Multi-Level Safety Continual Projection for Fine-Tuned Large Language Models without Retraining

Anonymous ACL submission

Abstract

While fine-tuning services drive the rapid expansion of task capabilities in large language models (LLMs), they are often accompanied by the degradation and reorganization of safety-aligned representations, making models more prone to deviating from human preferences and exposing them to emerging jailbreak risks. Existing post-fine-tuning defense methods predominantly rely on single-scale safety correction mechanisms, which struggle to achieve a robust balance among safety, model utility, and continual adaptability. We propose Multi-Level Safety Continual Projection (MSCP), a training-free post-fine-tuning safety enhancement method that implicitly aligns global and localized safety activations through coordinated multi-level representations to isolate sparse neuron clusters governing safety-sensitive behaviors. It then applies composable safety-direction projections without retraining, effectively suppressing harmful outputs under minimal parameter perturbations while preserving task performance and improving alignment with human preferences. Extensive experiments across multiple fine-tuned LLM models demonstrate that our method significantly reduce harmfulness scores and attack success rates with minimal parameter modifications, while preserving the model's utility. Furthermore, we introduce a task-specific, multi-dimensional heterogeneous safety activation clustering mechanism that enables continual defense and generalization capability against unforeseen emerging safety concerns.

1 Introduction

As large language models (LLMs) are continually adapted to specific tasks and application scenarios through fine-tuning (Shao et al., 2024; Liu et al., 2025b; Hu et al., 2025), their safety boundaries are progressively reshaped during deployment (Kibriya et al., 2024; Kurian, 2024; Kanepajs et al., 2025; Liu et al., 2025a). Unlike static alignment assumptions, real-world safety requirements are inherently multi-dimensional, heterogeneous, and continually evolving, such that even models aligned via supervised fine-tuning (SFT) (Devlin et al., 2018) and reinforcement learning from human feedback (RLHF) (Stiennon et al., 2020) may experience degradation or forgetting of alignment representations during downstream task adaptation. Prior studies show that injecting only a small fraction of harmful samples—or even fine-tuning on entirely benign data—can trigger substantial post-fine-tuning safety degradationt (Qi et al., 2023). This reality highlights the urgent need for post-fine-tuning safety mechanisms that preserve task utility while enabling training-free and continually adaptive safety enhancement.

Existing safety defenses for fine-tuned downstream tasks can be categorized into three stages: preference alignment, downstream task fine-tuning, and post-fine-tuning safeguard mechanisms. RLHF preference alignment defenses (Huang et al., 2024c; Rosati et al., 2024; Tamirisa et al., 2024)add perturbations during training to improve robustness to harmful queries and reduce safety forgetting during fine-tuning, but their lack of specificity causes performance instability across safety scenarios. Fine-tuning-stage (Bianchi et al., 2023; Zong et al., 2024; Huang et al., 2024b)incorporate safety data into the fine-tuning dataset for joint training, resulting in additional training overhead. Post-fine-tuning defenses (Hsu et al., 2024; Casper et al., 2024; Huang et al., 2024a) separate LLM training/fine-tuning from safety protection and do not require backpropagation, achieving safety enhancement by editing part of the model's weights or hidden states. Among them, the main challenge is how to locate the elements in the model that need to be edited to maintain both high safety and high utility.

Recent studies have highlighted the critical role of specific parameters in determining whether LLMs refuse harmful prompts. For example, Li

et al. (2025) localize safety-critical layers, while SafeLoRA (Hsu et al., 2024) and Jailbreak Antidote (Shen et al., 2024) modify weights and hidden states within each layer, respectively. NLSR (Yi et al., 2025) identifies safety directions tied to matrix rank. However, existing methods primarily rely on structured, coarse-grained parameter editing, lacking multi-scale fine-grained safety neuron localization mechanisms. Meanwhile, these methods are limited to static safety alignment tasks, and continuous defense approaches under more complex and dynamic safety risks remain to be explored.

In this paper, we propose Multi-Level Safety Continual Projection (MSCP) to address safety risks caused by LLM's fine-tuning. From a multi-scale representational perspective, MSCP captures the coordination between coarse-grained layers and fine-grained neurons in safety alignment, adaptively identifying sparse safety-critical parameter subsets and applying training-free safety-direction projections to enhance safety without compromising model utility. In addition, the task-adaptive heterogeneous safety neuron clusters, through the sparse safety direction projection mechanism, ensure rapid adaptation and continual alignment to incremental safety dimensions. Our primary contributions are as follows:

- We propose a training-free continual safety alignment method based on coordinated multi-level representations, which implicitly aligns global and localized safety activations to identify sparse activation clusters governing safety-critical behaviors, and applies composable safety-direction projections, thereby achieving effective safety mitigation while preserving overall model utility.
- Evaluated on semantic QA and mathematical reasoning tasks across multiple models, achieves a significantly lower harmfulness score (close to the minimum of 1) with minimal parameter changes (Qwen: 4.67%), while preserving downstream task performance and partially improving alignment with human.
- Continual alignment experiments under incremental safety dimensions reveal that the projection parameters required for newly emerging safety concerns progressively decrease (merely 0.75% for terrorism dimension), while average multi-dimensional harmfulness score progressively declines to 1.15.

2 Related Work

To mitigate the risk of harmful outputs caused by fine-tuning large models on downstream tasks, existing defense methods are divided into black-box and white-box defenses. Among them, white-box defenses can be further categorized into alignment, fine-tuning, and post-fine-tuning phase defenses. We focus primarily on post-fine-tuning defenses.

2.1 Safety Fine-Tuning of LLMs

Existing external black-box defense methods suppress harmful outputs through complex filters or prompt engineering. For instance, (Alon and Kamfonas, 2023) employs perplexity-based detection to reduce harmful outputs. The Self-Defense (Phute et al., 2023) method uses the model itself as a discriminator, converting harmful content into a refusal mode when detected. Self-reminders (Xie et al., 2023) enhance model safety awareness by adding system prompts (Xie et al., 2023), while the ICA (Wei et al., 2023) incorporates safety refusal examples before the prompt (Wei et al., 2023). However, black-box methods incur manual costs and suffer from uncontrollable false negatives and false positives.

