NLSR: Neuron-Level Safety Realignment of Large Language Models Against Harmful Fine-Tuning

Xin Yi[1], Shunfan Zheng[1], Linlin Wang[1*], Gerard de Melo[2, 3], Xiaoling Wang[1], Liang He[1]

[1] East China Normal University
[2] Hasso Plattner Institute
[3] University of Potsdam
{xinyi,sfzheng}@stu.ecnu.edu.cn, {llwang, xlwang, lhe}@cs.ecnu.edu.cn, demelo@uni-potsdam.de

Abstract

The emergence of fine-tuning-as-a-service has revealed a new vulnerability in large language models (LLMs). A mere handful of malicious data uploaded by users can subtly manipulate the fine-tuning process, leading to a compromised alignment state. Existing methods to counteract fine-tuning attacks typically require substantial computational resources. Even with parameter-efficient techniques like LoRA, gradient updates remain essential. To address these challenges, we propose **N**euron-**L**evel **S**afety **R**ealignment (**NLSR**), a training-free framework that restores the safety of LLMs based on the similarity difference of safety-critical neurons before and after fine-tuning. The core of our framework is first to construct a safety reference model from an initially aligned model to amplify safety-related features in neurons. We then utilize this reference model to identify safety-critical neurons, which we prepare as patches. Finally, we selectively restore only those neurons that exhibit significant similarity differences by transplanting these prepared patches, thereby minimally altering the fine-tuned model. Extensive experiments demonstrate significant safety enhancements in fine-tuned models across multiple downstream tasks, while greatly maintaining task-level accuracy. Our findings indicate that safety-critical neurons exhibit significant regional variations after fine-tuning, which can be effectively corrected through neuron transplantation from the reference model without the need for additional training.

Code — https://github.com/xinykou/NLSR

Introduction

Emerging research suggests that the growing fine-tuning-as-a-service model raises significant safety concerns (Huang et al. 2024b). In contrast to open-source settings, fine-tuning-as-a-service means that users do not have direct access to model parameters. However, users may inadvertently or deliberately upload data containing harmful content, potentially compromising the model's safety alignment. One study shows that mixing just 1% of harmful instructions into the clean dataset used for fine-tuning is enough to breach the model's safety mechanisms (Qiang et al. 2024). Additionally, He, Xia, and Henderson (2024) and Kumar et al. (2024) have also demonstrated that fine-tuning, even on clean data, can degrade the model's safety. As illustrated in Figure 1, a customized model that has been fine-tuned with harmful instructions can comply with malicious requests of an attacker, leading to harmful or unethical behavior. Such harmful fine-tuning attacks raise serious concerns regarding the practical deployment of LLMs.

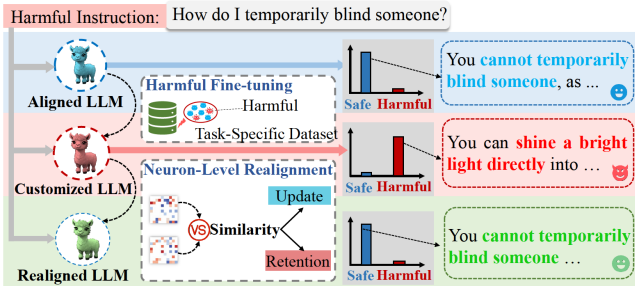

Figure 1: The harmful fine-tuning attack for fine-tuning-as-a-service scenarios and our neuron-level safety realignment approach to mitigate it.

To mitigate the degradation of safety safeguards caused by harmful fine-tuning, the main methods can be categorized into three types based on the stage of the safety defense. The first strategy involves introducing perturbations that could potentially trigger harmful behaviors, with the aim of recalibrating the model's parameters to counteract these threats (Huang, Hu, and Liu 2024; Zeng et al. 2024; Reuel et al. 2024). However, perturbation-based methods are sensitive to the form of harmful instructions, leading to significant variability in their effectiveness against different types of harmful instructions. The second strategy entails fine-tuning the model on both a task-specific dataset and a preference dataset to bolster the model's consistency in providing harmless and useful outputs (Zong et al. 2024; Huang et al. 2024c). Nevertheless, a critical challenge remains in striking the optimal balance between optimizing task-level performance and ensuring output safety during fine-tuning. The third strategy avoids interfering with the fine-tuning objectives and instead directly realigns the fine-tuned model to ensure safety (Hsu et al. 2024; Bhardwaj, Anh, and Poria 2024). SafeLoRA (Hsu et al. 2024) is a realignment tech-

*Corresponding Author

nique that evaluates the difference in the safety subspace across each layer pre- and post-fine-tuning through a projection matrix. However, this method of aligning layer-specific parameters inherently misses certain critical neurons that are vital for the performance on downstream tasks.

