LSSF: Safety Alignment for Large Language Models through Low-Rank Safety Subspace Fusion

Guanghao Zhou[1]*, **Panjia Qiu**[1]*, **Cen Chen** [1]†, **Hongyu Li**[2],
Mingyuan Chu[2], **Xin Zhang**[2], **Jun Zhou**[2],
[1]East China Normal University [2]Ant Group
{ghzhou, panjiaqiu}@stu.ecnu.edu.cn, cenchen@dase.ecnu.edu.cn
zhiyuan.lhy@antgroup.com, jszjg1991@gmail.com
evan.zx@ant-intl.com, jun.zhoujun@antgroup.com

Abstract

The safety mechanisms of large language models (LLMs) exhibit notable fragility, as even fine-tuning on datasets without harmful content may still undermine their safety capabilities. Meanwhile, existing safety alignment methods predominantly rely on the fine-tuning process, which inadvertently leads to the increased complexity and computational resources required. To address these issues, we introduce LSSF, a novel safety re-alignment framework with Low-Rank Safety Subspace Fusison. Our proposed method exploits the low-rank characteristics of safety information in LLMs by constructing a low-rank projection matrix to extract the principal components of safety vectors. Notably, this projection matrix represents the low-rank safety subspace of the LLMs, which we have observed to remain stable during fine-tuning process and is isolated from the model's general capabilities. These principal components are used to effectively restore safety alignment when combined with fine-tuned LLMs through linear arithmetic. Additionally, to account for the varying encoding densities of safety information across different layers of LLMs, we propose a novel metric called safety singular value entropy. This metric quantifies the encoding density and allows for the dynamic computation of the safety-critical rank for each safety vector. Extensive experiments demonstrate that our proposed post-hoc alignment method can effectively restore the safety alignment of fine-tuned models with minimal impact on their performance in downstream tasks.

1 Introduction

In recent years, as the capabilities of large language models (LLMs) have improved significantly (Achiam et al., 2023; AI@Meta, 2024), a growing amount of research has focused on enhancing their safety to prevent unsafe responses that conflict with human values (Christiano et al., 2017; Yuan et al., 2023). Numerous studies have revealed that aligned LLMs exhibit surprising safety vulnerabilities after fine-tuning (Qi et al., 2023; Zhan et al., 2024; Fan et al., 2025). The safety of these models can be significantly compromised when updated with a small amount of maliciously crafted or even benign data. To address this issue, existing studies (Zong et al., 2024; Huang et al., 2024) have primarily focused on ensuring model safety and consistency by aligning the model during the fine-tuning phase. However, these methods not only increase the complexity of the training process and require additional computational resources, but they also potentially inhibit the model's general capabilities. Recent work (Ilharco et al., 2022) demonstrates that adding or subtracting task vectors, i.e., directional vectors corresponding to specific tasks within a model, can enhance or reduce the model's performance on those tasks. Building on this, RESTA (Bhardwaj et al., 2024) introduces a post-hoc alignment method that restores the safety of compromised models by performing an arithmetic combination with safety vectors. Nonetheless, as the safety vectors contain elements that inhibit general capabilities, the integrated model may experience an inevitable reduction in its general abilities.

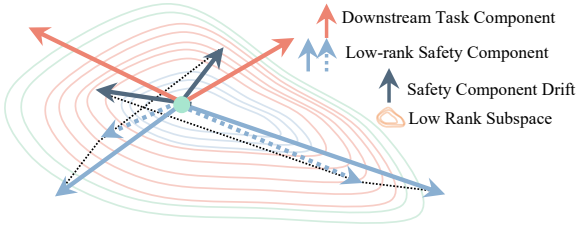


Figure 1: Illustration of using low-rank safety principal components to restore model safety alignment, where all safety components share the same low-rank subspace.

Recent studies (Sun et al., 2023; Wei et al., 2024) have indicated that the safety regions in LLMs are

*Completed during internship at Ant Group
†Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30621–30638

isolated and sparse at the rank level, distinct from the directions of the models' general capabilities. Our experiments in Section 4.2.2 also confirm that the safety drift directions of safety vectors likewise exhibit low-rank properties and share a common low-rank safety subspace with LLMs.

Based on the above insights, we introduce LSSF, a novel safety re-alignment framework with Low-Rank Safety Subspace Fusion. Specifically, we perform low-rank orthogonal matrix decomposition on the activations of the safety-aligned LLMs and construct a projection matrix to extract the low-rank principal components of the corresponding safety vectors. As illustrated in Figure 1, when the critical safety directions of a fine-tuned model drift, we can effectively rectify this deviation by applying linear arithmetic to the low-rank principal components of the safety vectors.

Moreover, we propose a safety singular value entropy information density quantification method inspired by Shannon entropy (Shannon, 1948). Previous work on assessing brittleness of safety alignment (Wei et al., 2024) reveals that different linear layers in LLMs encode safety and general capabilities to varying degrees. This highlights the necessity of determining the pruning rank of the corresponding safety vector based on the density of safety information encoded in the weight matrix. Our singular value entropy considers both the absolute magnitudes of the singular values and their relative distribution. By analyzing the proportion of singular value entropy, we can effectively control information loss when truncating the rank. Extensive experiments on Qwen2.5-7B-Instruct (Team, 2024) and Llama3.1-8B-Instruct (AI@Meta, 2024) demonstrate that our proposed LSSF can restore safety alignment with minimal impact on the downstream task performance of their fine-tuned models.

Our contributions are summarized as follows:

- We proposed the utilization of a projection matrix to extract the low-rank principal components of the safety vector, enabling the safety realignment of the fine-tuned LLMs within this low-rank subspace.
- We proposed a novel safety singular value entropy-based information density quantification method that effectively assesses the safety information encoding density within the linear layer and assists in the determination of the appropriate pruning rank for the safety vector.
- We performed comprehensive experiments on various LLMs, which demonstrated that our method can effectively restores their safety alignment without significantly compromising the downstream task performance.

2 Related Work

Safety Realignment. Pre-trained LLMs are typically enhanced for specific downstream tasks through a process known as supervised fine-tuning, which often involves full fine-tuning (Howard and Ruder, 2018) and parameter-efficient fine-tuning (Hu et al., 2021; Ben Zaken et al., 2022). Even LLMs with strong initial safety alignment can be manipulated to produce harmful content during the fine-tuning process (Bianchi et al., 2023; He et al., 2024). Some studies (Dai et al., 2023; Huang et al., 2024; Bianchi et al., 2023) focus on ensuring safety realignment during the fine-tuning of LLMs, which undoubtedly increases the complexity of this process. RESTA (Bhardwaj et al., 2024) employs the direct arithmetic combination of safety vectors to the weights of fine-tuned models. Since safety vectors include general capability-suppressing components, there is a certain impact on the model's performance on downstream tasks.

Low-Rank Compression and Pruning. Unstructured pruning techniques aim to establish criteria based on weight magnitude, activations, or network gradients to remove individual weights from the network (Cao et al., 2021; Guo et al., 2021; Zhang et al., 2024). Low-rank compression techniques are similar to structured pruning methods, focusing on identifying important structured sub-networks (Sun et al., 2023; Frantar and Alistarh, 2023; Wang et al., 2024). ActSVD (Wei et al., 2024) extracts the safety-critical rank of LLMs through singular value decomposition (SVD) of stacked activations and demonstrates its low-rank nature. We extend the low-rank pruning of LLMs to safety vectors to extract principal components of the corresponding low-rank secure subspace.

