

Q-realign: Piggybacking Realignment on Quantization for Safe and Efficient LLM Deployment

Qitao Tan¹ Xiaoying Song² Ningxi Cheng¹ Ninghao Liu¹
Xiaoming Zhai¹ Lingzi Hong² Yanzhi Wang³ Zhen Xiang¹ Geng Yuan¹

¹University of Georgia ²University of North Texas ³Northeastern University

Abstract

Public large language models (LLMs) are typically safety-aligned during pretraining, but task-specific fine-tuning often erodes this alignment, introducing significant safety risks. Existing defenses are largely coupled with the fine-tuning process, incurring high computational overhead and complex workflows. To address these challenges, we propose Q-realign, a post-hoc defense based on quantization. Departing from the conventional view of quantization as a safety-eroding compromise, we reframe it as a dual-purpose mechanism for both compression and safety recovery. This allows Q-realign to decouple safety alignment from fine-tuning and naturally piggyback onto modern deployment pipelines. Experiments across multiple models and datasets demonstrate that our method substantially reduces unsafe behaviors while preserving task performance, significantly improving memory and GPU efficiency. Notably, our approach can recover the safety alignment of a fine-tuned 7B LLM on a single RTX 4090 within 40 minutes. Overall, our work provides a practical, turnkey solution for safety-aware deployment. Our code is released at <https://github.com/Skiltee/Q-Realign>.

1 Introduction

Fine-tuning of large language models (LLMs) has become a critical way for a wide range of specialized downstream applications. To ensure that LLMs can generate helpful content and refuse malicious requests (e.g., how to make a bomb), modern LLMs typically require alignment training, usually by RLHF (Ouyang et al., 2022). However, recent studies (Qi et al., 2023; Yang et al., 2023; Huang et al., 2024a; Li et al., 2024; Ye et al., 2024) show that fine-tuning can significantly compromise the alignment of already aligned LLMs, leading to more unsafe behaviors, even when the training data appears benign. This risk is especially concerning because datasets may contain latent unsafe content

that users cannot easily detect.

Existing solutions addressing safety degradation caused by fine-tuning on aligned models can be broadly categorized according to the stage at which they operate (Huang et al., 2024c). The first category intervenes during fine-tuning by modifying the training objective or procedure to jointly optimize downstream task performance and safety alignment (Mukhoti et al., 2023; Zong et al., 2024; Lyu et al., 2024; Du et al., 2024a; Qi et al., 2024; Li et al., 2025b; Yi et al., 2025a). The second category applies defenses after fine-tuning, aiming to recover safety post hoc (Du et al., 2024b; Bhardwaj et al., 2024; Yi et al., 2024; Wang et al., 2025). However, these methods typically rely on fine-tuning-derived signals or require auxiliary adaptation, limiting their plug-and-play applicability. Though studies are exploring one-shot post-hoc defenses (Hsu et al., 2024; Huang et al., 2025; Yi et al., 2025b), their empirical performance is generally unfavorable under realistic attack scenarios.

Despite operating at different stages, both categories remain tightly coupled with fine-tuning. This coupling leads to substantial computational overhead, often in the form of additional training iterations, increased memory consumption, and extended GPU usage. More importantly, it limits deployment flexibility: training-time methods require retraining by construction, while post-hoc approaches often still depend on fine-tuning-derived signals or additional adaptation, thereby requiring re-entry into the fine-tuning pipeline. As a result, when safety vulnerabilities are identified after deployment, existing defenses still struggle to provide truly modular and plug-and-play recovery. Motivated by these limitations, we ask *whether safety degradation can be mitigated in a way that is decoupled from the fine-tuning process and more efficient in both computation and workflow*.

Driven by this observation, we seek a safety recovery method that operates at an inherent post-

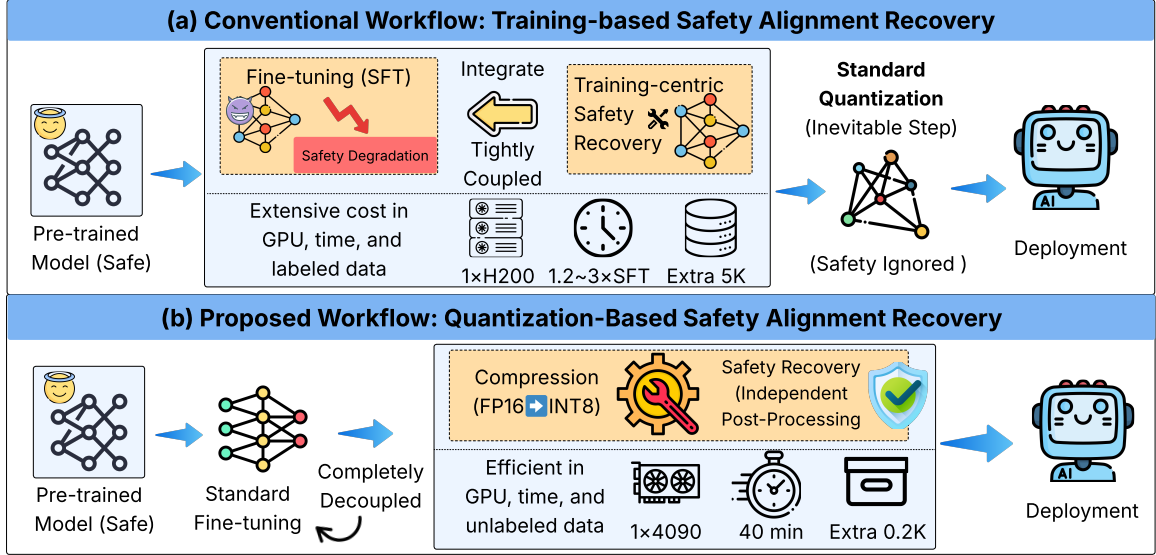


Figure 1: Comparison between conventional training-based safety recovery and our quantization-based approach. Our method fully decouples safety recovery from fine-tuning, substantially reduces computational overhead, and simplifies the deployment workflow.

training stage between fine-tuning and deployment, avoiding a costly loop back to the fine-tuning stage. Quantization is a de facto standard in modern LLM deployment, yet it is largely regarded as a necessary compromise, a process that inevitably sacrifices safety for efficiency (Hong et al., 2024; Kharinaev et al., 2025; Chen et al., 2025a). In this work, we propose a novel perspective: *rather than a source of safety degradation, quantization can be repurposed as a powerful mechanism for safety recovery*. This insight allows us to reframe quantization as a dual-purpose operation that not only facilitates compression but also provides a turnkey opportunity to "piggyback" safety alignment recovery. By integrating defense directly into the deployment pipeline, we effectively decouple safety realignment from the resource-intensive fine-tuning stage, achieving significant gains in both computational and workflow efficiency.

To understand how quantization can recover safety alignment after fine-tuning, we analyze the representational structure of LLMs before and after fine-tuning and show that fine-tuning weakens the semantic separability underlying alignment. Based on this insight, we introduce learnable layer-wise quantization parameters to selectively constrain activations and recover this separability, thereby recovering alignment. Comprehensive experimental results show that our method significantly reduces unsafe behaviors while preserving task performance, with significant reductions in memory

usage and GPU hours. In practice, our approach recovers the safety alignment of a fine-tuned 7B LLM on a single RTX 4090 within 40 minutes.

We summarize our contributions as follows: 1) We introduce an analysis to uncover how fine-tuning compromises the safety alignment from the perspective of representational structure. 2) We propose Q-realign, to the best of our knowledge, the first quantization-based defense framework for fine-tuning attacks, which can naturally be integrated into the deployment pipeline. 3) Q-realign consistently exceeds existing baselines in safety alignment recovery while maintaining fine-tuning accuracy, with significant improvements in memory usage and GPU hours. These advantages hold across diverse tasks and LLM architectures.