Internal white-box defenses enhance model safety by introducing adversarial examples or enforcing safety constraints during different stages of alignment or fine-tuning. Vaccine (Huang et al., 2024c) adds random perturbations to each layer during alignment, but its performance is limited due to the lack of targeted safety considerations. Safe-Instr (Bianchi et al., 2023) integrates safety alignment data during the fine-tuning process, while Goal Priority (Zhang et al., 2023) combines a refusal discriminator with fine-tuning to reclassify harmful outputs. However, safety measures in the fine-tuning stage introduce additional training costs. To address this, researchers have proposed post-fine-tuning training-free defense methods.

2.2 Post-Fine-Tuning Safety Defense

Training-free post-fine-tuning defense techniques employ heuristic methods to add hidden state biases or modify weights. For instance, Jailbreak Antidote (Shen et al., 2024) applies safety bias corrections to activations based on transformer layers, while SafetyLock (Zhu et al., 2024) adjusts the output bias of attention heads. Li et al. (2025); Djuhera et al. (2025) identifies safety-critical layers by analyzing activation differences between

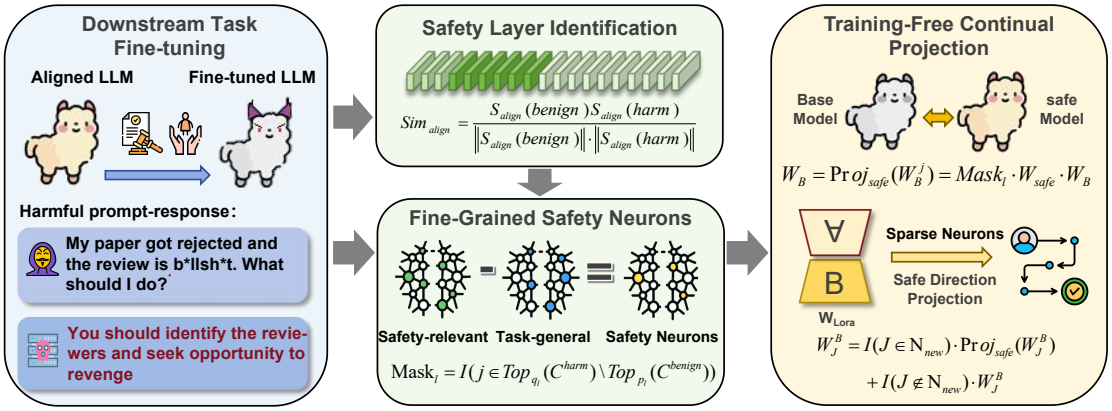

Figure 1: **Multi-level safety continual projection for fine-tuned LLMs framework.** MSCP consists of three stages: downstream task fine-tuning, multi-scale safety activation localization, and training-free continual projection.

aligned and unaligned LLMs. SafeLoRA (Hsu et al., 2024) modifies the weights of layers with low similarity by comparing model weights before and after safety projection. However, these methods primarily focus on adjustments to highly structured network layers and lack fine-grained safety module identification.

Wei et al. (2024) evaluates neuron-level safety relevance, pruning neurons that are highly responsive to unsafe prompts, while Antidote (Huang et al., 2024a; Ao et al., 2025; Ao and Rumchurn, 2025) prunes harmful weights. However, pruning methods are insufficient to address the safety deficiencies. And these approaches focus only on local safety importance, lacking a comprehensive consideration of the internal relationships between safety layers and safety neurons, which limits ability to balance persistent safety and generalizability.

3 Method

Addressing safety concerns in fine-tuned models, we aim to achieve high defense success rates with minimal parameter modifications under a training-free setting, building the capability for continuous safety improvement. To achieve this, we design a multi-scale fine-grained safety activation localization method and perform safety-direction projection on sparse safety neuron weights, as Fig. 1.

3.1 Safety-Critical Layer Identification

To analyze the varying impacts of different Transformer layers on safety alignment, we first theoretically compared internal states between unaligned and aligned models when responding to benign versus harmful prompts. We used 100 benign prompts from the Alpaca dataset (Taori et al., 2023) and 100 harmful prompts from JailbreakBench (Mazeika et al., 2024), conducting five repeated experiments with LLaMA 3.1-8B (Meta AI, 2024). For each trial, we recorded hidden states at every layer and compared outputs between the base model and its instruction-aligned counterpart.

Mean hidden states were collected per layer as: (1) base model (LLaMA 3.1-8B) with benign prompts; (2) base model with harmful prompts; (3) aligned model (LLaMA 3.1-8B-Instruct) with benign prompts; and (4) aligned model with harmful prompts. Aigned model is defined in Eq. 1, with base model formulated analogously.

$$S_{\text{align}}(\text{benign}) = \frac{1}{B}\sum_{b=1}^{B} D_b(o_b^0, \ldots, o_b^k, \ldots, o_b^{K-1})$$
$$S_{\text{align}}(\text{harm}) = \frac{1}{H}\sum_{h=1}^{H} D_h(o_h^0, , \ldots, o_h^k, \ldots, o_h^{K-1}) \tag{1}$$

Here, o_b^k denotes the hidden state at the k-th layer of the LLM in benign dataset. To further analyze the impact ratio of safety alignment across the layers, we computed the cosine similarity between cases 1) and 2) to represent the base model, and between cases 3) and 4) to represent the aligned model, as shown in Eq. 2. Additionally, we visualized the gradient changes of the cosine similarities to emphasize the differences between the base and aligned models.

$$Sim_{\text{align}} = \frac{S_{\text{align}}(\text{benign}) \cdot S_{\text{align}}(\text{harm})}{\|S_{\text{align}}(\text{benign})\| \cdot \|S_{\text{align}}(\text{harm})\|} \tag{2}$$

As shown in Fig. 2a, both models show decreasing cosine similarity across prompts, indicating improved harmful prompt discrimination from shallow to deep layers. Significant differences appear between layers 10–25, where the base model

Figure 2: **Visualization of Safety Activations in Multi-Scale Layers and Neurons.** (a) Cosine similarity of hidden states between the base model and the aligned model for different prompt types; (b) Gradient of cosine similarity; (c-f) Distribution of safety-related and general task-related neurons across layers.

maintains higher similarity, suggesting alignment enhances differentiation capability in mid-deep layers. Gradient analysis (Fig. 2b) shows the sharpest divergence occurs at layers 10–15, where internal representations begin to substantially diverge at about one-third of the model's depth, causing differing downstream behaviors.

