model exhibits the poorest performance in both ASR and utility scores. This demonstrates that amplifying the purified steering vector is crucial for an effective defense. Performance peak for both metrics at γ_{amp}=2.0. However, further increases in the amplification factor (γ_{amp}=3.0 and 4.0) result in a decline in performance, with the model's utility score showing a more significant decline. This finding suggests that a moderate degree of amplification effectively strengthens the safety direction of the projected vector. In contrast, an excessive gain magnifies the signal to an unreasonable degree, thereby ultimately corrupting the original semantic integrity required to produce coherent replies.

VI. CONCLUSION

This paper addresses a critical challenge in large Vision Language Models (VLMs): how to ensure safety against jailbreak attacks while preserving model utility, all without incurring significant inference latency. We propose SafeSteer, an efficient and effective adaptive defense framework addressing fundamental limitations in VLM safety. We first identify that a core limitation of existing activation steering methods lies in their failure to separate safety signals from contextual noise. To address this, we introduce the "safety subspace" concept to effectively purify the steering vector. Building on this, SafeSteer employs a Sense-and-Intervene mechanism that combines precise steering vector with adaptive strength adjustment, achieving a favorable balance among safety, utility, and efficiency. Extensive experiments demonstrate that this method significantly reduces the attack success rate while maintaining, and even enhancing, the model's performance with minimal computational overhead. We hope that our work on adaptive, precise steering will inspire future research into more accurate and reliable VLM safety methods.

VII. ACKNOWLEDGMENT

Xiyu Zeng, Shuchao Pang, Liming Lu, Haotian Zhu and Enguang Liu are supported by the National Key Research and Development Program of China under (Grant No.2023YFB2703900) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No.KYCX24_0723 and Grant No.KYCX25_0814).

REFERENCES

[1] J. Chen, D. Zhu, X. Shen, X. Li, Z. Liu, P. Zhang, R. Krishnamoorthi, V. Chandra, Y. Xiong, and M. Elhoseiny, "Minigpt-v2: large language model as a unified interface for vision-language multi-task learning," *arXiv preprint arXiv:2310.09478*, 2023.

[2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[3] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[4] Z. Ying, A. Liu, T. Zhang, Z. Yu, S. Liang, X. Liu, and D. Tao, "Jailbreak vision language models via bi-modal adversarial prompt," *arXiv preprint arXiv:2406.04031*, 2024.

[5] Z. Ying, S. Wu, R. Hao, P. Ying, S. Sun, P. Chen, J. Chen, H. Du, K. Shen, S. Wu *et al.*, "Pushing the limits of safety: A technical report on the atlas challenge 2025," *arXiv preprint arXiv:2506.12430*, 2025.

[6] S. Liang, J. Liu, J. Zhai, T. Fang, R. Tu, A. Liu, X. Cao, and D. Tao, "T2vshield: Model-agnostic jailbreak defense for text-to-video models," *arXiv preprint arXiv:2504.15512*, 2025.

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

[8] X. Qi, K. Huang, A. Panda, P. Henderson, M. Wang, and P. Mittal, "Visual adversarial examples jailbreak aligned large language models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 19, 2024, pp. 21 527–21 536.

[9] Y. Gong, D. Ran, J. Liu, C. Wang, T. Cong, A. Wang, S. Duan, and X. Wang, "Figstep: Jailbreaking large vision-language models via typographic visual prompts," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 22, 2025, pp. 23 951–23 959.

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

[11] Y. Zong, O. Bohdal, T. Yu, Y. Yang, and T. Hospedales, "Safety fine-tuning at (almost) no cost: A baseline for vision large language models," *arXiv preprint arXiv:2402.02207*, 2024.

[12] Y. Zhang, L. Chen, G. Zheng, Y. Gao, R. Zheng, J. Fu, Z. Yin, S. Jin, Y. Qiao, X. Huang *et al.*, "Spa-vl: A comprehensive safety preference alignment dataset for vision language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 19 867–19 878.

[13] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[14] N. Panickssery, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. M. Turner, "Steering llama 2 via contrastive activation addition," *arXiv preprint arXiv:2312.06681*, 2023.

[15] P. Wang, D. Zhang, L. Li, C. Tan, X. Wang, K. Ren, B. Jiang, and X. Qiu, "Inferaligner: Inference-time alignment for harmlessness through cross-model guidance," *arXiv preprint arXiv:2401.11206*, 2024.

[16] Z. Cao, Y. Yang, and H. Zhao, "Scans: Mitigating the exaggerated safety for llms via safety-conscious activation steering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 22, 2025, pp. 23 523–23 531.

[17] W. Zhao, J. Guo, Y. Hu, Y. Deng, A. Zhang, X. Sui, X. Han, Y. Zhao, B. Qin, T.-S. Chua *et al.*, "Adasteer: Your aligned llm is inherently an adaptive jailbreak defender," *arXiv preprint arXiv:2504.09466*, 2025.