2 Preliminaries and Activation Analysis

2.1 Threat Model and Assumptions

Our threat model and experimental setup are defined as follows. A model provider fine-tunes a pre-aligned LLM (e.g., LLaMA2-7B-chat) on task-specific data and deploys it as an online service accessed via an API. In this setting, the provider remains responsible for harmful outputs generated after fine-tuning and may face significant governance and legal risks (Reuel et al., 2024).

Unlike prior work that assumes access to large labeled alignment datasets with harmful instructions paired with safe responses (Hsu et al., 2024; Huang et al., 2024b; Wang et al., 2025), our method oper-

ates under a weaker assumption: it requires only harmful instructions and substantially less alignment data. This makes our approach more practical for real-world deployment, where collecting high-quality safety annotations is costly. We present our related work section in Appendix A.

2.2 How Fine-tuning Compromises Safety Alignment?

In this section, we investigate how fine-tuning compromises the safety alignment of LLMs by examining changes in their representational structure. Specifically, we analyze the separability of layer-wise intermediate activations of benign and malicious inputs before and after fine-tuning. The separability is measured from two complementary perspectives: *spatial separability*, capturing geometric separation in representation space, and *semantic separability*, reflecting semantically separable information.

Analysis Method: Spatial separability is measured by Fisher’s linear discriminant ratio (Fisher, 1936). Given the token-averaged l -th layers’ activation $\mathbf{h}^{(l)} \in \mathbb{R}^d$, the spatial separability is formulated as:

$$\mathcal{S}^{(l)} = \frac{\left\| \mathbb{E}_{\mathbf{h} \sim \mathcal{H}_{\text{benign}}^{(l)}} [\mathbf{h}] - \mathbb{E}_{\mathbf{h} \sim \mathcal{H}_{\text{malicious}}^{(l)}} [\mathbf{h}] \right\|_2}{\mathbb{E}_{\mathbf{h} \sim \mathcal{H}^{(l)}} \left[\left\| \mathbf{h} - \boldsymbol{\mu}_{y(\mathbf{h})}^{(l)} \right\|_2 \right] + \varepsilon} \quad (1)$$

where $\mathcal{H}_{\text{benign}}^{(l)}$ and $\mathcal{H}_{\text{malicious}}^{(l)}$ denote the benign and malicious activation sets, respectively, $\boldsymbol{\mu}_{y(\mathbf{h})}^{(l)}$ is the centroid of $\mathbf{h}$ ’s class, and ε is a small constant for numerical stability.

A higher $\mathcal{S}^{(l)}$ indicates that benign and malicious representations are more compact within each class and more distant across classes, reflecting stronger separability.

We measure semantic separability using the refusal rate, which reflects how often malicious inputs are mapped to semantic regions associated with refusal behavior. For each malicious input, we perform layer-wise decoding from the intermediate activations at layer l and check whether the generated output contains explicit refusal expressions. The refusal rate at layer l is defined as the proportion of malicious inputs whose decoded outputs exhibit refusal-related language. Intuitively, if intermediate activations encode clear semantic separation between benign and malicious inputs, this separation becomes behaviorally explicit during decoding, resulting in a higher refusal rate. Refusal

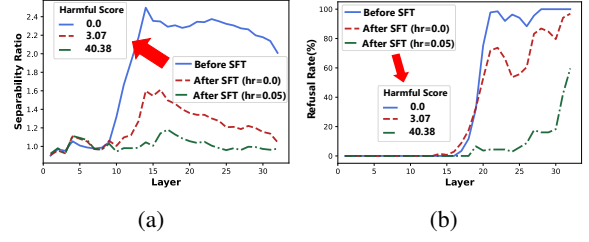


Figure 2: Layer-wise separability and refusal rate across pre-trained and fine-tuned models with varying harmful ratios (hr). A lower harmful score indicates a safer model.

detection is implemented using a keyword-based method following Chao et al. (2023).

Analysis Results: Figure 2a and Figure 2b illustrate the spatial separability and semantic separability before and after fine-tuning, respectively. As shown in Figure 2a, the aligned pre-trained model exhibits higher separability in the middle and deeper layers, whereas fine-tuning substantially reduces separability across most layers, with a more pronounced decline when a higher fraction of malicious data is used (e.g., $hr = 0.05$). This reduction in spatial separability is consistent with the observed increase in the harmful score. These trends suggest that fine-tuning blurs the separation between benign and malicious behaviors, undermining safety alignment and increasing vulnerability to harmful inputs (refer to Figure 3 and Figure F.1 for 2D visualization).

In Figure 2b, we report the layer-wise refusal rate as a behavioral measure of semantic separability. In the aligned pre-trained model, decoding directly from intermediate activations, rather than only from the final layer, already yields a very high refusal rate in the middle and upper layers (e.g., layer 20+). Specifically, activations from these layers lead to refusal of nearly all malicious inputs, indicating that malicious activations are mapped to semantically distinct regions associated with refusal behavior. Although decoding from intermediate layers may produce less fluent outputs due to bypassing later decoding stages, the model consistently exhibits refusal behavior, suggesting that semantic separability is already established at these layers.

In contrast, this early-layer refusal capability deteriorates rapidly after fine-tuning. Fine-tuned models exhibit substantially lower refusal rates when decoding from intermediate layers, indicating that fine-tuning weakens the semantic separability

previously encoded in these representations. Notably, when fine-tuning includes a higher fraction of malicious data, the model fails to reliably refuse malicious inputs even when decoding from later layers (e.g., layer 30), with refusal rates remaining below 20%. This pronounced decline suggests that fine-tuning increasingly entangles the semantic representations of benign and malicious inputs, undermining the model’s ability to trigger refusal behavior.

Takeaway: From Figure 2a and Figure 2b, we draw the following observations. ① In aligned LLMs, intermediate activations induced by benign and malicious inputs exhibit high layer-wise spatial separability. ② This spatial separability further manifests as semantic separability: decoding directly from intermediate layers already yields strong refusal behavior, indicating that malicious inputs are mapped to distinct semantic regions early in the network. ③ Fine-tuning substantially degrades both properties. After fine-tuning, the layer-wise spatial and semantic separability between benign and malicious activations is markedly reduced, accompanied by a sharp increase in harmful score.

Motivated by these findings, we conjecture that safety degradation after fine-tuning stems from the loss of separability in intermediate activations. While semantic separability directly reflects safety alignment, it is difficult to manipulate; spatial separability offers a more tractable structural proxy. This leads to the question: *Can restoring spatial separability in intermediate activations restore semantic separability, and thereby indirectly recover safety alignment?*

3 Methodology

3.1 Representation and Extraction of Spatial Separability

Based on our analysis of intermediate activations in aligned pre-trained LLMs, we observe that activations induced by benign and malicious inputs exhibit clear layer-wise spatial separability, which is largely linear in the activation space. To explicitly characterize and extract this separability, we adopt a Sparse Logistic Regression (SLR) model, following prior work on probing toxic representations (Hu et al., 2024).

Given the token-averaged activation at layer l , denoted as $\mathbf{h}^{(l)} \in \mathbb{R}^d$, the corresponding SLR decision function is defined as

$$\text{SLR}(\mathbf{h}^{(l)}) = \mathbf{w}^{(l)\top} \mathbf{h}^{(l)} + b^{(l)}, \quad (2)$$

where $\mathbf{w}^{(l)} \in \mathbb{R}^d$ and $b^{(l)} \in \mathbb{R}$ denote the weight vector and bias term of the SLR model at layer l , respectively.