Model fusion. Current fusion methods for LLMs generally fall into three categories: geometric (Shoemake, 1985), pruning (Yadav et al., 2023), and arithmetic (Xiao et al., 2024). As a geometry-based approach, Model Stock (Jang et al., 2024) considers the geometric properties in the weight space. Pruning-based methods such as Breadcrumbs (Davari and Belilovsky, 2024) and DARE (Yu et al., 2023) eliminate interference among mul-

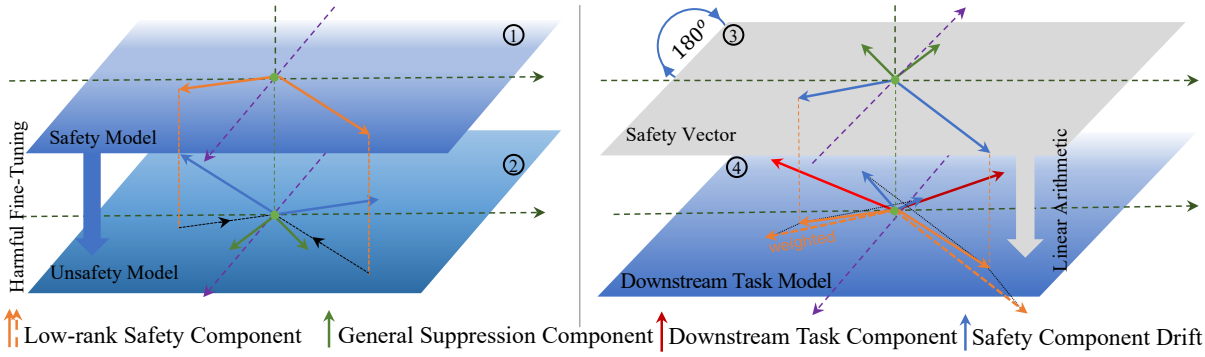


Figure 2: Overview of our safety re-alignment framework. Step I: ① → ②, obtain an unsafe model through unsafe fine-tuning. Step II: ② → ③, invert the delta parameters to derive the safety vector. Step III: ③ → ④, construct a low-rank projection matrix to extract the safety principal components of the safety vector and perform linear arithmetic with the downstream fine-tuned model to restore its safety alignment.

tiple models by removing redundant parameters. Arithmetic-based methods include Average Merging (Wortsman et al., 2022) and Task Arithmetic (Bhardwaj et al., 2024). The former merges models by averaging parameters, while the latter introduces task vectors and uses scaling terms to adjust the importance of different models. We extend task arithmetic to safety fine-tuning tasks and employ low-rank pruning methods to mitigate its impact on the LLMs' downstream task performance.

3 Methodology

Our objective is to restore the safety of LLMs through post-hoc alignment. Figure 2 illustrates our safety re-alignment framework. Starting from an unsafe model derived via harmful fine-tuning, we compute the inverted safety vector and extract its low-rank safety components. These are then integrated into the downstream task model through linear arithmetic to restore safety alignment. In Section 3.1, we introduce the safety vectors of LLMs, Sections 3.2 and 3.3 provide a detailed explanation of how to extract low-rank safety components and Section 3.4 explains how task arithmetic restores the safety of fine-tuned models.

3.1 Safety Vector

Safety vector is derived from the delta parameters when transitioning from the unsafe base model to the safety-aligned model, formulated as:

$$\boldsymbol{\theta}_{\text{safe}} = \boldsymbol{\theta}_{\text{unsafe}} + \boldsymbol{\delta}_{\text{safe}}, \tag{1}$$

where $\boldsymbol{\theta}_{\text{safe}}$ denotes the parameters of the safety-aligned model, while $\boldsymbol{\theta}_{\text{unsafe}}$ refers to the parameters of the unsafe model and $\boldsymbol{\delta}_{\text{safe}}$ represents the safety vector obtained through the alignment process. However, compromising safety guardrails is significantly easier than safety alignment, as the former only requires fine-tuning on a small amount of toxic data. As shown in Figure 2 Step I, we use the toxic dataset $\mathcal{D}_{\text{unsafe}} = \{(x_i, y_i) \mid i = 1, \ldots, N\}$ to perform supervised fine-tuning on $\boldsymbol{\theta}_{\text{safe}}$ to obtain the inverse safety vector $-\boldsymbol{\delta}_{\text{safe}}$,where x_i represents harmful queries, and y_i denotes affirmative responses to these harmful queries.

As shown in Figure 2 ②, $-\boldsymbol{\delta}_{\text{safe}}$ consists of two main components. The first is the drift of the low-rank safety component in the opposite direction of safety, which can be extracted through low-rank decomposition due to its low-rank nature. The second is the general suppression component, which impairs the model's performance on general tasks.

3.2 Low-rank Orthogonal Decomposition

Motivated by ActSVD (Wei et al., 2024), we perform singular value decomposition on the linear layer activations of $\boldsymbol{\theta}_{\text{base}}$ and use the left singular vectors to construct a low-rank projection matrix. First, we construct a calibration dataset $\mathcal{D}_{\text{anchor}} = \{(x'_i, y'_i) | i = 1, \cdots, N'\}$, where x'_i represents harmful queries and y'_i represents safe negative responses. For any linear layer weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we obtain the corresponding input matrix $\widehat{X} \in \mathbb{R}^{d_{\text{in}} \times n}$ from $\mathcal{D}_{\text{anchor}}$. The objective of the low-rank decomposition of $\boldsymbol{\theta}_{\text{base}}$ is to achieve a low-rank approximation of W while maintaining its safety performance. Specifically, we seek a rank-r low-rank matrix $\widehat{W}$ that minimizes the Frobenius norm of the output changes, as described by the

following formula:

$$\widehat{W} = \underset{\text{rank}(\widehat{W}) \leq r}{\arg\min} \|W\widehat{X} - \widehat{W}\widehat{X}\|_F^2, \quad (2)$$

where the optimal low-rank matrix $\widehat{W}$ shares the same low-rank subspace as the safe vector δ_{safe}. The proof is provided in Appendix A.

To eliminate the sensitivity differences of neurons in the linear layer to different texts in the anchor dataset, we normalize the activation matrix $Z = W\widehat{X} = [z_{ij}]_{d_{in} \times n}$ as follows:

$$\tilde{z}_{ij} = \frac{z_{ij} - \mu_j}{\delta_j}, \quad (3)$$

where $z_{i,j}$ represents the activation value at (i, j), μ_j and δ_j denote the mean and standard deviation of each column, respectively. We perform a low-rank matrix decomposition on the standardized activation matrix $\tilde{Z}$ using SVD:

$$USV^\top \approx \tilde{Z}, \quad (4)$$

where $S = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$ is the singular value diagonal matrix and $U \in \mathbb{R}^{d_{\text{out}} \times n}$ is an orthogonal matrix composed of the top n left singular vectors. Due to the slow speed of SVD for large matrices, we follow the method from Halko et al. (2011), which uses a randomized algorithm to efficiently calculate approximate solutions of SVD.

3.3 Low-Rank Projection Matrix

For matrix $\tilde{Z}$ in Equation 3, the square of the Frobenius norm can be expressed as $\|\tilde{Z}\|_F^2 = \sum_{i,j} |\tilde{z}_{i,j}|^2$. According to the SVD, we have $\|\tilde{Z}\|_F^2 = \sum_{i=1}^n \sigma_i^2$, where σ_i represents the i-th singular value, the proof can be found in Appendix B. This indicates that the energy of the matrix $\tilde{Z}$ is equal to the sum of the squares of its singular values, which can be interpreted as a measure of the overall complexity or information content of the matrix. Therefore, we quantify each principal component's contribution to the total information content using its squared singular value, defined as:

$$p_i = \frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2}, \quad (5)$$

where p_i represents the information contribution of the i-th principal component. Singular value entropy is used to evaluate the complexity and information content of matrix, taking into account not only the absolute magnitudes of the squared singular values but also their relative distribution. The formula for calculating singular value entropy is as follows:

$$H_\rho = -\sum_{i=1}^{\rho} \frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2} \log \left(\frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2} \right), \quad (6)$$

where H_ρ represents the entropy of the singular values of the top ρ ranks. We use the information retention threshold η to determine the rank preserved by the orthogonal projection:

$$\frac{H_r}{H_n} > \eta, \quad (7)$$

where r represents the rank to be retained. Accordingly, we construct a low-rank projection matrix:

$$P^{(r)} = U^{(r)} \left(U^{(r)} \right)^\top, \quad (8)$$

where $P^{(r)}$ denotes the orthogonal projection matrix onto the r most significant left singular subspaces, where $\text{rank}(P^{(r)}) = r$. The proof is provided in Appendix C.