[18] X. Guo, F. Yu, H. Zhang, L. Qin, and B. Hu, "Cold-attack: Jailbreaking llms with stealthiness and controllability," *arXiv preprint arXiv:2402.08679*, 2024.

[19] X. Liu, N. Xu, M. Chen, and C. Xiao, "Autodan: Generating stealthy jailbreak prompts on aligned large language models," *arXiv preprint arXiv:2310.04451*, 2023.

[20] J. Yu, X. Lin, Z. Yu, and X. Xing, "Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts," *arXiv preprint arXiv:2309.10253*, 2023.

[21] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[22] N. Carlini, M. Nasr, C. A. Choquette-Choo, M. Jagielski, I. Gao, P. W. W. Koh, D. Ippolito, F. Tramer, and L. Schmidt, "Are aligned neural networks adversarially aligned?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 61 478–61 500, 2023.

[23] E. Bagdasaryan, T.-Y. Hsieh, B. Nassi, and V. Shmatikov, "Abusing images and sounds for indirect instruction injection in multi-modal llms," *arXiv preprint arXiv:2307.10490*, 2023.

[24] Y. Zhao, T. Pang, C. Du, X. Yang, C. Li, N.-M. M. Cheung, and M. Lin, "On evaluating adversarial robustness of large vision-language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 54 111–54 138, 2023.

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

[26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[27] Y. Chen, K. Sikka, M. Cogswell, H. Ji, and A. Divakaran, "Dress: Instructing large vision-language models to align and interact with humans via natural language feedback," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 239–14 250.

[28] M. Li, L. Li, Y. Yin, M. Ahmed, Z. Liu, and Q. Liu, "Red teaming visual language models," *arXiv preprint arXiv:2401.12915*, 2024.

[29] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[30] Z. Sun, S. Shen, S. Cao, H. Liu, C. Li, Y. Shen, C. Gan, L.-Y. Gui, Y.-X. Wang, Y. Yang *et al.*, "Aligning large multimodal models with factually augmented rlhf," *arXiv preprint arXiv:2309.14525*, 2023.

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

[32] Y. Ding, B. Li, and R. Zhang, "Eta: Evaluating then aligning safety of vision language models at inference time," *arXiv preprint arXiv:2410.06625*, 2024.

[33] X. Zhang, C. Zhang, T. Li, Y. Huang, X. Jia, X. Xie, Y. Liu, and C. Shen, "A mutation-based method for multi-modal jailbreaking attack detection," *CoRR*, 2023.

[34] C. Burns, H. Ye, D. Klein, and J. Steinhardt, "Discovering latent knowledge in language models without supervision," *arXiv preprint arXiv:2212.03827*, 2022.

[35] L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodolà, "Relative representations enable zero-shot latent space communication," *arXiv preprint arXiv:2209.15430*, 2022.

[36] H. Wang, G. Wang, and H. Zhang, "Steering away from harm: An adaptive approach to defending vision language model against jailbreaks," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 29 947–29 957.

[37] C. Lu, J. Gallagher, J. Michala, K. Fish, and J. Lindsey, "The assistant axis: Situating and stabilizing the default persona of language models," *arXiv preprint arXiv:2601.10387*, 2026.

[38] L. Sheng, C. Shen, W. Zhao, J. Fang, X. Liu, Z. Liang, X. Wang, A. Zhang, and T.-S. Chua, "Alphasteer: Learning refusal steering with principled null-space constraint," *arXiv preprint arXiv:2506.07022*, 2025.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[41] S. Zhao, R. Duan, F. Wang, C. Chen, C. Kang, J. Tao, Y. Chen, H. Xue, and X. Wei, "Jailbreaking multimodal large language models via shuffle inconsistency," *arXiv preprint arXiv:2501.04931*, 2025.

[42] Z. Ying, A. Liu, T. Zhang, Z. Yu, S. Liang, X. Liu, and D. Tao, "Jailbreak vision language models via bi-modal adversarial prompt," *IEEE Transactions on Information Forensics and Security*, 2025.

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

[44] Y. Li, H. Guo, K. Zhou, W. X. Zhao, and J.-R. Wen, "Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 174–189.

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

[46] M. He, Y. Liu, B. Wu, J. Yuan, Y. Wang, T. Huang, and B. Zhao, "Efficient multimodal learning from data-centric perspective," *arXiv preprint arXiv:2402.11530*, 2024.

[47] W. Luo, S. Ma, X. Liu, X. Guo, and C. Xiao, "Jailbreakv: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks," *arXiv preprint arXiv:2404.03027*, 2024.

[48] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[49] Y. Zhang, L. Chen, G. Zheng, Y. Gao, R. Zheng, J. Fu, Z. Yin, S. Jin, Y. Qiao, X. Huang, F. Zhao, T. Gui, and J. Shao, "Spa-vl: A comprehensive safety preference alignment dataset for vision language model," 2024.