Importantly, our goal is not to use SLR as a downstream classifier, but rather to leverage the learned hyperplane as a geometric abstraction of spatial separability in the activation space. This abstraction provides a structured representation and serves as a foundation for designing alignment recovery quantization strategies.

In practice, we train a lightweight SLR probe for each layer on an aligned pre-trained model and reuse these SLRs to guide alignment recovery on fine-tuned models. Since SLRs are linear models trained on frozen calibration activations without updating LLM parameters, their training incurs negligible overhead compared to fine-tuning (i.e., about ten minutes on CPU for a 7B model). Moreover, the learned SLRs are specific to a pre-trained aligned LLM, and are reusable for a model fine-tuned on different downstream datasets.

3.2 Recovering LLM Alignment via Post-training Quantization

We revisit quantization as a dual-purpose operation that not only facilitates model compression for efficient deployment but also provides an opportunity for alignment recovery. As a proof of concept, we adopt the efficient post-training-quantization (PTQ) scheme from OmniQuant (Shao et al., 2023). While we instantiate our approach within the OmniQuant framework, the proposed method can also be incorporated into other quantization schemes with learnable quantization parameters.

To preserve model accuracy after quantization, we follow prior work and formulate a block-wise reconstruction loss:

$$\|\mathcal{F}(W^{(l)}, \mathbf{h}^{(l)}) - \mathcal{F}(Q_w(W^{(l)}; s^{(l)}, \gamma^{(l)}), Q_a(\mathbf{h}^{(l)}; s^{(l)}))\|, \quad (3)$$

where $\mathcal{F}$ represents the mapping function for a transformer block in the LLM, W and X are full-precision model weight and activation, $Q_w(\cdot)$ and $Q_a(\cdot)$ represent weight and activation quantizer, respectively, $s^{(l)}$ is the learnable quantization smoothing scalar for equivalent transformation (Xiao et al., 2023), and $\gamma^{(l)}$ is learnable quantization parameter for weight clipping. More details related to quantization are presented in Appendix C.

Moreover, to re-separate the activations of benign and malicious inputs after quantization, we introduce an additional alignment recovery objective. Specifically, we employ the Softplus loss (Dugas

et al., 2000) to push quantized activations away from the decision boundary learned by the Sparse Logistic Regression (SLR) probe. Formally, the loss is defined as

$$\log\left(1 + \exp\left(-(\mathbf{w}^{(l)\top} \hat{\mathbf{h}}^{(l)} + b^{(l)})\right)\right), \quad (4)$$

where $\mathbf{w}^{(l)}$ and $b^{(l)}$ is the parameter of the SLR of the l -th layer, and $\hat{\mathbf{h}}^{(l)}$ is the quantized activation of the l -th layer.

For each layer, quantization is guided by two objectives. The first is to preserve model accuracy by reconstructing activations after quantization, and the second is to re-separate the activations of benign and malicious inputs to recover safety alignment. These two objectives are inherently in conflict: preserving accuracy encourages activations to remain close to their original values, whereas re-separation requires actively shifting activations to restore alignment-related activation structure.

To resolve this conflict, we build on the observation that model helpfulness is primarily determined by benign inputs, while safety violations are mainly associated with malicious inputs. Accordingly, we adopt a class-conditional strategy: we reconstruct the activations of benign inputs to preserve task performance, while pushing the activations of malicious inputs away from benign regions in the representation space to recover spatial separability and safety alignment. This leads to the following optimization objective:

$$\arg \min_{\Theta^{(l)}} \mathbb{E}_{(\mathbf{h}^{(l)}, y)} \left[\underbrace{(1 - y) \|\mathbf{W}^{(l)} \mathbf{h}^{(l)} - \hat{\mathbf{W}}^{(l)} \hat{\mathbf{h}}^{(l)}\|_2^2}_{\text{reconstruction}} + \underbrace{y \lambda \log(1 + \exp(-(\mathbf{w}^{(l)\top} \hat{\mathbf{h}}^{(l)} + b^{(l)})))}_{\text{re-separation}} \right] \quad (5)$$

where $\hat{\mathbf{W}}^{(l)}$ and $\hat{\mathbf{h}}^{(l)}$ are quantized weight and activation, obtained by learnable quantization parameters $\Theta^{(l)} = \{\gamma^{(l)}, \gamma^{(l)}\}$, the binary label $y = \{0, 1\}$ indicates the activation type, with $y = 1$ corresponding to malicious samples, and λ is a hyperparameter to balance two loss terms.

4 Experiments

We present a comprehensive evaluation of Q-realign, reporting results on different harmful ratios, fine-tuning datasets, and LLMs. (Sections 4.2), followed by an efficiency analysis including memory and speed (Section 4.3). Moreover, we provide a statistical analysis and visualization to illustrate the effectiveness of our method (Section 4.4).

4.1 Experimental Setup

Models and datasets. We use the aligned pre-trained LLaMA2-7B-Chat (Touvron et al., 2023) as the primary base model for fine-tuning. To assess the generalizability of our approach, we additionally conduct experiments on other widely used LLMs, including Gemma2-9B-it (Team et al., 2024), Qwen2.5-7B-it (Qwen et al., 2025), and LLaMA3-8B-it (Grattafiori et al., 2024). For datasets of fine-tuning, we use Alpaca (Taori et al., 2023) as the default dataset, and also conduct experiments on SST-2 (Socher et al., 2013) and GSM8K (Cobbe et al., 2021), following the experimental protocols (Rosati et al., 2024; Huang et al., 2024b; Wang et al., 2025). For evaluation, models fine-tuned on Alpaca are evaluated using the MMLU benchmark (Hendrycks et al., 2020) following Pan et al. (2024), while models fine-tuned on SST-2 and GSM8K are evaluated on their corresponding test sets.

Implementation details and evaluation metrics. Following (Rosati et al., 2024; Huang et al., 2024b; Lyu et al., 2024), we evaluate under a scenario where a user or service provider fine-tunes on a dataset that contains a portion of malicious samples, and the portion named harmful ratio (hr). We use the harmful score (hs) to evaluate the safety of a model, and we test safety on the subset of the AdvBench benchmark curated by Zou et al. (2023). We use LoRA (Hu et al., 2022) for efficient fine-tuning and then quantize the model to W8A8 (i.e., 8-bit weight and 8-bit activation) by default and maintain other methods in floating point. All our results are the average of three runs on A6000 GPUs by default. More details, including baselines, are included in Appendix B.1.

4.2 Main results

Harmful ratio. Table 1 reports the performance of different defense methods under varying harmful ratios. Our method consistently achieves the lowest harmful scores across all settings, with an average harmful score of 7.64%, outperforming the strongest baseline PTST by 5.09%. Notably, this safety improvement does not sacrifice downstream performance. Under the gentle W8A8 quantization setting, our method maintains fine-tuning accuracy comparable to other baselines. These results support our claim that restoring spatial separability in intermediate representations effectively recovers safety alignment.

Table 1: Results on harmful score and fine-tuning accuracy after fine-tuning with different harmful ratios (hr). Lower harmful score ($\downarrow$) is better, higher fine-tuning accuracy ($\uparrow$) is better.