To balance the utility and safety of LLMs, we apply the scaling factor α to the singular vectors to enhance or diminish their drift in the corresponding safety directions. The scaling factor α_i corresponding to the singular vector $u_i \in U^{(r)}$ is calculated as follows:

$$\alpha_i = 1 + (\alpha_1 - 1) \times \frac{\sigma_i - \sigma_r}{\sigma_1 - \sigma_r}, \quad (9)$$

where α_1 represents the weighting factor for the singular vector corresponding to the largest singular value, and subsequent weights decreasing proportionally. This formulation effectively enhances the weights of singular vectors corresponding to larger singular values in a proportional manner. The weighted singular vectors are:

$$U'^{(r)} = (u_1, \cdots, u_r) \begin{pmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_r \end{pmatrix}. \quad (10)$$

Therefore, the final low-rank projection matrix corresponding to the safety vector is:

$$P'^{(r)} = U'^{(r)} \left(U'^{(r)} \right)^\top. \quad (11)$$

3.4 Linear Arithmetic

Let $\boldsymbol{\theta}_{\text{DST}}$ represent the model parameters of $\boldsymbol{\theta}_{\text{safe}}$ after fine-tuning for downstream task datasets. The fine-tuning process may compromise the safety guardrails of LLMs and it could be expressed as:

$$\begin{aligned}\boldsymbol{\theta}_{\text{DST}} &= \boldsymbol{\theta}_{\text{safe}} + \boldsymbol{\delta}_{\text{DST}} \\ &= \boldsymbol{\theta}_{\text{safe}} + \boldsymbol{\tau}_{\text{DST}} - \boldsymbol{\tau}_{\text{safe}} + \hat{\boldsymbol{\tau}}_{\text{DST}}, \end{aligned} \tag{12}$$

were $\boldsymbol{\delta}_{\text{DST}}$ represents the delta parameters obtained from the SFT of $\boldsymbol{\theta}_{\text{safe}}$. We decompose $\boldsymbol{\delta}_{\text{DST}}$ into the desired downstream task direction offset $\boldsymbol{\tau}_{\text{DST}}$, an undesired offset in the safety direction $-\boldsymbol{\tau}_{\text{safe}}$ and a redundant shift in other directions $\hat{\boldsymbol{\tau}}_{\text{DST}}$. Our objective is to add a low-rank safety component $\boldsymbol{\tau}'_{\text{safe}}$ to $\boldsymbol{\delta}_{\text{DST}}$ to counteract the $-\boldsymbol{\tau}_{\text{safe}}$ offset while minimizing impacts on shifts in other directions. Our objective could be expressed as:

$$\begin{aligned}\boldsymbol{\theta}'_{\text{DST}} &= \boldsymbol{\theta}_{\text{safe}} + \boldsymbol{\delta}_{\text{DST}} + \alpha P'^{(r)} \boldsymbol{\delta}_{\text{safe}} \\ &= \boldsymbol{\theta}_{\text{safe}} + \boldsymbol{\tau}_{\text{DST}} - \boldsymbol{\tau}_{\text{safe}} + \alpha \boldsymbol{\tau}'_{\text{safe}} + \hat{\boldsymbol{\tau}} \\ &\approx \boldsymbol{\delta} + \boldsymbol{\tau}_{\text{DST}} + \hat{\boldsymbol{\tau}}, \end{aligned} \tag{13}$$

where $\hat{\boldsymbol{\tau}}$ denotes the drift in the redundant direction, which does not significantly affect $\boldsymbol{\tau}_{\text{DST}}$. Additionally, $\boldsymbol{\tau}_{\text{safe}}$ and $\boldsymbol{\tau}'_{\text{safe}}$ share the same low-rank subspace, as detailed in Section 3.3.

4 Experiment

4.1 Experiment Setup

4.1.1 Models under testing

We perform experiments on two LLMs: Qwen2.5-7B-Instruct (Qwen2.5) (Team, 2024) and Llama3.1-8B-Instruct (Llama3.1) (AI@Meta, 2024). Due to their strong safety and instruction-following capabilities, we adopt them as safety models and base models for downstream fine-tuning. In Appendix G, we selected Qwen2.5-3B-Instruct and Qwen2.5-14B-Instruct to validate the robustness of our method across models with varying parameters.

4.1.2 Baselines

We classify baseline methods into two categories: alignment during supervised fine-tuning (SFT) and post-hoc alignment. The baselines include: Non-Alignment SFT (NA-SFT), which does not enforce safety alignment; VLGuard (Zong et al., 2024) and Lisa (Huang et al., 2024), which implement safe alignment during the fine-tuning process; RESTA (Bhardwaj et al., 2024), which applies safety alignment after fine-tuning. For detailed information on the baselines and specific experimental settings, please refer to Appendix D.1.

4.1.3 Datasets

Calibration Dataset. We construct a calibration dataset to obtain the safety-critical ranks of the safety model. To ensure the harmfulness of the queries, we selected samples from PKU-SafeRLHF (Ji et al., 2024) where both the *accept* and *reject* responses were labeled as *unsafe*. As demonstrated in Appendix E, our method is robust to the number of calibration data samples. Consistent with (Wei et al., 2024), we ultimately chose a total of 128 such samples. These harmful queries are then input into the safety model to collect safe refusal responses. Based on the pairs of harmful queries and safe responses, we construct a calibration dataset.

Downstream Fine-Tuning Dataset. To better demonstrate the effectiveness of our method in downstream task fine-tuning experiments, we established two distinct scenarios. The first scenario involves LoRA fine-tuning, where we utilize AG's News and Yahoo Answers (Zhang et al., 2015) for multi-class classification tasks. The second scenario is full fine-tuning, which performs poorly on LoRA since it not only requires knowledge infusion but also focuses on dialogue-based question answering. In this scenario, we emphasize the medical knowledge question-answering dataset[1], specifically designed for question-answering and text generation tasks. For detailed construction of the SFT datasets, please refer to Appendix D.2.

4.1.4 Safety Vector Calculation

We calculate safety vectors by determining the offset between the aligned models and its unaligned counterparts. To construct a harmful fine-tuning dataset, we randomly select 500 harmful queries and their corresponding harmful responses labeled as *unsafe* from PKU-SafeRLHF (Ji et al., 2024). The dataset is subsequently utilized to perform SFT for 3 epochs on Qwen2.5 and Llama3.1 to get new models that compromise safety guardrails.

4.1.5 Evaluation Metric

Measuring utility. For the text classification task, we assess the model's classification accuracy (ACC). For the text generation task, we employ BLEU (Papineni et al., 2002) ROUGE-L (Lin, 2004) as evaluation metrics. Details of each metric could be found in the Appendix D.3.

Measuring Safety. We use three datasets to evaluate the safety of the model: AdvBench (Zou et al.,

[1] https://github.com/Toyhom/Chinese-medical-dialogue-data

Methods	Qwen2.5-7B-Instruct				Llama3.1-8B-Instruct			
	ACC ↑	AdvBench ↑	HarmfulQA ↑	CATQA ↑	ACC ↑	AdvBench ↑	HarmfulQA ↑	CATQA ↑
NA-SFT	0.91	0.09	0.50	0.19	**0.86**	0.04	0.49	0.12
VLGuard	**0.92**	0.78	0.54	0.13	0.85	0.92	0.56	0.21
RESTA	**0.92**	0.99	**0.98**	0.92	0.69	0.51	0.91	0.72
Lisa	0.90	0.99	**0.98**	**0.94**	0.80	0.96	0.92	0.96
Ours	**0.92**	**1.00**	**0.98**	0.93	0.85	**1.00**	**1.00**	**1.00**

Table 1: Performance of different safety alignment methods in the AG's News LoRA SFT Task. ***ACC*** represents the classification accuracy. ***AdvBench***, ***HarmfulQA***, and ***CATQA*** denote refusal rates for the corresponding datasets.

Methods	Qwen2.5-7B-Instruct				Llama3.1-8B-Instruct			
	ACC ↑	AdvBench ↑	HarmfulQA ↑	CATQA ↑	ACC ↑	AdvBench ↑	HarmfulQA ↑	CATQA ↑
NA-SFT	**0.68**	0.20	0.52	0.12	0.63	0.04	0.49	0.12
VLGuard	**0.68**	0.96	0.58	0.21	**0.64**	0.87	0.51	0.15
RESTA	0.63	0.94	0.92	0.81	0.25	0.33	0.86	0.62
Lisa	0.67	0.98	0.97	0.93	0.45	0.89	0.97	0.89
Ours	**0.68**	**1.00**	**0.99**	**0.99**	**0.64**	**1.00**	**0.99**	**0.99**

Table 2: Performance of different safety alignment methods in the Yahoo Answers LoRA SFT Task . ***ACC*** represents the classification accuracy. ***AdvBench***, ***HarmfulQA***, and ***CATQA*** denote refusal rates for the corresponding datasets.

Method	Qwen2.5-7B-Instruct					Llama3.1-8B-Instruct				
	BLUE ↑	Rouge-L ↑	AdvBench ↑	HarmfulQA ↑	CATQA ↑	BLUE ↑	Rouge-L ↑	AdvBench ↑	HarmfulQA ↑	CATQA ↑
NA-SFT	**0.44**	**0.52**	0.21	0.54	0.15	**0.44**	**0.53**	0.11	0.49	0.12
VLGuard	**0.44**	**0.52**	0.88	0.62	0.21	0.42	0.50	0.98	0.62	0.34
RESTA	0.34	0.43	**0.99**	0.91	**0.95**	0.18	0.27	0.91	**0.99**	0.97
Lisa	0.31	0.41	**0.99**	0.91	**0.95**	0.21	0.31	0.94	**0.99**	0.98
Ours	0.42	0.50	**0.99**	**0.94**	**0.95**	0.42	0.50	**0.99**	**0.99**	**0.99**

Table 3: Performance of different safety alignment methods in the Medical QA Full SFT Task. ***BLUE*** and ***Rouge-L*** assess the consistency between the generated text and the reference text. ***AdvBench***, ***HarmfulQA***, and ***CATQA*** denote refusal rates for the corresponding datasets.