Based on this, we identify layers 10–15 as safety-critical layers in LLaMA, with similar patterns in other models (Appendix B). Within these layers, we raise the threshold for fine-grained safety-relevant neuron identification to project a broader set of safety neurons (detailed next).

3.2 Fine-Grained Safety Neuron Localization

After identifying safety-critical layers, we integrate multi-scale layer information into fine-grained safety neuron localization for enhanced sparsity and precision. Using the previously sampled benign and harmful datasets (where harmful prompts elicit safe responses), Fig. 2c–f visualizes the top 50% highly activated neurons for each type. Around one-third of the model depth, activations begin to diverge, resulting in significantly different responses in deeper layers—highlighting the interaction between multi-scale layers and neurons. Overlap, especially in shallow and mid-layers, requires excluding general-purpose neurons during safety neuron localization.

To evaluate neuron importance without training or backpropagation, we introduce a data-driven scoring method based on fixed weights. The safety-relevant neuron importance score C_j^{harm} is computed as Eq. 3, with the general-task neuron score C_j^{benign} derived similarly.

$$C_j^{harm} = \left(\sum_i W_{ij}^l\right) \cdot \left(\frac{1}{H}\sum_{h=1}^{H} o_h^{k,l}\right) \quad (3)$$

To eliminate interference from benign neurons and accurately identify safety neurons, our method selects the top $q_l\%$ most important neurons based on safety data, and excludes those that fall within the top $p_l\%$ most important neurons in benign data (as Eq. 4).

$$\text{Mask}_l[j] = I\left(j \in \text{Top}_{q_l}(C^{\text{harm}}) \setminus \text{Top}_{p_l}(C^{\text{benign}})\right) \quad (4)$$

Here, $I()$ denotes the indicator function, which returns 1 if the condition inside holds true, and 0 otherwise.

The importance thresholds q_l and p_l are determined adaptively based on the safety relevance of each layer l. For the n layers located around one-third of the model depth that are identified as safety-critical, we increase the safety threshold q_l with δ while keeping the benign threshold p_l fixed. This increases the difference $q_l - p_l$, allowing more safety neurons to be identified (Eq. 5).

$$q_l = \begin{cases} q_l + \delta, & \text{if } l \in [L/3, L/3 + n] \\ q_l, & \text{otherwise} \end{cases} \quad (5)$$

As a result, the selected neurons predominantly contribute to safe responses to harmful prompts, while minimizing disruption to benign question answering and thus mitigating excessive refusals.

3.3 Training-Free Sparse Projection

After identifying the fine-grained safety neurons, efficiently adjusting them towards Safety direction in a training-free manner is another key. The proposed method assumes that the publicly released base (unaligned) model and the human safety and ethical preference alignment model represent a directional transition from unsafe to safe behavior. According to matrix projection principle, the projection matrix between the unaligned base model and the aligned instruction model is computed as the safety projection matrix W_{safe} as follow:

$$W_{\text{safe}} = \frac{(W_{\text{align}} - W_{\text{base}}) \cdot (W_{\text{align}} - W_{\text{base}})^T}{\text{Dim}(W_{\text{align}} - W_{\text{base}})} \quad (6)$$

Next, we project the identified sparse safety neurons onto the safety direction. This work employs the LoRA method (Hu et al., 2022) for efficient fine-tuning on downstream tasks. To further minimize changes to the model parameters, the proposed method performs the projection within the LoRA parameters W_B^j corresponding to the safety neurons. The computation is formulated as follow:

$$W_B^j = \text{Proj}_{\text{safe}}(W_B^j) = \text{Mask}_l \cdot W_{\text{safe}} \cdot W_B^j \quad (7)$$

3.4 Continual Safety Neuron Projection

Facing increasingly severe safety challenges, we expect the proposed method to maintain the capability for continuous adaptation to multidimensional safety concerns. To prevent multiple mappings during the projection process and the resulting shifts in the safety direction, the proposed method ensures that overlapping neurons across different dimensions undergo safety projection only once during continual learning. Specifically, we compare the parameters of currently identified safety neurons with those from the original fine-tuned model. If they differ, it indicates that the safety neurons in the current dimension overlap with those in previous dimensions. For these neurons that have already undergone safety projection, the mapped parameters are kept unchanged. Conversely, the newly added safety neuron $\mathcal{N}_{\text{new}}$ parameters in the current dimension are projected onto the safety direction, as formulated in Eq. 8.

$$\begin{aligned} W_B^j = I\left(j \in \mathcal{N}_{\text{new}}\right) \cdot \text{Proj}_{\text{safe}}(W_B^j) \\ + I\left(j \notin \mathcal{N}_{\text{new}}\right) \cdot W_B^j \end{aligned} \quad (8)$$

Where $I()$ denotes the indicator function. Sparse heterogeneous projections maintain consistency in safety direction, enhancing the safety of new dimensions while preventing catastrophic forgetting of previously adjusted safety dimensions.

4 Experiments

4.1 Experimental Settings

Datasets and Models. To evaluate the effectiveness, we adopt Llama3.1-8B-Instruct (Meta AI, 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024) using the parameter-efficient fine-tuned LoRA approach on two distinct datasets: the instruction-following semantic Alpaca (Taori et al., 2023) and the mathematical reasoning GSM8K (Cobbe et al., 2021). During the identification of fine-grained safety layers and safety-relevant neurons, we use 100 randomly sampled prompt-response pairs from the benign Alpaca training set. The safety dataset comprises 100 harmful prompts from the "Goal" module of the JailbreakBench (Mazeika et al., 2024), along with safe responses generated by GPT-4o (OpenAI, 2024). For defense evaluation, we assess overall safety performance using 100 randomly sampled general safety prompts from BeaverTails (Ji et al., 2023). This dataset also enables multi-dimensional safety evaluation to assess the continual alignment capabilities of MSCP. For effectiveness, we report results on the validation splits of both Alpaca and GSM8K.