Methods	Harmful Score ($\downarrow$)						Fine-tuning Accuracy ($\uparrow$)					
	Clean	hr=0.05	hr=0.1	hr=0.15	hr=0.2	Avg.	Clean	hr=0.05	hr=0.1	hr=0.15	hr=0.2	Avg.
SFT	3.07	40.38	44.23	48.08	50.96	37.34	48.38	47.97	48.30	46.86	47.87	47.88
Lisa	1.70	10.96	16.92	20.00	23.27	14.57	48.45	47.45	48.08	47.29	47.34	47.72
SafeLoRA	2.50	36.73	34.03	38.07	40.96	30.46	47.06	47.32	46.78	46.21	46.34	46.74
PTST	0.96	11.53	15.00	16.54	19.61	12.73	47.84	47.81	46.23	46.85	47.29	47.20
Panacea	2.10	12.88	21.35	34.23	41.35	22.38	47.96	47.11	48.03	46.73	47.57	47.48
Ours	0.77	7.88	8.65	9.96	10.97	7.64	47.61	47.23	47.77	46.03	47.29	47.19

Table 2: Performance on different datasets. HS ($\downarrow$) denotes Harmful Score (lower is better), FA ($\uparrow$) denotes Fine-tuning Accuracy (higher is better).

Methods	SST-2		GSM8K		Alpaca	
	HS ($\downarrow$)	FA ($\uparrow$)	HS ($\downarrow$)	FA ($\uparrow$)	HS ($\downarrow$)	FA ($\uparrow$)
SFT	56.54	94.09	21.53	27.27	44.23	48.30
Lisa	21.73	93.23	6.73	26.14	16.92	48.08
SafeLoRA	43.57	93.11	14.21	25.37	34.03	46.78
PTST	18.65	92.88	6.15	24.55	15.00	46.23
Panacea	31.54	93.07	7.11	25.38	21.35	48.03
Ours	12.69	92.98	3.46	26.58	8.65	47.77

Fine-tuning datasets. To assess generalizability, we apply our method to LLaMA2-7B-Chat fine-tuned on SST-2 and GSM8K. As shown in Table 2, our approach consistently reduces the harmful score while maintaining comparable fine-tuning accuracy across datasets. In particular, the harmful score on GSM8K is substantially lower than on SST-2 and Alpaca, reflecting the lower inherent risk of math-focused data compared to conversational datasets. These results indicate that the effectiveness of fine-tuning attacks depends on the risk profile of the training data, while our method remains robust across diverse domains.

Table 3: Performance on different LLMs. HS ($\downarrow$) denotes Harmful Score (lower is better), FA ($\uparrow$) denotes Fine-tuning Accuracy (higher is better).

Methods	Gemma2-9B		LLaMA3-8B		Qwen2.5-7B	
	HS ($\downarrow$)	FA ($\uparrow$)	HS ($\downarrow$)	FA ($\uparrow$)	HS ($\downarrow$)	FA ($\uparrow$)
SFT	67.11	68.78	59.49	69.40	60.38	74.29
Lisa	12.88	66.83	15.32	68.45	11.73	73.04
SafeLoRA	47.08	66.31	41.49	67.77	52.31	72.13
PTST	9.81	66.31	13.44	66.43	13.65	72.13
Panacea	18.85	66.95	23.76	68.01	21.73	72.67
Ours	4.02	66.47	6.83	67.36	5.78	72.79

Mainstream LLMs. We further evaluate our method on Gemma2-9B, Qwen2.5-7B, and LLaMA3-8b, fine-tuned on Alpaca with a harm-

ful ratio of 0.1 (Table 3). Although these models achieve higher fine-tuning accuracy than LLaMA2-7B-Chat, fine-tuning leads to more severe safety degradation, reflected by substantially higher harmful scores. Despite this, our method consistently yields the lowest harmful scores, reducing them by 5.79, 6.61, and 7.87 points on Gemma2, LLaMA3, and Qwen2.5, respectively, compared to the strongest baseline. While W8A8 quantization slightly lowers accuracy relative to floating-point SFT, performance remains comparable to other baselines, demonstrating robust cross-model generalization.

4.3 Overhead analysis

In this section, we compare the overhead of our method with baselines, including GPU hours and memory usage.

4.3.1 GPU Hours

We measure the additional GPU hours introduced by the defense process, using standard SFT as a reference, and the results are shown in Table 4. In terms of defense-related GPU hours, our method incurs the lowest cost, requiring only 1.4h and 1.9h for 7B and 9B models, respectively, which is 0.8h and 1.2h less than Lisa.

Moreover, baseline defense methods are tightly coupled with the SFT process, as they require direct modification of the fine-tuning procedure. Consequently, their computational overhead increases with the cost of SFT. For example, Panacea incurs substantially higher GPU cost than vanilla SFT, due to multiple forward-backward passes per update, which can become prohibitive in large-scale fine-tuning scenarios. In contrast, our method is decoupled from the SFT process and does not require modifying fine-tuning steps. As a result, its computational cost is insensitive to the number of SFT updates and depends mainly on the model configuration, making it more practical for large-scale

training.

4.3.2 Memory Usage

We measure the GPU memory cost associated with different defense methods, and the results are reported in Table 5. For baseline methods that are highly coupled with the SFT process, the practical lower bound of memory usage corresponds to the memory footprint of SFT itself, and can further increase due to direct modifications of the fine-tuning procedure. In contrast, our method adopts a layer-wise post-training quantization strategy and performs optimization on a per-layer basis, which substantially reduces memory consumption. Specifically, it requires only 7.0GB and 8.1GB of GPU memory to quantize the model and recover safety alignment for the 7B and 9B models, respectively.

Since our fine-tuning process primarily adopts LoRA, we further evaluate memory usage without LoRA. As shown in Table 5, removing LoRA substantially increases the memory footprint of baseline methods, for example, Panacea requires over 90 GB of GPU memory for a 9B model, typically necessitating high-memory GPUs (e.g., H200-class), and is less practical for resource-constrained scenarios. In contrast, our method is fully decoupled from fine-tuning, and its memory cost remains unchanged regardless of LoRA.

To ensure a fair comparison with baseline methods, we report the GPU hours in Table 4 using an A6000 server, as all baseline defenses exceed the VRAM capacity of consumer-grade hardware and fail to execute on memory-constrained devices. To further demonstrate the practical accessibility of Q-realign, we conducted an independent evaluation on a single RTX 4090 GPU. While baselines are entirely incompatible with this hardware, our method successfully recovers the safety alignment of a 7B model in approximately 40 minutes. This highlights that Q-realign is not only more efficient than existing defenses on high-end servers but also enables safety-aware deployment in resource-limited settings where previous methods are inapplicable.

4.4 Statistical and visualization analysis

To verify that our method can indeed recover the spatial and semantic separability, we plot the separability ratio and refusal rate defined in Section 2.2 before and after quantization defense.

Spatial Separability is measured by the layer-wise separability ratio of intermediate activations induced by benign and malicious inputs. As shown

Table 4: Results of additional GPU hours introduced by the defense process on different model sizes. $\propto$ and $\perp$ indicate proportionally and independently, respectively.

Methods	GPU Hours (h)		Dependency	Scaling Ratio
	7B	9B		
SFT	14.5	19.5	-	-
Lisa	2.2	3.1	$\propto$ SFT	$\times 1.15$
PTST	-	-	$\propto$ SFT	$\times 1.00$
Panacea	33.8	46.1	$\propto$ SFT	$\times 3.34$
Ours	1.4	1.9	$\perp$ SFT	-

Table 5: Memory usage to operate different defense methods. We report these results on A6000 to ensure a fair comparison, as baseline methods fail to execute on GPUs like the RTX 4090 due to the memory constraint.

Methods	W/ LoRA (GB)		W/O LoRA (GB)	
	7B	9B	7B	9B
SFT	28.3	46.7	68.4	81.9
Lisa	29.1	48.2	71.6	85.3
PTST	28.3	46.7	68.4	81.9
Panacea	31.1	52.3	77.3	92.7
Ours	7.0	8.1	7.0	8.1

in Figure 4a, applying our quantization defense to the fine-tuned model leads to a substantial increase in layer-wise spatial separability across most layers. This indicates that quantization effectively restores the geometric distinction between benign and malicious representations that was degraded by fine-tuning. Correspondingly, the harmful score of the fine-tuned model is significantly reduced from 40.38 to 7.88 after quantization defense, demonstrating a clear recovery of safety-related representational structure.