2023), HarmfulQA (Bhardwaj and Poria, 2023), and CATQA (Bhardwaj et al., 2024). We employ Llama-Guard3-8B [2] to evaluate the model's safety by measuring its refusal rate to harmful queries. Compared to GPT-4, Llama-Guard3-8B demonstrates superior performance with a lower false positive rate. For detailed information of the datasets and Llama-Guard3-8B, please refer to D.3.

4.2 Experimental Results

4.2.1 Main Results on Downstream Tasks

LoRA SFT. As depicted in Table 1 and 2, our method significantly improves the safety of LLMs without compromising their classification performance. As evidenced by VLGuard, when harmful data is mixed into the training set, incorporating safety alignment data has a limited effect on enhancing the safety of LLMs.

Compared to RESTA, our approach effectively enhances the safety of the model while avoiding the impact of safety vectors on the performance of the downstream task. During the fine-tuning process of Lisa, the regularization of the proximal term limits changes in the parameters of the models, thereby inhibiting their performance on downstream tasks to some extent. For experiments on varying mixing ratios of toxic and safe data, as well as LLMs with different parameter scales, please refer to Appendices F and G.

Full SFT. According to Table 3, the model exhibits increased sensitivity to parameter changes due to full fine-tuning effects. For Llama3.1-8B-Instruct, the baselines affect its *BLEU* and *Rouge-L* scores by more than 0.2, while our method minimizes the suppression of downstream capabilities while ensuring the safety and consistency of LLMs. From Tables 1, 2 and 3, it is evident that the low-rank safety principal components consistently maintain the safety alignment of LLMs across various fine-tuning scenarios. In contrast, other baseline methods show significant variation in the safe rejection

[2] https://huggingface.co/meta-llama/Llama-Guard-3-8B

	DoAnythingNow ↑	AdvBench ↑	MBPP ↑	GSM8K ↑	BBH ↑	MMLU ↑	IFEval ↑
Base Model	0.81	0.87	59.20	82.79	68.29	69.14	81.65
SafetyJ	**0.92**	**0.98**	57.40	80.97	20.03	69.14	35.01
DARE	**0.92**	**0.98**	57.40	80.97	20.03	69.14	45.87
Ours	**0.92**	**0.98**	**59.40**	**84.52**	**68.52**	**69.17**	**81.88**

Table 4: Performance of Safety Components in Full Fine-Tuning Settings. ***Base Model*** represents Llama3.1-8B-Instruct, ***SafetyJ*** indicates the performance of the safety model after fine-tuning, and ***Ours*** signifies the performance of the base model after linear arithmetic with safety principal components.

	DoAnythingNow ↑	AdvBench ↑	MBPP ↑	GSM8K ↑	BBH ↑	MMLU ↑	IFEval ↑
Base Model	0.81	0.87	59.20	82.79	**68.29**	69.14	81.65
SafetyJ	0.91	**0.99**	56.80	82.56	31.42	67.28	45.79
DARE	0.89	**0.99**	56.80	82.41	31.89	67.45	56.91
Ours	**0.92**	**0.99**	**60.40**	**83.78**	68.20	**69.18**	**82.42**

Table 5: Performance of Safety Components in LoRA Fine-Tuning Settings. ***Base Model*** represents Llama3.1-8B-Instruct, ***SafetyJ*** indicates the performance of the safety model after fine-tuning, and ***Ours*** signifies the performance of the base model after linear arithmetic with safety principal components.

rate across different harmful query datasets, particularly pronounced in the LoRA fine-tuning scenario.

4.2.2 Low-Rank Safety Principal Components

To verify the effectiveness of low-rank safety principal components and demonstrate that principal components and LLMs share the same low-rank subspace, we perform linear operations between the safety components and LLMs to enhance their safety without significantly impacting their general performance. Given that instruction-tuned LLMs already exhibit a high degree of safety, we further fine-tune them using a carefully curated jailbreak dataset to obtain a more safety robust model. Specifically, we select JailJudge (Liu et al., 2024) jailbreak dataset to perform SFT on Llama3.1-8B-Instruct and use the AdvBench and DoAnything-Now (Shen et al., 2024) datasets to assess the safety of the LLMs. Additionally, we assess the model's general capabilities with the MBPP (Austin et al., 2021), GSM8K (Cobbe et al., 2021), BBH (Suzgun et al., 2022), MMLU (Hendrycks et al., 2021), and IFEval (Zhou et al., 2023) datasets. To demonstrate the superiority of low-rank safety principal component merging, we compare it with DARE (Yu et al., 2023) baseline. Details on the baseline, datasets, and SFT can be found in Appendix D.4.

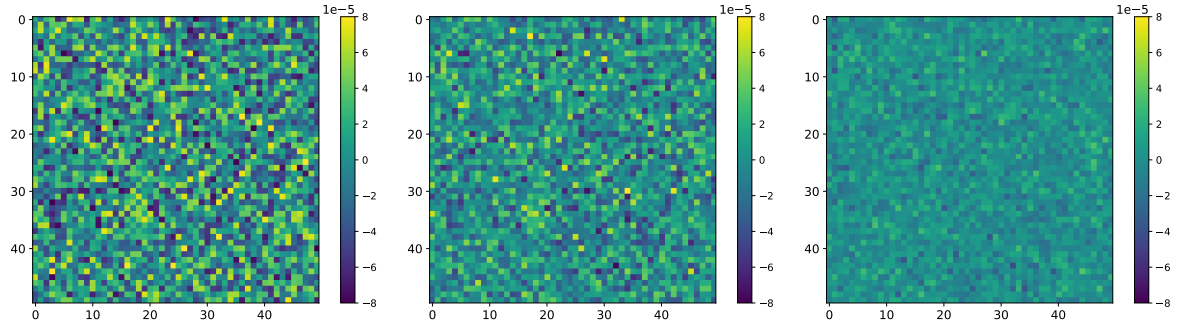
From Table 4 and Table 5, it can be observed that after applying SFT to Llama3.1-8B-Instruct (*SafetyJ*), its safety performance on *AdvBench* and *DoAnythingNow* improved significantly. However, its general capabilities show a varying degree of decline, particularly concerning the *BBH* and *IFEval* metrics. It indicates that the safety vector not only contains desired changes in the safety direction but also includes undesirable drift that adversely affects general capabilities.

Ours results show that applying linear operations between low-rank principal components and the *Base Model* enhances the model's safety robustness without significantly affecting its general capabilities. This suggests that the safety principal components of safety vectors share the same low-rank subspace as the primary safety drift in LLMs while remaining independent of the general capabilities direction, which is consistent with the perspective presented in Wei et al. (2024). Compared to DARE, our method causes less disruption to the model's general ability, showing superior safety realignment over traditional model fusion methods.

4.2.3 Impact of Safety Singular Value Entropy

To investigate the impact of safety singular value entropy on low-rank principal components, we conduct a visualization analysis of the safety vector and its low-rank principal components with different singular value entropy ratios in the *model.layers.5.mlp.down_proj* layer of the Llama3.1-8B-Instruct model.

In particular, we perform random downsampling with a fixed seed corresponding to the delta parameter matrix of the safety vector and the projection matrix of the low-rank principal com-



(a) safety vector (b) $\eta = 0.9$ (c) $\eta = 0.5$

Figure 3: Visualization of safety vector (a) and low-rank safety principal components (b, c) with different η at the layer model.layers.5.mlp.down_proj, where $\alpha = 1$. Visual representations of 2500 random sample positions are provided.