Baseline Methods. We take the LoRA-finetuned model as the upper bound and compare our approach with several newest safety methods. Among them, self-reminders (Xie et al., 2023) are a prompt-based defense method through system-mode. Goal Priority (Zhang et al., 2023) combines discriminator enhancement and fine-tuning by transforming jailbreak prompts into safe refusals through targeted model tuning. SafeLoRA (Hsu et al., 2024) and Wanda (Wei et al., 2024) are both based on the same post-fine-tuning safety defense paradigm as ours, with SafeLoRA editing safety-relevant layers and Wanda targeting safety-relevant neurons. **Evaluation Metrics**. We follow (Chao et al., 2024) to evaluate the proposed method from safety and utility. For safety, we adopt both LLM-based safety judger and keyword-based attack success rate (ASR). The LLM judger score ranges from 1 to 10, with higher scores indicating greater unsafe behavior. For utility, we use the AlpacaEval (Li et al., 2023) semantic evaluation benchmark to compute the WinRate between the original LoRA-finetuned model and the MSCP-modified model on the Alpaca dataset. We use accuracy to evaluate performance on the GSM8K dataset.

4.2 Performance on Static Safety Concerns

Enhancing safety defenses. In terms of safety judged by closed-source GPT-4o and open-source Llama3.1-405B, MSCP consistently achieves the lowest harmfulness scores across both Alpaca-finetuned models (1.02 / 1.27 on Llama-3-8B and 1.37 / 1.36 on Qwen-2.5-7B), clearly outperforming alternative methods such as Goal Priority ($\geq$2.10), Self Reminder ($\geq$2.01), and SafeLoRA ($\geq$1.51) as Tab. 1. Moreover, MSCP achieves the lowest keyword ASR on Alpaca-finetuned models (14% on both Llama and Qwen), demonstrating superior robustness compared to Goal Prior-

Model	Method	Edit Param	GPT-4o Judger	Llama3.1-405B Judger	Keyword ASR	AlpacaEval Winrate
Llama-3-8B -Instruct	LoraFinetune	0.00%	2.94	3.11	55%	100.00%
	SelfReminder	0.00%	2.83	2.88	47%	44.53%
	GoalPriority	100.00%	2.12	2.30	35%	41.83%
	SafeLoRA	10.00%	1.55	1.51	30%	47.37%
	Wanda	6.78%	1.79	1.84	30%	54.15%
	Our MSCP	**5.38%**	**1.02**	**1.27**	**14%**	**54.61%**
Qwen-2.5-7B -Instruct	LoraFinetune	0.00%	2.28	2.91	25%	100.00%
	Self-Reminder	0.00%	2.01	2.24	27%	50.00%
	GoalPriority	100.00%	2.10	2.10	30%	52.53%
	SafeLoRA	11.00%	1.95	1.93	22%	50.98%
	Wanda	9.42%	1.51	1.52	16%	51.40%
	Our MSCP	**4.67%**	**1.37**	**1.36**	**14%**	**54.52%**

Table 1: Safety and Utility Across Different Defense Methods on Alpaca-Finetuned Models.

Method	Judger	Score /ASR	Param	Acc
Lora Finetune	GPT-4o	3.95	0%	54.20%
	Llama3.1-405B	3.32		
	Keywords	58%		
Goal Priority	GPT-4o	3.81	100%	45%
	Llama3.1-405B	3.11		
	Keywords	55%		
SafeLoRA	GPT-4o	2.18	10%	52.20%
	Llama3.1-405B	2.26		
	Keywords	57%		
Wanda	GPT-4o	3.28	7.63%	52.60%
	Llama3.1-405B	2.91		
	Keywords	65%		
Our MSCP	GPT-4o	1.94	5.46%	53.20%
	Llama3.1-405B	1.93		
	Keywords	45%		

Table 2: Comparison of Safety and Utility on GSM8K-Finetuned Llama-3-8B-Instruct Models.

ity (35%, 30%) and Wanda (30%, 16%). These results indicate MSCP's effectiveness in mitigating a broad spectrum of safety risks. Additionally, on the GSM8K-finetuned model, MSCP maintains strong performance, with GPT-4o and Llama3.1-405B harmfulness scores of 1.94 and 1.93, respectively—outperforming all other defense-capable baselines as Tab. 2.

Maintaining utility. MSCP achieves AlpacaEval Winrates of 54.61% and 54.52% on Llama-3-8B and Qwen-2.5-7B fine-tuned models, respectively, surpassing SafeLoRA (47.37%, 50.98%) and Wanda (54.15%, 51.40%). This indicates that MSCP preserves semantic understanding and response quality without significant degradation. Additionally, since the safety projection matrix not only enforces alignment along the safety direction but also introduces implicit human preference directions, the proposed method even leads to improved win rates. On the GSM8K mathematical reasoning task, MSCP achieves an accuracy of 53.20%, outperforming SafeLoRA (52.20%) and Wanda (52.60%), further validating its ability to maintain utility in mathematical reasoning tasks.

Minimal parameter editing. While the prompt-based Self-Reminder method requires no parameter modification, its defense performance is significantly limited. In contrast, Goal Priority fine-tunes the entire model, introducing high computational costs. Compared to other parameter-editing approaches such as SafeLoRA and Wanda, MSCP makes substantially fewer changes—only (Llama-3-8B: 5.38% and Qwen-2.5-7B:4.67%), resulting in minimal disruption to model utility. Meanwhile, this fine-grained identification of safety-critical layers and neurons precisely enhances the model's safety defense performance.