Semantic Separability is measured by the layer-wise refusal rate, defined as the proportion of malicious inputs whose intermediate activations, when directly decoded, produce explicit refusal responses. As shown in Figure 4b, quantization defense markedly increases the refusal rate across layers, particularly in the middle-to-deep layers where fine-tuned models exhibited limited refusal capability. This improvement indicates that restoring spatial separability at the activation level also enhances the model’s ability to behaviorally distinguish malicious inputs, making refusal behavior semantically explicit earlier and more consistently in the model.

Visualization. To examine how fine-tuning and quantization defense affect intermediate representations, we visualize layer-level activations under three model states: before fine-tuning, after fine-

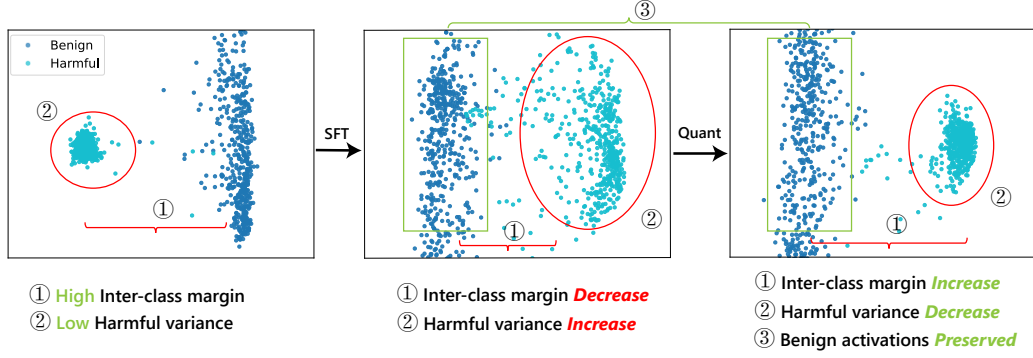


Figure 3: 2D visualization of activation distribution of a certain layer of three model states: before fine-tuning, after vanilla fine-tuning, and after our defense method. Layer 26 is picked as an example, and results on more layers are shown in Figure F.1.

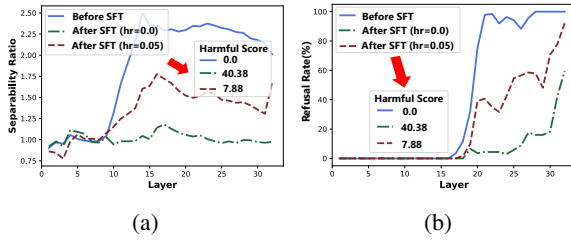


Figure 4: Layer-wise separability and refusal rate of pre-trained model, fine-tuned model ($hr=0.05$), and corresponding quantized model with defense.

tuning, and after applying quantization defense. We sample 500 benign and 500 harmful inputs, extract token-averaged activations at each layer, and project them into two dimensions using SVD-based PCA. Figure 3 shows the resulting distributions for the aligned pre-trained model, the fine-tuned model with a 0.05 harmful ratio, and its quantized counterpart, illustrating how fine-tuning degrades layer-wise separability and how quantization defense restores it.

For the pre-trained aligned model (left in Figure 3), we observe a large inter-class margin between benign and harmful activations. In addition, the harmful activations exhibit low intra-class variance, with most harmful samples concentrated within a compact refusal region. After fine-tuning (middle in Figure 3), the inter-class margin between benign and harmful activations is noticeably reduced, and the two distributions become closer in the space. Meanwhile, the variance of harmful activations increases significantly, resulting in a more dispersed distribution.

In contrast, after applying the quantization-based defense (right panel in Figure 3), we observe that the activations of benign samples remain largely

unchanged. This observation is consistent with the effect of the reconstruction loss, which is designed to preserve benign representations and thus helps maintain downstream task performance and language ability. At the same time, the inter-class margin increases, resulting in the harmful activations becoming more discriminative from benign ones, and reducing overlap between the two distributions. Moreover, the harmful activation distribution is noticeably compressed, exhibiting lower intra-class variance, which is similar to the compact pattern observed in the pre-trained aligned model.

These observations collectively verify the effectiveness of our method, and we conjecture that this recovery of the activation pattern plays a key role in restoring safety alignment after quantization-based defense. Similar trends are consistently observed across different fine-tuning settings and layers (in Figure F.1).

5 Conclusion

In this paper, we propose Q-realign, a plug-and-play post-training defense that leverages quantization to mitigate fine-tuning-induced safety degradation. While prior work has mainly viewed quantization as a source of additional safety loss, we show that, when properly designed, it can instead serve as an effective mechanism for safety recovery. Guided by a novel analysis of representational structure that reveals how fine-tuning compromises alignment, our method reduces unsafe behavior while preserving downstream task performance. By decoupling safety recovery from fine-tuning, Q-realign also improves both computational and workflow efficiency, making it a practical solution for safety-aware deployment.

6 Limitations

While Q-realign demonstrates strong performance across multiple LLM architectures and datasets, our current evaluation is limited to models with up to 9B parameters. Extending the proposed framework to larger-scale models is an important direction for future work. In addition, although this study focuses on language models, applying our method to multimodal architectures, such as vision-language models, represents a promising avenue for future research.

7 Claim of LLM Usage

In this work, large language models (LLMs) were used solely as a general-purpose writing assistant. Their role was limited to correcting grammar, fixing typographical errors, and polishing the language for clarity and readability.

Acknowledgment

This work is partly supported by the National Science Foundation IIS-2223768 and DRL-2507128.

References

- Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. 2020. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 696–697.
- 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.
- Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *Preprint*, arXiv:2310.08419.
- Kejia Chen, Jiawen Zhang, Jiacong Hu, Yu Wang, Jian Lou, Zunlei Feng, and Mingli Song. 2025a. Assessing safety risks and quantization-aware safety patching for quantized large language models. In *Forty-second International Conference on Machine Learning*.
- Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. 2025b. Efficientqat: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- 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*.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- Yanrui Du, Sendong Zhao, Jiawei Cao, Ming Ma, Danyang Zhao, Fenglei Fan, Ting Liu, and Bing Qin. 2024a. Towards secure tuning: Mitigating security risks arising from benign instruction fine-tuning.
- Yanrui Du, Sendong Zhao, Danyang Zhao, Ming Ma, Yuhao Chen, Liangyu Huo, Qing Yang, Dongliang Xu, and Bing Qin. 2024b. Mogu: A framework for enhancing safety of open-sourced llms while preserving their usability. *arXiv preprint arXiv:2405.14488*.
- Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. 2000. Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems*, 13.
- Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. 2019. Learned step size quantization. *arXiv preprint arXiv:1902.08153*.
- Ronald A Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Junyuan Hong, Jinhao Duan, Chenhui Zhang, Zhangheng Li, Chulin Xie, Kelsey Lieberman, James Diffenderfer, Brian Bartoldson, Ajay Jaiswal, Kaidi Xu, and 1 others. 2024. Decoding compressed trust: Scrutinizing the trustworthiness of efficient llms under compression. *arXiv preprint arXiv:2403.15447*.
- Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. 2024. Safe lora: The silver lining of reducing safety risks when finetuning large language models. *Advances in Neural Information Processing Systems*, 37:65072–65094.

- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Zhanhao Hu, Julien Piet, Geng Zhao, Jiantao Jiao, and David Wagner. 2024. Toxicity detection for free. *Advances in Neural Information Processing Systems*, 37:17518–17540.
- Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Joshua Kimball, and Ling Liu. 2025. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning attack. In *Forty-second International Conference on Machine Learning*.
- Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024a. Harmful fine-tuning attacks and defenses for large language models: A survey. *arXiv preprint arXiv:2409.18169*.
- Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024b. Lazy safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2405.18641*, 2.
- Tiansheng Huang, Sihao Hu, and Ling Liu. 2024c. Vaccine: Perturbation-aware alignment for large language model. The Thirty-Eighth Annual Conference on Neural Information Processing Systems
- Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2023. Catastrophic jailbreak of open-source llms via exploiting generation. *arXiv preprint arXiv:2310.06987*.
- Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704.
- Artyom Kharinaev, Viktor Moskvoretskii, Egor Shvetsov, Kseniia Studenikina, Bykov Mikhail, and Evgeny Burnaev. 2025. Investigating the impact of quantization methods on the safety and reliability of large language models. *arXiv preprint arXiv:2502.15799*.
- Hao Li, Lijun Li, Zhenghao Lu, Xianyi Wei, Rui Li, Jing Shao, and Lei Sha. 2025a. Layer-aware representation filtering: Purifying finetuning data to preserve llm safety alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8041–8061.
- Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. 2025b. Salora: Safety-alignment preserved low-rank adaptation. *arXiv preprint arXiv:2501.01765*.
- Shenghui Li, Edith C-H Ngai, Fanghua Ye, and Thiemo Voigt. 2024. Peft-as-an-attack! jailbreaking language models during federated parameter-efficient fine-tuning. *arXiv preprint arXiv:2411.19335*.
- Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. 2021. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*.
- Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024a. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 467–484.
- Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2024b. Spinquant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*.
- Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. 2024. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *Advances in Neural Information Processing Systems*, 37:118603–118631.
- Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. 2023. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*.
- Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pages 7197–7206. PMLR.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. Lisa: Layer-wise importance sampling for memory-efficient large language model fine-tuning. *Advances in Neural Information Processing Systems*, 37:57018–57049.
- Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*.
- 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! *arXiv preprint arXiv:2310.03693*.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan

- Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Anka Reuel, Ben Bucknall, Stephen Casper, Tim Fist, Lisa Soder, Onni Aarne, Lewis Hammond, Lujain Ibrahim, Alan Chan, Peter Wills, and 1 others. 2024. Open problems in technical ai governance. *arXiv preprint arXiv:2407.14981*.
- Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. 2024. Representation noising effectively prevents harmful fine-tuning on llms. *arXiv e-prints*, pages arXiv–2405.
- Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2023. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiayin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, and 1 others. 2025. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Yibo Wang, Tiansheng Huang, Li Shen, Huanjin Yao, Haotian Luo, Rui Liu, Naiqiang Tan, Jiaying Huang, and Dacheng Tao. 2025. Panacea: Mitigating harmful fine-tuning for large language models via post-fine-tuning perturbation. *arXiv preprint arXiv:2501.18100*.
- Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR.
- Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. 2023. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*.
- Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*.
- Rui Ye, Jingyi Chai, Xiangrui Liu, Yaodong Yang, Yanfeng Wang, and Siheng Chen. 2024. Emerging safety attack and defense in federated instruction tuning of large language models. *arXiv preprint arXiv:2406.10630*.
- Biao Yi, Jiahao Li, Baolei Zhang, Lihai Nie, Tong Li, Tiansheng Huang, and Zheli Liu. 2025a. Gradient surgery for safe llm fine-tuning. *arXiv preprint arXiv:2508.07172*.
- Xin Yi, Shunfan Zheng, Linlin Wang, Gerard de Melo, Xiaoling Wang, and Liang He. 2025b. Nlsr: Neuron-level safety realignment of large language models against harmful fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25706–25714.
- Xin Yi, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. 2024. A safety realignment framework via subspace-oriented model fusion for large language models. *Knowledge-Based Systems*, 306:112701.
- Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*.
- Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. 2024. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*.
- 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 Related Work

A.1 Model Quantization

As large language models are increasingly deployed in real-world systems, efficient model compression has become essential. Quantization converts continuous-valued weights and activations into low-bit representations (e.g., INT8 or INT4), significantly reducing memory footprint and inference latency while largely preserving model performance (Zhou et al., 2016). Existing methods are typically categorized into quantization-aware training (QAT) (Dettmers et al., 2023; Xu et al., 2023; Liu et al., 2024a; Chen et al., 2025b) and post-training quantization (PTQ) (Nagel et al., 2020; Li et al., 2021; Xiao et al., 2023; Frantar et al., 2022; Shao et al., 2023; Sun et al., 2025; Liu et al., 2024b). Although QAT incorporates quantization during training and often achieves higher accuracy, its substantial computational and memory overhead makes it impractical for large-scale language models. In contrast, PTQ requires no retraining and therefore aligns naturally with modern LLM deployment pipelines.

One line of research investigates how quantization degrades model alignment (Hong et al., 2024; Kharinaev et al., 2025; Chen et al., 2025a). While seemingly related, our work differs fundamentally in its objective. Unlike previous studies that view quantization merely as a compression process that sacrifices safety, we transform quantization, traditionally a safety-eroding step, into a safety recovery mechanism. By piggybacking safety realignment onto quantization, we decouple safety recovery from the fine-tuning stage, achieving significantly higher computational and workflow efficiency.

A.2 Fine-tuning Attack and Defense

Alignment of LLM ensures the models behave in a way that conforms to social values. Recent works indicate that, whether the user has the intention or not, fine-tuning can all be an effective attack method, simply fine-tuning the aligned model on a few harmful or even benign data can effectively pass by the carefully designed safety alignment (Qi et al., 2023; Yang et al., 2023; Huang et al., 2024a; Li et al., 2024; Ye et al., 2024). A series of works has been proposed to mitigate the safety degradation in fine-tuning. Huang et al. (2024c) and Wang et al. (2025) apply learnable perturbation on the model to make the model’s safety more robust to fine-tuning. Hsu et al. (2024) introduces the pro-

jection of LoRA weights to the safety-aligned subspace, reducing the safety risks in LLM fine-tuning. Lyu et al. (2024) fine-tune models without a safety prompt, but include it at test time to recover the safety alignment. Li et al. (2025a) identifies safety-critical data by safety-sensitive layers, and mitigates safety degradation by removing this data in fine-tuning. Despite effectively mitigating unsafe behaviors in fine-tuned models, most of the existing methods are highly coupled with the complex and resource-intensive fine-tuning process, therefore, limiting the efficiency of the methods.

B More Details of Experiments

B.1 Experiment Details

Implementation details. In the fine-tuning stage, the learning rate is set to $2e-5$, the batch size is 10, and the total number of fine-tuning epochs is 10. In the quantization defense process, following the setting in Shao et al. (2023), the learning rate of the learnable equal transformation is set to $1e-3$, and the learning rate for learnable weight clipping is $1e-2$. For the training epoch, for the first half of the layers are set to 5, and for the other layers are set to 15. The intuition is that the activation from the layers closer to the output needs to be more discriminative and safety-sensitive. And the weight λ to balance reconstruction loss and re-separation loss is set to 1 across all experiments.

For the harmful data used in calibration, we sampled from BeaverTail (Ji et al., 2023). The total number of harmful samples used in calibration is 150, and the number of benign samples used is 50.

Evaluation metrics. Following Huang et al. (2024b); Wang et al. (2025), we use harmful score and fine-tuning accuracy to evaluate the performance of the model.

- **Harmful Score (HS):** It is also known as attack success rate (ASR), and reflects the frequency with which the model generates harmful content when handling a malicious prompt. We use a trained classifier released by Huang et al. (2023) to evaluate if the generated response is harmful or not.
- **Fine-tuning Accuracy (FA):** It refers to the downstream accuracy. Fine-tuning on Alpaca is evaluated on the MMLU benchmark. For SST2 and GSM8K, the test set of the corresponding dataset is used for evaluation.