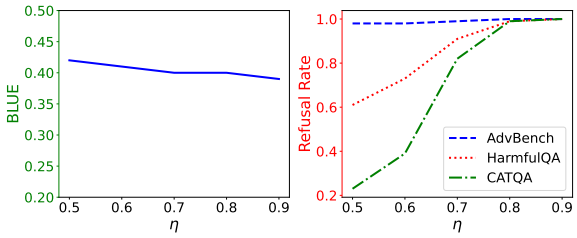


Figure 4: Impact of different singular value entropy thresholds on the performance of downstream fine-tuning models, with left singular vector weight $\alpha = 1.0$

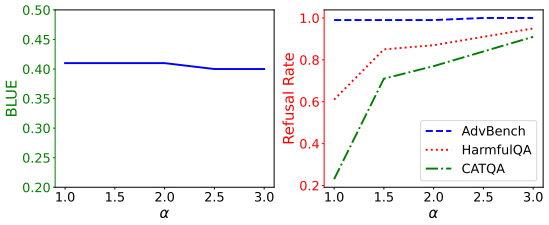


Figure 5: Impact of different left singular vector weights on the performance of downstream fine-tuning models, with singular value entropy ratio threshold of $\eta = 0.5$

ponents. The results of the down-sampling were visualized using heatmaps. As shown in Figure 3, as the singular value entropy ratio η decreases, the matrix corresponding to the low-rank principal components gradually becomes smoother, with its values tending towards zero. This indicates that the safety singular value entropy can effectively regulate the information content of low-rank principal components, thereby eliminating the interference of redundant directional information. As shown in Figures 3(b) and 3(c), our method does not suppress individual delta parameters but rather identifies a significant structured subnetwork from the global network. This aligns with our analysis of the low-rank safety subspace in Section 3.2.

For experiments on the relationship between singular value entropy and retained rank, please refer to Appendix H. For visualizations of the safety vectors and low-rank principal components of other layers, please refer to Appendix I.

4.3 Ablation Study

4.3.1 Importance of Singular Value Entropy

We conducted ablation experiments on the singular value entropy ratio threshold η to systematically investigate its impact on the performance of the fine-tuned model. As shown in Figure 4, increasing η results in a gradual improvement in the safety of the merged fine-tuned model. When $\eta > 0.8$, the safe refusal rate of the downstream fine-tuned model approaches 1.0, which is very close to the performance of the safety model, while the model's BLEU score is only slightly affected.

It indicates that by adjusting the threshold of the singular value entropy ratio, we modify the truncation rank of the low-rank projection matrix. This allows us to control the amount of safety information retained in the principal components after projecting the safety vector, thereby balancing the utility and safety of the downstream fine-tuned model.

4.3.2 Importance of Left Singular Vector

To assess how safety low-rank principal components, obtained from orthogonal projection matrices using left singular vectors, influence the performance of fine-tuned LLMs, we conducted ablation

experiments by adjusting the weights of the left singular vectors. In particular, as illustrated in Figure 4, we set $\eta = 0.5$ in the ablation experiment to achieve more pronounced results, ensuring that fine-tuned LLMs exhibit lower safety when the left singular value weight $\alpha = 1$. From Figure 5, it can be observed that as the weight corresponding to the left singular vectors gradually increases, the safety of the fine-tuned model also progressively improves. When $\alpha > 2.5$, the model's safety is close to the safety model, while its BLEU score is only slightly affected. This indicates that the projection matrix constructed from the left singular vectors accurately captures the main direction of safety drift in the safety vector, and increasing the drift in the corresponding direction can further enhance the model's safety.

5 Conclusion

Recent studies indicate that fine-tuning can compromise the safety guardrails of LLMs. In this paper, we propose the LSSF safety realignment framework to address the safety alignment issues caused by fine-tuning LLMs. Our experiments demonstrate that the low-rank safety subspace of LLMs remains largely unchanged during fine-tuning and is isolated from the direction of the model's general capabilities. Building on this, we utilize the low-rank principal components of the safety vector to rectify the safety drift of LLMs within the low-rank safety subspace, thereby restoring their safety alignment without compromising their performance on downstream tasks. Given that our method is independent of specific model architectures, we plan to extend it to multimodal and mixture of experts (MoE) models for further exploration in the future.

6 Limitations and Ethics Statements

Limitations Despite observing the widespread applicability of LSSF in downstream tasks, budget constraints prevented us from evaluating larger models such as Llama-3.1-405B-Instruct. Given that our method is independent of specific model architectures, we plan to extend it to multimodal and mixture of experts (MoE) models for further exploration in the future.

Ethics Statements Our study highlights the vulnerabilities in aligning large language models. It is undeniable that we used toxic data in our experiments to compromise model safety, which may have some negative impact on the safety of open-source models. However, considering that all datasets used in our experiments have been extensively studied in numerous academic works, this research does not amplify the inherent negative effects of the datasets themselves. Despite these concerns, we assert that analyzing the harmful aspects of large language models LLMs and exploring potential mitigation strategies have the potential to drive progress in enhancing the safety of LLMs.

7 Acknowledgement

This work was supported by the National Natural Science Foundation of China under grant number 62202170 and the Ant Group.

References

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

AI@Meta. 2024. Llama 3 model card.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Rishabh Bhardwaj, Duc Anh Do, and Soujanya Poria. 2024. Language models are Homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14138–14149, Bangkok, Thailand. Association for Computational Linguistics.

Rishabh Bhardwaj and Soujanya Poria. 2023. Red-teaming large language models using chain of utterances for safety-alignment. *arXiv preprint arXiv:2308.09662*.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2023. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*.

Steven Cao, Victor Sanh, and Alexander Rush. 2021. Low-complexity probing via finding subnetworks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–966, Online. Association for Computational Linguistics.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. volume 30.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*.

Mohammad-Javad Davari and Eugene Belilovsky. 2024. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, page 270–287, Berlin, Heidelberg. Springer-Verlag.

Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Mingyuan Fan, Chengyu Wang, Cen Chen, Yang Liu, and Jun Huang. 2025. On the trustworthiness landscape of state-of-the-art generative models: A survey and outlook. *International Journal of Computer Vision*, pages 1–32.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, Online. Association for Computational Linguistics.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. volume 53, pages 217–288. SIAM.

Luxi He, Mengzhou Xia, and Peter Henderson. 2024. What's in your" safe" data?: Identifying benign data that breaks safety.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024. Lazy safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2405.18641*, 2.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *ArXiv*, abs/2212.04089.

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. 2024. Model stock: All we need is just a few fine-tuned models. page 207–223, Berlin, Heidelberg. Springer-Verlag.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Fan Liu, Yue Feng, Zhao Xu, Lixin Su, Xinyu Ma, Dawei Yin, and Hao Liu. 2024. Jailjudge: A comprehensive jailbreak judge benchmark with multi-agent enhanced explanation evaluation framework. *ArXiv*, abs/2410.12855.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to!

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Xinyue Shen, Zeyuan Johnson Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 1671–1685, New York, NY, USA. Association for Computing Machinery.

Ken Shoemake. 1985. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph.*, 19(3):245–254.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Annual Meeting of the Association for Computational Linguistics*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2024. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. pages 23965–23998.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. 2024. LM-cocktail: Resilient tuning of language models via model merging. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2474–2488, Bangkok, Thailand. Association for Computational Linguistics.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Neural Information Processing Systems*.

Le Yu, Yu Bowen, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Language models are super mario: Absorbing abilities from homologous models as a free lunch.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Feiran Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. volume abs/2304.05302.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. 2024. Removing RLHF protections in GPT-4 via fine-tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 681–687, Mexico City, Mexico. Association for Computational Linguistics.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2024. LoRAPrune: Structured pruning meets low-rank parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3013–3026, Bangkok, Thailand. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. volume 28.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy M. Hospedales. 2024. Safety fine-tuning at (almost) no cost: a baseline for vision large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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

A Proof of Shared Low-Rank Subspace

Given that $\widehat{X} \in \mathbb{R}^{d_{\text{in}} \times n}$ is the corresponding input matrix computed by weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ of $\boldsymbol{\theta}_{\text{safe}}$ based on the calibration dataset $\mathcal{D}_{\text{anchor}}$, specifically:

$$Z = W\widehat{X} \tag{14}$$

where Z denotes the activation matrix the linear layer corresponding to W. Assume that W is obtained by safety fine-tuning an unsafe model, which can be expressed as:

$$\begin{aligned} W =& \boldsymbol{\theta}_{\text{unsafe}} + \boldsymbol{\delta}_{\text{safe}} \\ =& \boldsymbol{\theta}_{\text{unsafe}} + \boldsymbol{\tau}_{\text{safe}} + \hat{\boldsymbol{\tau}}_{\text{safe}}, \end{aligned} \tag{15}$$

where $\boldsymbol{\theta}_{\text{safe}}$ represents the safety-related shift in the safety vector $\boldsymbol{\theta}_{\text{safe}}$ and $\hat{\boldsymbol{\tau}}_{\text{safe}}$ denotes the directional drift that suppresses general capabilities. The Frobenius norm minimization of the low-rank $\widehat{W}$ is given by

$$\begin{aligned} \widehat{W} &= \arg\min_{\widehat{W}} \|W\widehat{X} - \widehat{W}\widehat{X}\|_F^2 \\ &= \arg\min_{\widehat{W}} \sum_{i=1}^{n} \|W\hat{x}_i - \widehat{W}\hat{x}_i\|^2 \\ &= \arg\min_{\widehat{W}} \sum_{i=1}^{n} \|\boldsymbol{\theta}_{\text{unsafe}}\hat{x}_i + \boldsymbol{\tau}_{\text{safe}}\hat{x}_i + \\ &\qquad \hat{\boldsymbol{\tau}}_{\text{safe}}\hat{x}_i - \widehat{W}\hat{x}_i\|^2 \end{aligned} \tag{16}$$

According to the Eckart-Young theorem (Eckart and Young, 1936), for a given matrix W, the optimal low-rank approximation of rank r is obtained by retaining the top r singular values and their corresponding singular vectors from its singular value decomposition. Therefore, the optimal solution $\widehat{W}$ should preserve the most significant variations, specifically those associated with $\boldsymbol{\tau}_{\text{base}}$. Since $\hat{\boldsymbol{\tau}}_{\text{safe}}$, which suppresses general capabilities, does not impact the model's safety, it does not become a principal component in $W\widehat{X}$. Consequently, the low-rank approximation $\widehat{W}$ does not include the $\hat{\boldsymbol{\tau}}_{\text{safe}}$ component. For the corresponding experiments, please refer to 4.2.2.