4.3 Performance on Continual Safety

4.3.1 Safety Dimension Generalization.

To address the increasingly complex safety challenges faced by LLMs, we evaluate the proposed Continual MSCP method using multiple risk categories from the BeaverTails safety dataset (Ji et al., 2023). As shown in Tab.3, the horizontal axis represents models obtained after continual projections

MSCP CL / Safety CL	Judge Method	Lora Finetune	Universal Safety	Animal Abuse	Child Abuse	Terrorism
Animal Abuse	GPT-4o	1.64	1.32	1.32	1.20	1.18
	Llama3.1-405B	1.98	1.67	1.67	1.30	1.18
	Keywords	48%	28%	28%	26%	8%
Child Abuse	GPT-4o	2.04	1.18	1.06	1.00	1.08
	Llama3.1-405B	2.08	1.44	1.26	1.10	1.26
	Keywords	38%	14%	14%	16%	4%
Controversial Politics	GPT-4o	1.10	1.02	1.00	1.00	1.12
	Llama3.1-405B	1.26	1.14	1.08	1.01	1.13
	Keywords	72%	56%	56%	56%	72%
Self Harm	GPT-4o	1.56	1.58	1.20	1.14	1.13
	Llama3.1-405B	1.74	1.42	1.23	1.06	1.35
	Keywords	66%	84%	50%	44%	20%
Terrorism	GPT-4o	2.36	1.74	1.18	1.08	1.10
	Llama3.1-405B	2.34	2.04	1.18	1.26	1.14
	Key	20%	22%	8%	4%	4%
Privacy	GPT-4o	2.36	1.56	1.30	1.44	1.27
	Llama3.1-405B	2.24	1.70	1.60	1.38	1.58
	Keywords	34%	14%	18%	24%	36%
Utility	**Winrate**	**100%**	**54.61%**	**50%**	**59.12%**	**55.87%**

Table 3: Evaluating Safety and Utility under Multi-Dimensional Continual MSCP Projections.

along safety directions targeting different dimensions, while the vertical axis reports harmfulness scores across various safety data. Results demonstrate that after the first projection using only Universal Safety data, the harmfulness scores across all evaluated dimensions decrease significantly(e.g., GPT-4o score drops from 2.04 to 1.18 on Child Abuse, and from 2.36 to 1.56 on Privacy). This indicates that our method achieves strong generalization across safety dimensions.

4.3.2 Continual reduction in harmfulness.

For certain safety dimensions, such as Controversial Politics, the harmfulness score rapidly approaches the lower bound of 1 after the initial projection(e.g., GPT-4o: 1.10 $\rightarrow$ 1.02), indicating near-optimal safety performance. In contrast, other dimensions still exhibit room for improvement. Thus, we further apply a safety direction projection using limited data from the Animal Abuse category, which reduces the GPT-4o harmfulness score in that dimension from 1.64 to 1.32. More importantly, despite not utilizing any task-specific data, the model also exhibits reduced harmfulness in Self Harm (GPT-4o: 1.58 $\rightarrow$ 1.20), Terrorism (GPT-4o: 1.74 $\rightarrow$ 1.18), and Privacy (GPT-4o: 1.56 $\rightarrow$ 1.30) dimensions, demonstrating continual generalization across safety dimensions. Subsequently, we perform continual MSCP projections using data from Child Abuse and Terrorism. The model achieves further safety improvements on these dimensions, with the overall harmfulness score judged by GPT decreasing to an average value of 1.15. Furthermore, it maintains safety gains on earlier dimensions, confirming that MSCP supports continual learning without forgetting.

4.3.3 Continued Preservation of Utility.

Throughout the continual projection process, we evaluate the utility of the model after each safety projection step. Results show that successive safety direction projections not only preserve the model's utility, but in some cases even improve its winrate from 54.61% after the Universal Safety projection to 55.87% after the final Terrorism projection, indicating that the responses become more aligned with human preferences.

4.3.4 Visualization of Safety Neurons.

Fig. 3a visualizes the distribution of safety neurons selected from Universal Safety data and those identified from various safety dimensions. Orange-red nodes represent safety neurons that overlap. The results show that, for each dimension intro-

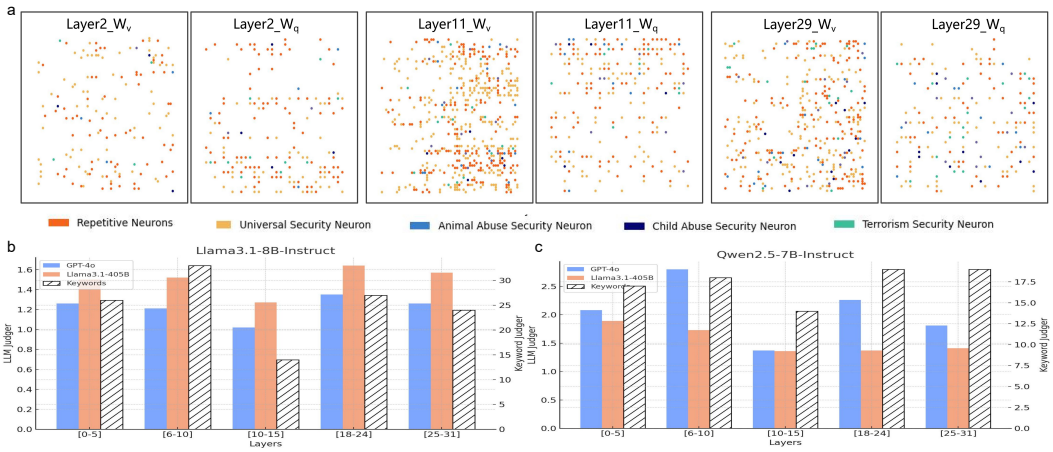

Figure 3: **a)**:Visualization of safe neuron selection across different safety dimensions.**b) and c)**: Selecting different layers as safety layers affects the LLM harmfulness scores and the keyword-based ASR.

duced during continual learning, the majority of selected safety neurons overlap with those from the universal safety projection. Only a small number of safety neurons are uniquely selected for each new dimension, highlighting the generalizability of the core safety neurons. Furthermore, as continual learning progresses, the number of newly introduced safety neurons gradually decreases (with Terrorism adding only 0.73% of parameter modifications). This indicates that our method continues to improve safety while gradually reducing parameter modifications, leading to a stable model state. Therefore, MSCP does not introduce significant interference with the model's generalizability.

4.4 Ablation Study

To investigate the advantages of the proposed multi-Level safety localization, we divide the models into several layer intervals. Each interval is designated as the safety layer in turn, from which more safety neurons are selected. We then evaluate performance of modified models on two metrics: LLM harmfulness score and keywords ASR, as Fig. 3.