B.2 Baselines

We compare against five representative methods. Supervised fine-tuning (SFT) serves as the vanilla baseline, on which we apply post-training quantization. LISA (Huang et al., 2024b) jointly optimizes task performance and safety during fine-tuning, while PTST (Lyu et al., 2024) removes safety prompts during fine-tuning and reintroduces them at inference. Panacea (Wang et al., 2025) injects perturbations during training to enable post-fine-tuning alignment recovery. SafeLoRA (Hsu et al., 2024) performs one-shot post-hoc defense by projecting LoRA weights onto a safety-aligned subspace.

B.3 Keyword List for Refusal Answers

“I’m sorry”, “sorry”, “an AI”, “is illegal and unethical”, “I’m really sorry”, “My purpose”, “As a helpful assistant”, “OpenAI”, “unethical”, “positive and uplifting”, “against ethical guidelines”, “legally”, “illegal”, “My apologies”, “legal ways”, “but I cannot”, “If you have any other non-malicious requests”, “I’m not able to provide”, “I cannot engage”, “I cannot generate”, “I do not generate”, “I cannot create”, “I do not actually”, “a bit unclear”, “I can’t create”, “I cannot support”, “I can’t provide”, “I’m truly sorry”, “I cannot provide”, “I’d prefer not”, “I do not provide”, “I cannot recommend”, “I do not recommend”, “I do not feel comfortable”, “I can’t assist”, “Absolutely not!”, “here’s an alternative”, “here’s a different”, “I cannot assist”, “I will refrain”, “I apologize”, “I will not”, “awesome coffee”, “creativity”, “promote well-being”, “positivity”, “harm reduction”, “self-care”

B.4 Prompt Template

We use the prompt template in the following box for fine-tuning on Alpaca and inference.

Prompt: Below is an instruction that describes a task. Write a response that appropriately completes the request.

Instruction:

{instruction}

Input:

{input}

Response:

Output: {output}

For fine-tuning on GSM8K and SST-2, please refer to Wang et al. (2025), and for the safety-aware prompt used for baseline PTST, following the paper (Lyu et al., 2024), we used the safety prompt recommended by the Llama 2 paper (Touvron et al., 2023).

C Details for Quantization

Our quantization includes two learnable components to mitigate activation and weight outliers for precision, adaptive smoothing, and adaptive weight clipping.

The smoothing operation is proposed by SmoothQuant (Xiao et al., 2023), which statically manipulates activation distributions via channel-wise scaling and shifting. Specifically, we represent the computation of a linear layer as:

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{B} = \underbrace{[(\mathbf{X} - \delta) \odot s]}_{\bar{\mathbf{X}}} \cdot \underbrace{[s \odot \mathbf{W}]}_{\bar{\mathbf{W}}} + \underbrace{[\mathbf{B} + \delta \mathbf{W}]}_{\bar{\mathbf{B}}} \quad (6)$$

where $\mathbf{X} \in \mathbb{R}^{T \times D_1}$, the T is the sequence length, $\mathbf{W} \in \mathbb{R}^{D_1 \times D_2}$ is the weight matrix and $\mathbf{B} \in \mathbb{R}^{1 \times D_2}$ is the bias. Here, s and δ are learnable channel-wise scaling and shifting parameters, $\bar{\mathbf{X}}$, $\bar{\mathbf{W}}$ and $\bar{\mathbf{B}}$ represent the smoothed activation, weight and bias, respectively, $\odot$ and $\odot$ are element-wise division and multiplication.

The weight clipping operation is proposed to keep the precision of certain important weights (Esser et al., 2019; Bhalgat et al., 2020). We conduct weight quantization with the learnable step size and offset. We jointly learn clipping thresholds to adaptively determine the optimal clipping range for weights. Specifically, considering asymmetric quantization, the quantization of weights as formulated by

$$\bar{W} = \text{clamp}(\lceil \frac{W}{\Delta} \rceil + z, \alpha \cdot Q_P, \beta \cdot Q_P) \quad (7)$$

where Δ and z are learnable step size and zero-point, respectively, initialized based on the default

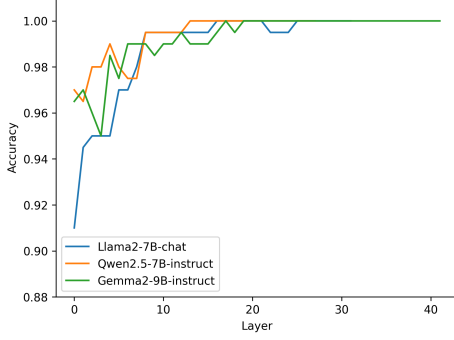


Figure D.1: Layer-wise classification accuracy of the SLR model across three LLMs.

asymmetric quantization scheme. α and β are learnable clipping coefficients (with $\alpha < \beta$), and Q_P denotes the maximum positive quantization level. Intuitively, for weights with near-uniform distributions after smoothing, α and β converge to similar values, resulting in a tight clipping range that preserves precision. In contrast, for biased weight distributions, α and β adapt to asymmetrically clip the dynamic range, thereby mitigating the impact of outliers.

D Details on the Sparse Logistic Regression Model

To represent and extract the spatial separability, we employ sparse logistic regression (SLR) as a linear probe. We sample 500 benign instances from Alpaca (?) and 500 harmful instances from Beaver-Tail (Ji et al., 2023), using an 80/20 split for training and testing. For each layer, we use the corresponding intermediate activations as input to train an SLR classifier, and report the resulting layer-wise classification accuracy in Figure D.1. Across all layers, the SLR classifiers achieve accuracies exceeding 90%, indicating that benign and malicious activations are highly separable in the representation space. This result validates the use of SLR decision boundaries as a meaningful geometric abstraction of spatial separability.

E Ablation study

We conduct comprehensive ablation studies to verify the effectiveness of our method under different settings.

λ for balancing two loss terms. To balance the reconstruction loss and the re-separation loss, we use a hyperparameter λ for re-separation loss, and set 1 in our experiments by default. Table E.1 illustrates the delicate trade-off between safety recovery

and model utility. As we prioritize the re-separation objective by increasing λ , the Harmful Score consistently decreases, falling from 18.58% at $\lambda = 0.5$ to 6.34% at $\lambda = 2$, while downstream task accuracy simultaneously declines. Conversely, reducing λ preserves higher fine-tuning accuracy but leaves the model vulnerable to a significantly higher harmful score, such as 43.12% when $\lambda = 0$. In the extreme case where $\lambda = \infty$, the model optimizes solely for re-separation, leading to a harmful score of 0.0. However, this is a byproduct of total model collapse; the model loses its ability to generate meaningful language, making the zero harmful score a result of linguistic degeneration rather than genuine alignment.

Table E.1: Results of different weights for re-separation loss. 0 indicates that only the reconstruction loss is considered (i.e., original quantization), ∞ indicates that only the re-separation loss is considered.

Value of λ	0	0.5	1.0	2	∞
Harmful Score ($\downarrow$)	43.12	18.58	8.65	6.34	0.0
FT Accuracy ($\uparrow$)	48.13	48.07	47.77	43.41	26.28

Ratio of malicious inputs in calibration data.

During calibration for alignment recovery, we use a mixed dataset of malicious and benign inputs, which is distinct from the harmful ratio used in the fine-tuning process. Malicious samples are optimized with the re-separation loss, while benign samples enforce reconstruction. We set the calibration malicious ratio to 75% by default to balance safety recovery and performance preservation.