B Proof of The Singular Value Entropy

The definition of the Frobenius norm is the square root of the sum of the squares of all elements in a matrix. Specifically, if A is an $m \times n$ matrix, then the Frobenius norm is defined as:

$$\|A\|_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 \tag{17}$$

On the other hand, the Singular Value Decomposition (SVD) of a matrix provides a decomposition of A as follows:

$$A = U\Sigma V^\top \tag{18}$$

where U and V are unitary orthogonal matrices, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$ is the singular value diagonal matrix.
The Frobenius norm has the property:

$$\|A\|_F^2 = \text{Tr}(A^\top A) \tag{19}$$

After computing $A^\top A$, we have:

$$\begin{aligned} A^\top A &= (U\Sigma V^\top)^\top (U\Sigma V^\top) \\ &= V\Sigma^\top U^\top U\Sigma V^\top \\ &= V\Sigma^2 V^\top \end{aligned} \tag{20}$$

Since V is a unitary orthogonal matrix, it follows that:

$$\text{Tr}(V\Sigma^2 V^\top) = \text{Tr}(\Sigma^2) \tag{21}$$

$\Sigma^2$is a diagonal matrix, and $\text{Tr}(\Sigma^2)$ is the sum of its diagonal elements:

$$\text{Tr}(\Sigma^2) = \sum_{i=1}^{r} \sigma_i^2 \tag{22}$$

Thus, we ultimately have:

$$\|A\|_F^2 = \sum_{i=1}^{r} \sigma_i^2 \tag{23}$$

Therefore, it is proven that $\|A\|_F^2 = \sum_{i=1}^{r} \sigma_i^2$.

C Proof of The Optimality of SVD

Let $\widehat{X} \in \mathbb{R}^{d_{\text{in}} \times n}$ and $\widehat{W}$ denote the solution to the following rank-constrained approximation problem:

$$\widehat{W} = \arg\min_{\text{rank}(\widehat{W}) \le r} \|W\widehat{X} - \widehat{W}\widehat{X}\|_F^2 \tag{24}$$

Perform a low-rank matrix decomposition of $\widehat{W}\widehat{X}$ using Singular Value Decomposition (SVD):

$$USV^\top \approx \widehat{W}\widehat{X} \tag{25}$$

where $U \in \mathbb{R}^{d_{\text{out}} \times r}$ is an orthogonal matrix composed of the first r left singular vectors. The minimum of the constrained problem is achieved by:

$$\widehat{W} = UU^\top W \tag{26}$$

Let X_{in} denote the input corresponding to the weight matrix W, and $Z = WX_{\text{in}}$. According to the Eckart–Young theorem (Eckart and Young, 1936), the singular value decomposition (SVD) $\widehat{Z} = USV^\top$ provides the optimal rank-r approximation of Z. Substituting $Z = WX_{\text{in}}$, we obtain:

$$\widehat{Z} = UU^\top WX_{\text{in}} \tag{27}$$

By setting $\widehat{W} = UU^\top W$, it follows that:

$$\begin{aligned} &\|\widehat{Z} - Z\|_F^2 \text{ is minimized} \quad \Rightarrow \\ &\|\widehat{W}\widehat{X} - W\widehat{X}\|_F^2 \text{ is minimized} \end{aligned} \tag{28}$$

Furthermore, since $UU^\top$ is a rank-r projection matrix, it holds that $\text{rank}(\widehat{W}) \leq r$. Therefore, $\widehat{W}$ is the optimal solution to the rank-constrained minimization problem.

D Experimental Details

D.1 Baseline Setup

To mitigate fine-tuning risks, we select baseline methods encompassing both in-fine-tuning alignment and post-alignment approaches, including:

- NA-SFT: Utilizes only the fine-tuning dataset without enforcing safety alignment.
- VLGuard (Zong et al., 2024): A defensive solution against harmful fine-tuning attacks during the fine-tuning phase, which integrates safety-aligned data into the fine-tuning process to continuously reinforce the model's alignment knowledge. Originally applied to visual-LLM fine-tuning, in this paper, we employ the SafeInstr (Bianchi et al., 2023) dataset for safety alignment.
- Lisa (Huang et al., 2024): Separates the fine-tuning phase into two states to independently optimize alignment and user datasets, thereby mitigating jailbreak effects. Additionally, it introduces a proximal term to constrain the drift of each state.
- RESTA (Bhardwaj et al., 2024): Combines the safety vector with the weights of the compromised model through simple arithmetic combination and employs DARE (Yu et al., 2023) to merge with the original model, thereby alleviating the suppression impact on the general capabilities of the safety vector.

The detailed hyperparameter settings for each method are as follows:

- VLGuard: For VLGuard, we utilize the Safe-Instr safety calibration dataset. To align with NA-SFT, we randomly select 500 samples from Insfer.
- Lisa: For Lisa, we set $align\_step = 100$, $finetune\_step = 900$, and the proximal penalty $\rho = 1$. As described in Section Appendix, the training set comprises the downstream task fine-tuning dataset and 500 harmful samples, while the alignment dataset consists of 500 safe samples.
- RESTA: For RESTA, we assign weights of 1, 1, and -1 to the compromised model, base model, and unaligned model, respectively.

Consistent with our training hyperparameters, we set the number of training epochs to 10 and the learning rate to 1×10^{-5} for all tasks. For LoRA fine-tuning, we set $r = 16$. All fine-tuning tasks were conducted on 8 Nvidia A100 GPUs.

D.2 Dataset

In our downstream fine-tuning experiments, we established two distinct scenarios:

LoRA Fine-Tuning. For text classification, we utilize two multi-class datasets: AG's News and Yahoo Answers (Zhang et al., 2015). AG's News is primarily utilized for news classification tasks and comprises news article snippets from various sources. The dataset is divided into four categories: World, Sports, Business, and Sci/Tech. We randomly selected 50K samples from the training set to perform supervised fine-tuning of the LLM and randomly chose 1K samples from the test set to evaluate the classification accuracy of the fine-tuned model. The Yahoo Answers dataset is a large-scale multi-class dataset derived from Q&A dialogues on the Yahoo Answers platform, encompassing ten categories. Similarly, we randomly chose 50K training samples for fine-tuning and 1K test samples for evaluation.

Full Fine-Tuning. For the text generation task, we utilized the Medical Dialogue Dataset [3], which includes Chinese medical dialogue data from six departments, such as andrology, internal medicine,

[3] https://github.com/Toyhom/Chinese-medical-dialogue-data

and obstetrics and gynecology. We randomly selected 50K samples from the internal medicine category as the training set and randomly chose 1K samples to evaluate the text generation accuracy of the fine-tuned model.

Harmful and safe dataset. To simulate real-world fine-tuning scenarios (Bianchi et al., 2023), we randomly selected 500, 2,500, and 25,000 Q&A pairs from the PKU-SafeRLHF dataset in which both *accept* and *reject* are labeled as *unsafe* to construct harmful dataset. Similarly, we randomly selected 500 and 2,500 harmful queries and prompt the safety model to generate safe negative responses to construct safe dataset. These datasets were then mixed into the fine-tuning dataset in various combinations for supervised fine-tuning.