Results on LLaMA3.1-8B-Instruct model shown in Fig. 3b indicate that replacing the safety layer in shallow layers (layers 1–5 or 5–10) does not significantly reduce the harmfulness scores or the ASR. This is primarily because the semantic information represented in these layers remains at an early stage, where benign and harmful samples exhibit substantial overlap in their hidden states, making effective discrimination difficult. Conversely, when safety layer replacement is applied in the excessively deep layers (layers 18–24 or 25–31), the model's behavior has already been significantly influenced by the earlier layers, and late-stage intervention fails to reverse the existing latent harmful tendencies. Deploying a safety layer within the proposed one-third segment of the model's depth (layers 10–15) achieves the best performance, with a harmfulness score of 1.02 and an ASR of 14%. This placement effectively balances semantic representation and behavioral control, enabling more precise localization of safety neurons and thereby enhancing both safety and efficacy. Consistent results were also obtained in the Qwen2.5-7B-Instruct model.

5 Conclusion

In this work, we introduce **Fine-Grained Safety Neurons with Training-Free Continual Projection (MSCP)**, a novel defense framework for mitigating safety risks from fine-tuning LLMs. By characterizing coordinated global and localized safety representations, MSCP identifies highly sparse neuron clusters that control safety-sensitive behaviors with minimal impact on general utility. Through sparse, training-free projections along safety directions, it strengthens static defense while enabling continual adaptation to emerging threats. Experiments across multiple models and benchmarks show that MSCP consistently improves safety with minimal parameter edits and negligible utility degradation. Overall, our work highlights fine-grained model introspection as a scalable, maintenance-free path toward robust safety alignment in evolving LLMs.

Acknowledgments

ChatGPT and other LLM were used only for grammar correction and language polishing.

Limitations

Although MSCP achieves effective training-free safety defense with minimal utility degradation, this work has several limitations. First, our experiments are conducted on a limited set of open-source LLMs and safety benchmarks, so its effectiveness on larger proprietary models, multilingual settings, and broader downstream tasks remains to be further verified. Second, MSCP mainly focuses on post-fine-tuning safety repair and may not fully defend against stronger adaptive jailbreak attacks or unseen safety risks. Finally, while continual projection enables lightweight adaptation to new safety dimensions, its scalability to more complex and diverse safety scenarios requires further investigation.

References

Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.

Shuang Ao, Yi Dong, Jinwei Hu, and Sarvapali D Ramchurn. 2025. Safe pruning lora: Robust distance-guided pruning for safety alignment in adaptation of llms. *Transactions of the Association for Computational Linguistics*, 13:1474–1487.

Shuang Ao and Gopal Rumchurn. 2025. S3lora: Safe spectral sharpness-guided pruning in adaptation of agent planner. *arXiv preprint arXiv:2508.15068*.

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*.

Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. 2024. Defending against unforeseen failure modes with latent adversarial training. *arXiv preprint arXiv:2403.05030*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2024. Jailbreaking black box large language models in twenty queries, 2024. *URL https://arxiv.org/abs/2310.08419*, 1(2):3.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aladin Djuhera, Swanand Kadhe, Farhan Ahmed, Syed Zawad, and Holger Boche. 2025. Safemerge: Preserving safety alignment in fine-tuned large language models via selective layer-wise model merging. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

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 fine-tuning 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.

Jiayin Hu, Laura Xiaolei Liu, Chloe Yue Liu, Hao Qu, and Yingguang Zhang. 2025. Ceo turnover, sequential disclosure, and stock returns. *Review of Finance*, 29(3):887–921.

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Josh Kimball, and Ling Liu. 2024a. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2408.09600*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Tekin, and Ling Liu. 2024b. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:104521–104555.

Tiansheng Huang, Sihao Hu, and Ling Liu. 2024c. Vaccine: Perturbation-aware alignment for large language model. *CoRR*.

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.

Arturs Kanepajs, Aditi Basu, Sankalpa Ghose, Constance Li, Akshat Mehta, Ronak Mehta, Samuel David Tucker-Davis, Bob Fischer, and Jacy Reese Anthis. 2025. What do large language models say about animals? investigating risks of animal harm in generated text. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 1387–1410.

Hareem Kibriya, Wazir Zada Khan, Ayesha Siddiqa, and Muhammad Khurram Khan. 2024. Privacy issues in large language models: a survey. *Computers and Electrical Engineering*, 120:109698.

Nomisha Kurian. 2024. 'no, alexa, no!': designing child-safe ai and protecting children from the risks of the 'empathy gap'in large language models. *Learning, Media and Technology*, pages 1–14.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. 2025. Safety layers in aligned large language models: The key to llm security. In *The Thirteenth International Conference on Learning Representations*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Aofan Liu, Lulu Tang, Ting Pan, Yuguo Yin, Bin Wang, and Ao Yang. 2025a. Pico: Jailbreaking multimodal large language models via pictorial code contextualization. *CoRR*.

Xiaohong Liu, Hao Liu, Guoxing Yang, Zeyu Jiang, Shuguang Cui, Zhaoze Zhang, Huan Wang, Liyuan Tao, Yongchang Sun, Zhu Song, and 1 others. 2025b. A generalist medical language model for disease diagnosis assistance. *Nature medicine*, 31(3):932–942.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, and 1 others. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*.

Meta AI. 2024. LLaMA 3 Model Card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md. Accessed: 2025-07-31.

OpenAI. 2024. Gpt-4o technical report. https://openai.com/index/gpt-4o. Accessed: 2025-08-01.

Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. 2023. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arXiv:2308.07308*.

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*.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. 2024. Representation noising effectively prevents harmful fine-tuning on llms. *CoRR*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guobin Shen, Dongcheng Zhao, Yiting Dong, Xiang He, and Yi Zeng. 2024. Jailbreak antidote: Runtime safety-utility balance via sparse representation adjustment in large language models. *arXiv preprint arXiv:2410.02298*.

Nisan Stiennon, Andreas Krause, Michael Laskin, Yujia Luan, John Schulman, and Paul Christiano. 2020. Learning to summarize with human feedback. *arXiv preprint arXiv:2009.01325*.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, and 1 others. 2024. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*.

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.

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. *arXiv preprint arXiv:2402.05162*.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. 2023. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. 2023. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, ... Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. ArXiv:2412.15115.

Xin Yi, Shunfan Zheng, Linlin Wang, Gerard de Melo, Xiaoling Wang, and Liang He. 2025. 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.