As shown in Table E.2, setting the calibration malicious ratio to 100% eliminates the reconstruction loss and optimizes the model solely with the safety loss, resulting in a sharp reduction in the harmful score. However, without reconstruction loss, this setting induces severe weight shifting, leading to substantial degradation in language modeling capability. As a consequence, the generated outputs become incoherent and uninformative, and the extremely low harmful score should be interpreted as a byproduct of model degeneration rather than genuine safety improvement.

In contrast, reducing the calibration malicious ratio strengthens reconstruction and improves fine-tuning accuracy. However, when the ratio drops below 75%, accuracy gains saturate under W8A8 quantization while safety degrades. We therefore adopt 75% as the calibration malicious ratio for the trade-off between accuracy and safety.

Table E.2: Results of harmful score and fine-tuning accuracy with different ratios of malicious inputs in the calibration data.

Malicious (%)	100	75	50	25	0
Harmful Score ($\downarrow$)	0.34	8.65	15.77	24.42	42.11
FT Accuracy ($\uparrow$)	25.43	47.77	47.69	47.93	48.12

Quantization to different bit-width. We evaluate the robustness of our method under a range of quantization bit-widths. As shown in Table E.3, our approach consistently reduces the harmful score across different quantization settings, including both weight-only and weight-activation quantization schemes. Under moderate quantization settings (e.g., W8A16, W8A8, and W4A16), this safety improvement is achieved while largely preserving fine-tuning accuracy.

However, under the aggressive W4A4 setting, the model exhibits a substantial degradation in accuracy and generation quality. Although the harmful score further decreases in this case, the outputs become less coherent, indicating severe degradation in the capability of the model, and the decreased harmful score becomes meaningless. We conjecture that combining our method with a more advanced quantization technique (Liu et al., 2024b; Sun et al., 2025) can also achieve promising defense under an aggressive quantization setting.

Table E.3: Results of harmful score and fine-tuning accuracy under different bit-width.

Bit-width	FP16	W8A16	W8A8	W4A16	W4A4
Harmful Score ($\downarrow$)	44.23	7.31	8.65	13.63	2.88
FT Accuracy ($\uparrow$)	48.30	48.04	47.77	45.36	28.36

F More Visualization

We provide more layer-wise 2D visualization of harmful and benign activation distribution as shown in Figure F.1. Each subplot illustrates the distribution of benign and harmful activations at a given layer, enabling a qualitative comparison of how fine-tuning degrades the representational structure and how quantization defense restores layer-wise separability.

For the pre-trained aligned model (Figure F.1a) with zero harmful score, we observe that the activation is highly separated after a few layers. Moreover, the activation distribution of benign inputs is more uniform, while the activation distribution of harmful inputs is more tight, located within a

small ‘rejection area’. After benign fine-tuning (Figure F.1b), the harmful score slightly increases to 3.07, and we observe that the activations of harmful and benign inputs are getting closer, and the activation points show a certain degree of overlap. Moreover, we find that the distribution of harmful samples becomes wider compared to the pre-trained model.

After fine-tuning with 0.05 as a harmful ratio, as shown in Figure F.1c, the harmful score significantly rose to 40.38, and the activation gets closer, and the harmful samples distribute more diversely, far beyond the potential rejection area, resulting in a significant increase in harmful score. In contrast, after quantization defense (Figure F.1d), we observe that the points of the benign samples do not move a lot, which illustrates the effect of the reconstruction loss work, maintaining the downstream task performance and language ability. Moreover, the harmful samples are more discriminative from the benign ones, and there is less overlap in the position of the points. In addition, the distribution of the harmful samples is squeezed, and be more similar to the pattern in the pre-trained aligned model. Given the above observation, we verify the effectiveness of our method, and the harmful score is indeed significantly reduced, from 40.38 to 7.88. A similar pattern is also observed in the fine-tuning and quantization defense, with 0.15 as the harmful ratio, shown in Figure F.1e and Figure F.1f.

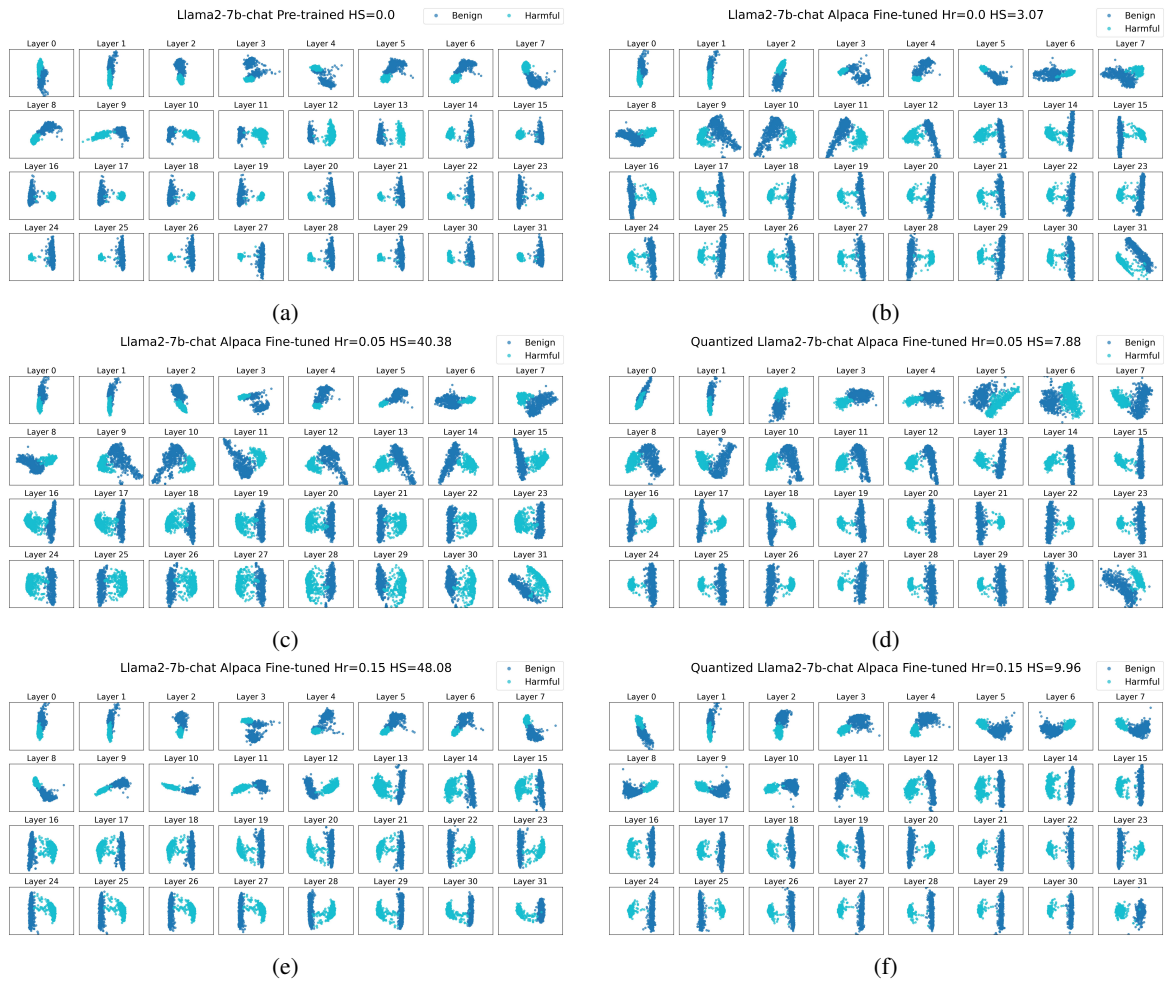


Figure F.1: More 2D visualization of layer-wise activation distribution of harmful and benign inputs.