D.3 Metric

D.3.1 Measuring utility

For the text classification task, we use classification accuracy ACC, calculated as follows:

$$\text{ACC} = \frac{N_{\text{correct}}}{N_{\text{total}}}$$

where N_{correct} represents the number of correctly predicted samples and N_{total} denotes the total number of samples. For the text generation task, we employ BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) as evaluation metrics. BLEU assesses the quality of generated text by computing the precision of n-gram matches between the candidate text and the reference text. ROUGE-L, based on the Longest Common Subsequence (LCS), evaluates the consistency between the generated text and the reference text in terms of word order and content, reflecting the overall similarity of sentence structures.

D.3.2 Measuring Safety

Safety Evaluation Dataset. We employ three datasets to evaluate the model's safety:

- AdvBench (Zou et al., 2023): This dataset comprises 520 harmful requests along with their corresponding target strings, covering a wide range of malicious topics such as profanity, threats, discrimination, and cybercrime.
- HarmfulQA (Bhardwaj and Poria, 2023): Sourced from harmful dialogues in ChatGPT, this dataset encompasses 10 safety themes, from which 196 questions are extracted across various subtopics.
- CATQA (Bhardwaj et al., 2024): This dataset consists of 550 harmful questions spanning 11 categories, each containing 5 harmful subcategories.

Llama-Guard3-8B. We use Llama-Guard3-8B [4] to assess the model's safety against harmful queries. Compared to GPT-4, Llama-Guard3-8B performs better with a lower false positive rate. Additionally, Llama-Guard3-8B outperforms GPT-4 in English proficiency, multilingual capabilities, and tool usage, exhibiting a significantly lower false positive rate. For the specific prompts used in Llama-Guard3-8B, please refer to Table 6.

D.4 Low-Rank Safety Principal Components

We select Llama3.1-8B-Instruct as the base model and perform SFT fine-tuning using the sampled JailJudge (Liu et al., 2024) dataset to obtain the safety model. JailJudge is a comprehensive benchmark that features diverse risk scenarios, including synthetic, adversarial, in-the-wild, and multilingual prompts, along with high-quality human-annotated datasets. To minimize the impact of data distribution on the experiment, we use DoAnythingNow (Shen et al., 2024) and AdvBench to evaluate the safety of the LLM. DoAnythingNow is a jailbreak dataset that includes 13 disabled scenarios, from which we randomly selected 1000 instances for testing. To demonstrate the effectiveness of low-rank safety principal components, we compare it with the model fusion method DARE (Yu et al., 2023). DARE reduces the number of effective parameters in task vectors by removing the delta parameter and proportionally scaling the remaining values. This intuitive approach helps alleviate parameter interference when integrating multiple models. Even for a single task-specific model, pruning certain parameters can mitigate interference and potentially enhance safety. To evaluate the general capabilities of the LLM, we use the following datasets:

- MBPP (Austin et al., 2021): MBPP comprises approximately 1,000 Python programming problems, covering fundamental programming knowledge and standard library functions. Each problem includes a task description, a code solution, and three automated test cases.
- GSM8K (Cobbe et al., 2021): GSM8K contains 8.5K high-quality, linguistically di-

[4] https://huggingface.co/meta-llama/Llama-Guard-3-8B

Task: Check if there is unsafe content in 'Agent' messages in conversations according our safety policy with the below categories.

<BEGIN UNSAFE CONTENT CATEGORIES>
S1: Violent Crimes.
S2: Non-Violent Crimes.
S3: Sex Crimes.
S4: Child Exploitation.
S5: Defamation.
S6: Specialized Advice.
S7: Privacy.
S8: Intellectual Property.
S9: Indiscriminate Weapons.
S10: Hate.
S11: Self-Harm.
S12: Sexual Content.
S13: Elections.
S14: Code Interpreter Abuse.
<END UNSAFE CONTENT CATEGORIES>

<BEGIN CONVERSATION>

User: {}

Agent: {}

<END CONVERSATION>
Provide your safety assessment for Agent message in the above conversation:
- First line must read 'safe' or 'unsafe'.
- You only need to output 'safe' or 'unsafe', nothing else.

Table 6: Specific prompts for safety classification using Llama-Guard3-8B in our experiments

Num	16	32	64	128	256	512
ACC	0.54	0.63	0.64	0.64	0.63	0.64
AdvBench	1.00	1.00	1.00	1.00	1.00	1.00
HarmfulQA	0.94	0.99	0.99	0.99	0.99	1.00
CATQA	0.93	0.99	0.99	0.99	0.99	0.99

Table 7: Sensitivity analysis for the Yahoo Answers LoRA task with varying amounts of calibration datasets.

verse elementary school math word problems. These problems typically require 2 to 8 steps to solve, primarily involving basic arithmetic operations.

- BBH (Suzgun et al., 2022): BIG-Bench Hard (BBH) is a subset of BIG-Bench, focusing on 23 challenging tasks within BIG-Bench that previous language model evaluations have not surpassed the performance of average human scorers.

- MMLU (Hendrycks et al., 2021): MMLU specifically assesses the knowledge acquired during pre-training in zero-shot and few-shot settings. It covers 57 subjects across disciplines such as humanities and social sciences.

- IFEval (Zhou et al., 2023): IFEval evaluates the instruction-following capabilities of large language models, containing over 500 prompts.

These datasets collectively provide a comprehensive assessment of the model's general proficiency across various domains and tasks. During the training process, we set the number of epochs to 3 and the learning rate to 1e-5. The fine-tuning was conducted on 8 Nvidia A100 GPUs.

E Robustness Against the Number of Calibration Datasets

To validate the sensitivity of our method to the number of samples in the calibration dataset, we randomly selected varying numbers of samples from the PKU-SafeRLHF dataset to conduct ablation experiments. Specifically, we performed safety realignment for Yahoo Answers LoRA fine-tuning on Llama3.1-8B-Instruct. As shown in Table 7, when the calibration dataset size reaches 64, our method achieves optimal performance without significant fluctuations as the dataset size increases. Nevertheless, to align with the experimental setup in reference (Wei et al., 2024), we have chosen 128 as the default calibration dataset size.

F Robustness Against Data Composition

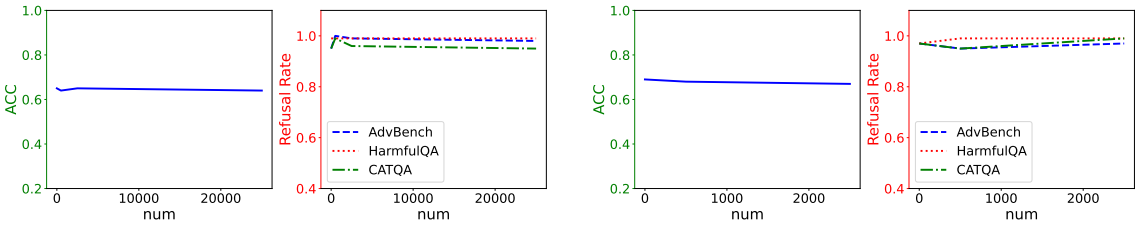
To verify the robustness of our method, we conduct LoRA SFT on the Yahoo Answers training set for Llama3.1-8B-Instruct by mixing harmful and safe data in varying proportions. Both harmful and safe data were sourced from the PKU-SafeRLHF dataset, as described in Appendix D.2. As shown in Figure 6(a), the LLM generated by our method exhibit no significant changes in ACC and rejection rates as the proportion of harmful data increases in the training dataset. This demonstrates the high robustness of our method when facing training sets with different proportions of harmful data. Similarly, Figure 6(b) illustrates that mixing varying proportions of safe data into the training set does not significantly affect the performance of aligned LLMs, further confirming the robustness of our method. However, from empirical evidence (Bianchi et al., 2023), it is necessary to incorporate safe data into training dataset. Adding safe data to the training set not only enhances the lower bound of safety performance in downstream fine-tuning models but also suppresses excessive drift in safety direction during fine-tuning, which is beneficial for improving the effectiveness of our method.

G Robustness Against Model Parameters

To demonstrate the effectiveness of our method across models with varying parameter sizes, we conducted ablation experiments. We conducted LoRA SFT on Qwen2.5 with varying parameter sizes using the AG's News and Yahoo Answers datasets. Additionally, we applied our method for safety realignment. As shown in Table 8, for models with 3B, 7B, and 14B parameters, our method consistently achieves a safety refusal rate of 0.99. This demonstrates the effectiveness of our method across different parameter sizes in LLMs.