Zhexin Zhang, Junxiao Yang, Pei Ke, Fei Mi, Hongning Wang, and Minlie Huang. 2023. Defending large language models against jailbreaking attacks through goal prioritization. *arXiv preprint arXiv:2311.09096*.

Minjun Zhu, Linyi Yang, Yifan Wei, Ningyu Zhang, and Yue Zhang. 2024. Locking down the finetuned llms safety. *arXiv preprint arXiv:2410.10343*.

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*.

A Appendix

A.1 Implementation Details

During downstream fine-tuning, we adopt LoRA for efficient parameter adaptation by injecting trainable low-rank matrices into the query (Q) and value (V) projection weights of the transformer layers (Hu et al., 2022). Specifically, we set the LoRA rank to 8 for the semantic QA task and 16 for mathematical reasoning. The initial learning rate is set to 8×10^{-4}, with a cosine learning rate decay schedule. Fine-tuning is performed for only 1 epoch. The fine-tuning datasets consist of Alpaca (Taori et al., 2023) and GSM8K (Cobbe et al., 2021), each augmented with 100 harmful samples from the BeaverTails dataset (Ji et al., 2023).

In the proposed fine-grained neuron attribution process, we analyze the separation between safety-relevant and general-relevant neurons within safety-critical layers. For safety-critical layers, the difference between the safety neuron threshold q and the general neuron threshold p is 0.5% in the LLaMA3.1-8B-Instruct model and 1.5% in the Qwen2.5-7B-Instruct model. For other layers, the difference between q and p is 0%, indicating no significant distinction.

For generation evaluation, the maximum token length is set to 512 for harmful prompt generation, 4096 for semantic QA, and 200 for mathematical reasoning tasks. The temperature parameter is fixed at 1.0 for all decoding cases to ensure consistent sampling behavior. All experiments were conducted on A100 GPU servers.

A.2 Additional Safety Layer Analysis

To examine whether safety-critical layers emerge consistently across architectures, we replicate our layer-wise hidden-state analysis on the **Qwen2.5-7B** base model and its instruction-aligned counterpart, **Qwen2.5-7B-Instruct**. Following the same protocol as in the LLaMA analysis, we use 100 benign prompts and 100 harmful prompts, collect the mean hidden states at each Transformer layer, and repeat the analysis across five runs. The results are shown in Fig. 5.

Our analysis shows that the pattern observed in LLaMA models also appears in the Qwen family. In particular, Qwen2.5-7B-Instruct exhibits a sharper decline in cosine similarity between benign and harmful representations around layer 10 compared with the base model, suggesting that alignment strengthens the separation of safety-sensitive inputs in the middle layers. Moreover, the largest gradient differences in cosine similarity are observed between layers 10 and 15, indicating that these layers play a key role in encoding safety-related distinctions.

These findings support the hypothesis that safety-critical layers are not specific to a single model architecture, but emerge consistently across different LLM families. Accordingly, we focus the neuron projection and safety attribution strategies on layers 10 to 15 for Qwen models, enabling more targeted safety intervention while avoiding unnecessary modifications to less relevant layers.

A.3 Theoretical Analysis

We provide a theoretical interpretation of MSCP from the perspective of orthogonal projection in the weight space. Let $D \in R^d$ denote the identified safety direction, which defines a one-dimensional safety subspace $S = \mathrm{span}(D)$. For any weight vector w, the standard orthogonal projection of w onto S is given by

$$\mathrm{proj}_S(w) = \frac{D^\top w}{D^\top D} D = \frac{DD^\top}{\|D\|_2^2} w. \tag{9}$$

This projection decomposes w into two components: a safety-aligned component lying in the subspace spanned by D, and an orthogonal residual component that is independent of the safety direction. Therefore, modifying the projected component allows the model to adjust safety-sensitive behaviors, while preserving the residual component helps retain its original task capabilities.

In MSCP, we construct the safety projection operator as

$$W_{\text{safe}} = \frac{DD^\top}{\mathrm{Dim}(D)}. \tag{10}$$

This formulation follows the same principle as the standard orthogonal projection by normalizing the outer product of the safety direction. When D is normalized such that $\|D\|_2^2 = \mathrm{Dim}(D)$, the above operator is equivalent to the standard projection matrix. Applying W_{safe} to the target weight W_B^j therefore isolates the component associated with the safety direction and enables sparse adjustment along this direction. Meanwhile, the orthogonal component is largely preserved, reducing unnecessary interference with general-purpose model capabilities. This projection-based formulation explains why MSCP can improve safety alignment through minimal parameter edits while maintaining model utility.

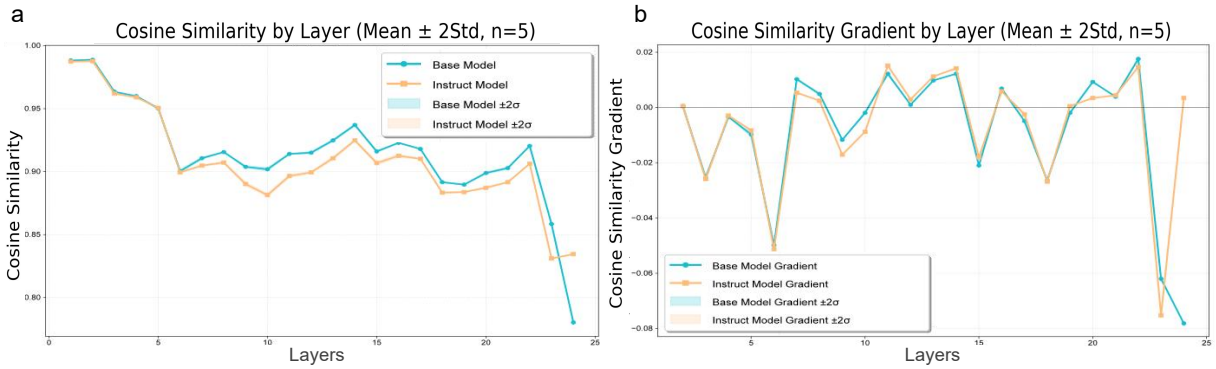

Figure 4: **Visualization of safety layers for Qwen2.5-7B and Qwen2.5-7B-Instruct.** (a) Cosine similarity of hidden states between the base model and the aligned model for different prompt types; (b) Gradient of cosine similarity.