[50] S. Wang, Y. Zhang, Y. Zhu, J. Li, Z. Wang, Y. Liu, and X. Ji, "Towards understanding how knowledge evolves in large vision-language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 29 858–29 868.

Shuchao Pang (Member, IEEE) received a Ph.D. degree in computer science from Macquarie University, Sydney, NSW, Australia, in 2020. He is currently a Professor at the Nanjing University of Science and Technology, Nanjing, China, and an Honorary Research Fellow with Macquarie University, Sydney, NSW, Australia. Before that, he was a Research Associate with the University of New South Wales, Sydney, and a Postdoctoral Researcher with Ingham Institute for Applied Medical Research, Sydney. His research interests include AI security, data security and privacy protection, LLMs and agent security.

Xiyu Zeng received the bachelor's degree from the School of Computer Science, Hunan University of Technology and Business, Changsha, China, in 2024. He is currently pursuing the master's degree with the School of Cyber Science and Engineering, Nanjing University of Science and Technology. His research interests include vision-language model (VLM) security.

Siyuan Liang received the Ph.D. degree in Cyberspace Security from the University of the Chinese Academy of Sciences. She is currently a Research Fellow with the College of Computing & Data Science at Nanyang Technological University. Her research interests include adversarial machine learning and computer vision, with a focus on developing robust AI models for secure visual perception.

Liming Lu received the bachelor's degree from the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Nanjing University of Science and Technology. His research interests include knowledge distillation and model security.

Haotian Zhu is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Nanjing University of Science and Technology. He received the bachelor's degree from the School of Computing and Artificial Intelligence, Southwest Jiaotong University. His research interests include AI application security and privacy protection.

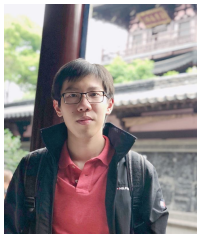
Chuanting Zhang (Senior Member, IEEE) received the Ph.D. degree from Shandong University, China, in 2019. He is currently an Associate Professor with the Institute of Intelligent Communication Technologies, Shandong University. Previously, he was a Senior Research Associate at the University of Bristol, U.K., and a Post-Doctoral Fellow at King Abdullah University of Science and Technology (KAUST), Saudi Arabia. His research focuses on spatial-temporal data analysis, federated learning, and the intersection of AI and networks. His accolades include the IEEE ICCT Young Scientist Award and the IEEE SmartData Best Paper Award.

Enguang Liu received the bachelor's degree from Henan Agricultural University, Zhengzhou, China, in 2022. He is currently pursuing the master's degree with the School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include embodied AI security.

Liang Zhang received the Ph.D. degree in electrical and computer engineering from King Abdullah University of Science and Technology, Saudi Arabia, in 2022. She was senior AI research engineer and system architect in Huawei technologies co. ltd, and is currently working as algorithm specialist in Jack company. Her expertise lies in UAV, llm, embodied AI and reinforcement learning. She is serving as reviewer for top journals such as TMC, TWC, ComMag, and TCOM etc.

Basem Shihada (M'04–SM'12, IEEE) received the Ph.D. degree in Computer Science from the University of Waterloo, Canada, in 2007. In 2008, he joined King Abdullah University of Science and Technology (KAUST) as a Founding Faculty member. His research focuses on cutting-edge wireless systems, including intelligent, underwater, molecular, and non-terrestrial networks. Notable innovations include Aqua-Fi (underwater Wi-Fi), Sun-Fi, and breath-based communication. He has received multiple best paper awards and published in prestigious journals like Nature Electronics and IEEE Transactions. In 2023, he received the exemplary editor award from IEEE Communications Letters.

Yongbin Zhou (Member, IEEE) is currently a full Professor with the Nanjing University of Science and Technology, Nanjing, China, and also a Visiting Research Fellow with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China. Prior to that, he was a full professor of the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences and a professor with the School of Cyber Security, University of Chinese Academy of Sciences. His main research interests include theories and technologies of cryptology, network and information security.



Minhui Xue is a Senior Research Scientist (lead of AI Security sub-team) at CSIRO's Data61, Australia. His current research interests are machine learning security and privacy, system and software security, and Internet measurement. He is the recipient of the ACM CCS Best Paper Award Runner-Up, ACM SIGSOFT distinguished paper award, Best Student Paper Award, and the IEEE best paper award, and his work has been featured in the mainstream press, including The New York Times, Science Daily, PR Newswire, Yahoo, The Australian Financial Review, and The Courier. He currently serves on the Program Committees of IEEE Symposium on Security and Privacy (Oakland) 2023, ACM CCS 2023, USENIX Security 2023, NDSS 2023, ACM/IEEE ICSE 2023, and ACM/IEEE FSE 2023. He is a member of both ACM and IEEE.