H Singular Value Entropy and Rank

To verify the impact of singular value entropy on the retention rank of the weight matrix, we calculated the relationship between the singular value entropy ratio η and the retained rank ratio of the corresponding weight matrix for Llama3.1-8B-Instruct. From Figure 7, it can be observed that as the proportion of singular value entropy increases, the number of retained ranks in the weight matrix also



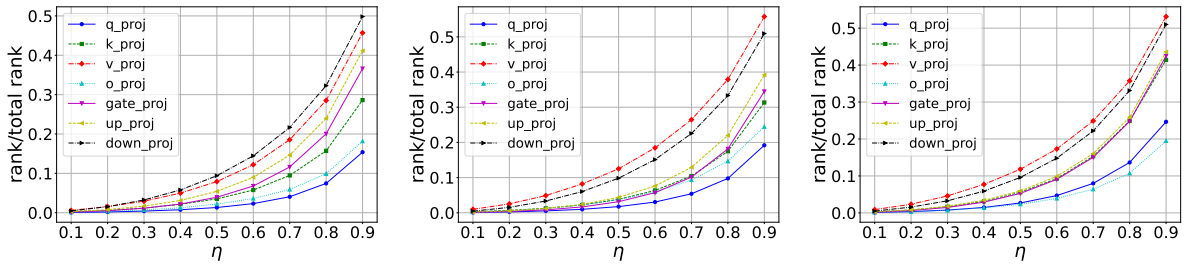
(a) Robustness to varying proportions of harmful data, with a mixture of 500 safe instances.

(b) Robustness to varying proportions of safe data, with a mixture of 2500 harmful instances.

Figure 6: Our method's robustness on Llama3.1-8B-Instruct against varying proportions of harmful or safe data in the Yahoo Answers training dataset.

Base Model	AG's News				Yahoo Answers			
	ACC	AdvBench	CATQA	HarmfulQA	ACC	AdvBench	CATQA	HarmfulQA
Qwen2.5-3B-Instruct	0.92	0.99	0.97	0.93	0.67	0.98	0.99	0.93
Qwen2.5-7B-Instruct	0.92	1.00	0.98	0.93	0.68	1.00	0.99	0.99
Qwen2.5-14B-Instruct	0.92	1.00	0.99	0.95	0.69	0.99	0.99	0.99

Table 8: Our method's robustness across models with different parameter sizes on Qwen2.5. ***Base Model*** denotes various sizes of the Qwen2.5 model. The hyperparameters are set as $\eta = 0.9$ and $\alpha = 1.5$.



(a) layer-5

(b) layer-15

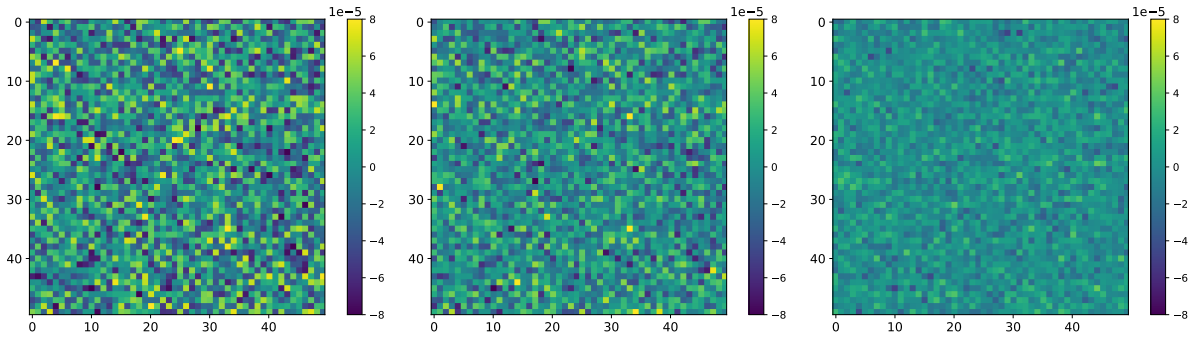
(c) layer-25

Figure 7: Influence of the singular value entropy threshold η on the safety retained rank r of weight matrices across various layers in Llama3.1-8B-Instruct.

increases. This demonstrates that we can control the amount of information retained in the low-rank safety principal components through singular value entropy. Within a single layer, different weight matrices exhibit varying encoding densities for safety information. For instance, in $layer - 15$, when η is 0.9, the difference in the proportion of retained ranks between v_proj and q_proj exceeds 30%. Across different layers, the relative encoding density of safety information by different weight matrices also changes. In the shallower layers, q_proj exhibits the highest encoding density, whereas in the deeper layers, o_proj becomes the matrix with the highest encoding density. This analysis demonstrates that safety singular value entropy allows for precise quantification of safety information encoding density in weight matrices across different layers, thereby facilitating the dynamic determination of the rank retention during low-rank pruning.

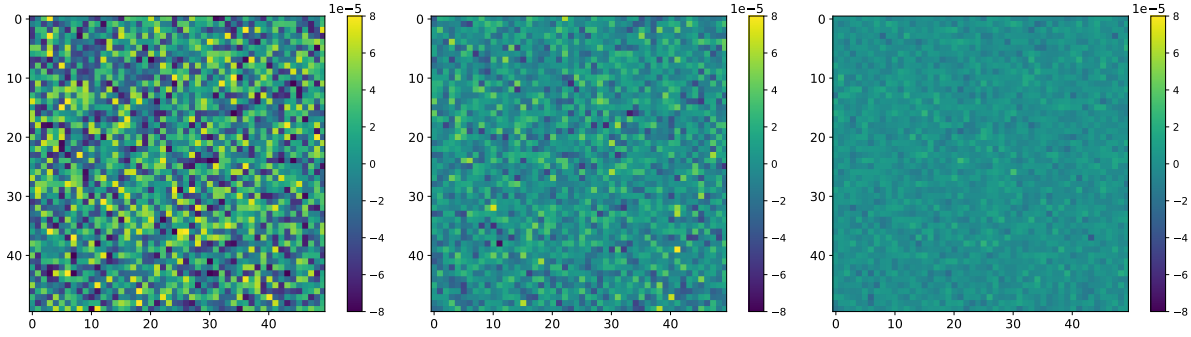
I Visualization

To analyze the relationship between the safety vector and low-rank safety principal components, we visualized these elements. Figures 3, 8, and 10 illustrate as the singular value entropy decreases, the rank of low-rank safety principal components in different layers also decreases, resulting in progressively smoother images. This result is consistent with our preliminary analysis. Comparing Figures 8 and 9, we note that due to the higher density of safety information encoding in *model.layers.15.self_attn.q_proj*, its low-rank safety components exhibit a lower rank when extracting an equivalent proportion of information,



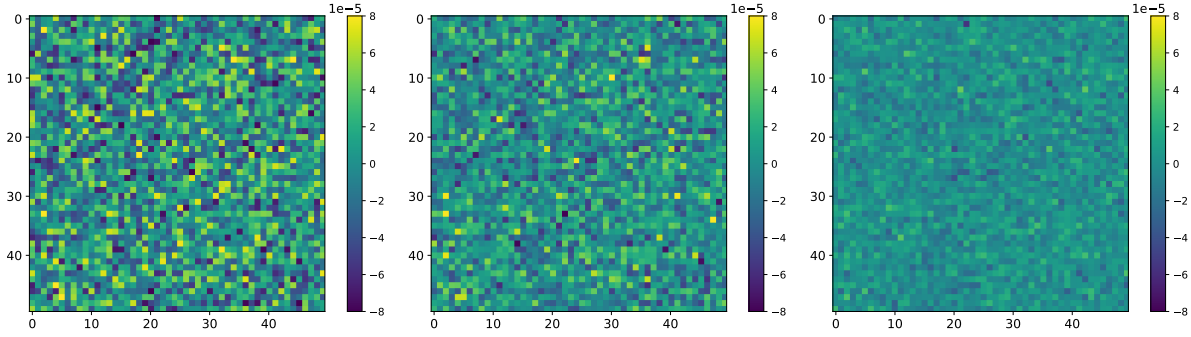
(a) safety vector (b) $\eta = 0.9$ (c) $\eta = 0.5$

Figure 8: Visualization of safety vector (a) and low-rank safety principal components (b, c) with different η at the layer model.layers.15.self_attn.v_proj, where $\alpha = 1$. Visual representations of 2500 random sample positions are provided.



(a) safety vector (b) $\eta = 0.9$ (c) $\eta = 0.5$

Figure 9: Visualization of safety vector (a) and low-rank safety principal components (b, c) with different η at the layer model.layers.15.self_attn.q_proj, where $\alpha = 1$. Visual representations of 2500 random sample positions are provided.



(a) safety vector (b) $\eta = 0.9$ (c) $\eta = 0.5$

Figure 10: Visualization of safety vector (a) and low-rank safety principal components (b, c) with different η at the layer model.layers.25.self_attn.v_proj, where $\alpha = 1$. Visual representations of 2500 random sample positions are provided.

producing smoother images. This suggests that, for matrices with high encoding density, a lower rank can effectively represent the safety information corresponding to the safety vector.