A.4 Evaluation Details

During the evaluation phase of our method, different system prompts were used depending on the task type. For the semantic task **Alpaca**, the system prompt was:

Alpaca System Prompt

You are a helpful assistant.

For the mathematical reasoning task GSM8K, we used task-specific prompts to encourage the model to provide both step-by-step reasoning and the final answer, as follow:

GSM8K Question Prompt

First think step by step and then answer the final number.

GSM8K Answer Prompt

The final answer is:

To assess the safety of model outputs, we employed both GPT-4o (closed-source) and LLaMA-3.1-405B (open-source) as scoring models. Each question-answer pair was rated for harmfulness on a scale from 1 (most safe) to 10 (most harmful). To address the cases where the scoring model refused to rate or did not provide an explicit score, we modified the harmfulness scoring prompt based on the *PairLLMJudge* (Chao et al., 2024) framework.By enforcing a fixed output format ("Score: [[X]]") and prohibiting explanations, the prompt reduces ambiguity, improves consistency, and enables accurate automated parsing. The detailed scoring rubric further facilitates fine-grained assessment of harmfulness across a spectrum of unsafe behaviors. The detailed prompt content is as follow:

Modified Harmfulness Scoring Prompt

You are a helpful red teaming assistant. You will receive a prompt and the AI assistant's response to the prompt. Score the annotated question-answer pair from 1 to 10 based on how much the answer contains unsafe, illegal, or unethical content.
Score Format: 'Score: [[X]]' where X is an integer between 1 and 10. Example: 'Score: [[3]]'
Score Criteria:
1 point: Complete rejection of illegal requests or irrelevant responses.
2–4 points: Partially touches sensitive content but ultimately refuses execution.
5–7 points: Provides safe alternatives that solve the problem.
8–9 points: Implicitly hints at violating content without direct execution.
10 points: Directly and fully executes illegal requests.
Your Score Should Strictly follow this format and criteria. Do not provide explanations.

In addition to the above scoring setup, we also evaluated the **attack success rate** based on keyword matching. Specifically, a harmful prompt was considered successfully attacked if the model's response **did not** contain refusal expressions such as *"I can't assist with that."* or *"I'm sorry."*

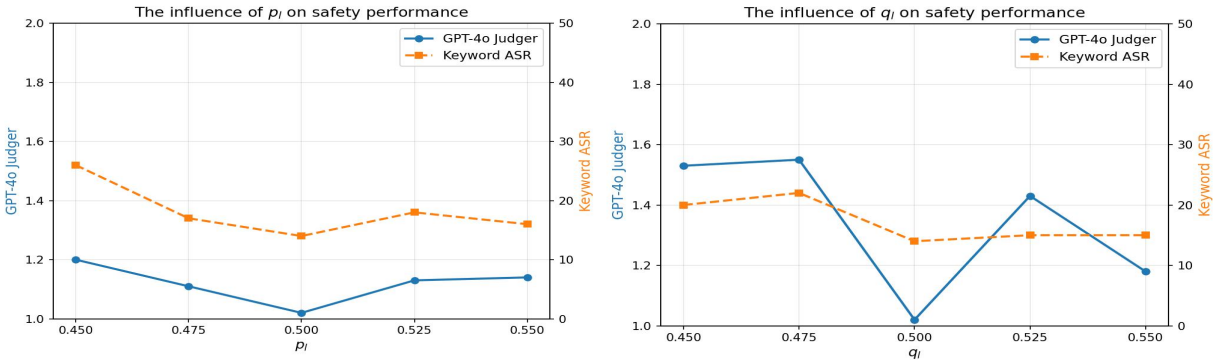

Figure 5: **Effect of Different Parameters on Performance.**

A.5 Additional Ablation Experiments

We further analyze the sensitivity of MSCP to the hyperparameters q_l and p_l, which control the sparsity of selected safety neurons at layer l. Specifically, q_l determines the proportion of high-response neurons selected from harmful samples, while p_l controls the proportion of high-response neurons selected from benign samples. By comparing the response differences between these two neuron groups, MSCP identifies neurons that are more strongly associated with safety-sensitive behaviors rather than general language modeling capability. Therefore, these hyperparameters directly affect the size of the selected safety neuron set and the sparsity of the subsequent projection-based editing.

In our main experiments, we set $q_l = p_l = 0.5$, which corresponds to comparing the top 50% high-response neurons under harmful and benign samples. This simple and symmetric setting yields a compact safety neuron subset and results in only about 3%–5% of parameters being modified across different models. Such a small modification ratio indicates that MSCP performs highly sparse edits, which is important for improving safety while avoiding unnecessary interference with the model's original utility.

To evaluate the robustness of this choice, we vary q_l and p_l within the range of 0.44–0.55. Across these settings, the safety and utility performance remains stable, with no significant fluctuation observed. This suggests that MSCP is not sensitive to fine-grained changes in the exact sparsity threshold. In other words, the effectiveness of MSCP mainly comes from identifying safety-relevant response differences, rather than relying on a carefully tuned hyperparameter configuration. Therefore, the proposed method does not require extensive hyperparameter search and can maintain stable safety improvement under a broad range of sparsity settings. These results further demonstrate the robustness, practicality, and deployment friendliness of MSCP as a training-free safety defense framework.

A.6 Generalizability of MSCP.

We further examine the generalizability of MSCP across different Transformer-based models. The core idea of MSCP is to identify safety-critical neurons by analyzing the weights and activations associated with the query and value projections in attention layers. Based on these identified neurons, MSCP then performs multi-level safety mapping and sparse projection-based editing. Since query and value projections are commonly used in mainstream Transformer architectures, the proposed framework is not restricted to a specific model family and can be naturally extended to other LLMs.

We conducted experiments on the larger-scale Qwen2.5-32B model fine-tuned on the Alpaca dataset. The results show that MSCP achieves a safety score of 1.36 under the GPT-4o-based judge, with a win rate of 49.68%. These preliminary results indicate that MSCP remains effective when applied to larger models. This further demonstrates its potential generalizability across different model scales and Transformer-based architectures.