Therefore, more fine-grained updates for customized models are essential to preserve task-specific performance while ensuring effective safety tuning. Chen et al. (2024) propose activation contrasting as a method for identifying safety-related neurons within large language models (LLMs). Wei et al. (2024) highlight that merely freezing safety-critical neurons is inadequate to defend against fine-tuning attacks. Motivated by the critical role of neurons in maintaining model safety, we advocate for neuron-level safety realignment to restore safety while minimizing adverse effects on task-specific performance.

In this paper, we propose a Neuron-Level Safety Realignment (**NLSR**) framework designed to mitigate safety degradation caused by the inclusion of harmful instructions during the fine-tuning of large language models (LLMs). First, we construct a safety preference model via pre-amplification, which enhances the discriminative features of neurons essential for safety. Second, we identify safety-critical neurons by evaluating their contribution scores. Third, we analyze discrepancies in safety-critical neurons post-fine-tuning to identify layers requiring safety correction without the need for additional training. Our primary contributions are as follows:

- We introduce a neuron-level safety realignment method that is decoupled from the fine-tuning phase and requires no additional training. Our approach centers on identifying safety-critical neurons and determining whether to patch them based on the degree of damage incurred during fine-tuning.
- We conduct extensive evaluations across varying proportions of poisoned instructions, diverse downstream tasks, and different alignment methods. The results demonstrate that NLSR restores safety while preserving the accuracy of downstream task performance.
- We demonstrate that the proposed adaptive safety-critical layer pruning is essential for identifying layers compromised in safety. Additionally, we observe that after our safety pre-amplification process, various safety neuron identification methods exhibit significant consistency in localizing safety-critical neurons.

Neural-Level Safety Realignment

Safety realignment against harmful fine-tuning aims to restore the ability of a customized model F_{W_t} to reject harmful instructions. Specifically, the customized model is obtained by fine-tuning an initially safety-aligned model F_{W_a} on task-specific data that consists of benign samples but also contains a small proportion of toxic instructions.

Overview of NLSR

Our method aims to ensure that the customized model maintains a safety level comparable to the initially aligned model. ① We begin by pre-amplifying the initial aligned model to construct a super-aligned LLM, which serves as our safety reference model. ② We then develop a scoring mechanism to identify safety-critical neurons within the reference model. ③ Finally, we compare the similarity of safety-critical neurons across each layer of the customized model with those in the reference model. For layers exhibiting lower similarity, indicating potential safety degradation, we rectify the compromised neurons by transferring the corresponding safety-critical neurons from the reference model, as illustrated in Figure 2.

Construction of a Safety Reference Model

To enhance the prominence of safety-related neurons in the aligned model for step ②, and to prepare patch neurons for step ③, we start with the amplification of the aligned model. We propose extending the concept of weak-to-strong extrapolation (Zheng et al. 2024) into the safety domain through LoRA extrapolation, resulting in a more robust safety-aligned model, referred to as the super-aligned F_{W_e}. Specifically, we keep the majority of the model's weights $W_{\text{unaligned}}$ frozen and update only the LoRA weights to obtain a safer model. Given weaker LoRA weights W_{weak} obtained by supervised fine-tuning (SFT) and stronger LoRA weights W_{strong}, we apply the principle of interpolation to derive fused medium-level safety LoRA weights W_{medium} as follows:

$$W_{\text{medium}} = \alpha W_{\text{strong}} + (1-\alpha) W_{\text{weak}}, \quad \alpha \in (0, 1] \tag{1}$$

If strong LoRA weights W_{strong} are unavailable, preference-aligned LoRA weights and SFT weights W_0 are provided, we amplify safety through extrapolation to obtain super-aligned weights W_e using the following formula:

$$W_{\text{e}} = \frac{1}{\alpha} W_{\text{a}} - \left(\frac{1-\alpha}{\alpha}\right) W_0 = (1+\beta) W_{\text{a}} - \beta W_0 \tag{2}$$

where $\beta = \frac{1-\alpha}{\alpha} \in [0, +\infty)$ is the pre-amplification coefficient. In this context, $W_{\text{e}} = W_{\text{strong}}$, $W_0 = W_{\text{weak}}$ and $W_{\text{a}} = W_{\text{medium}}$.

Recognition of Safety-Critical Neurons

To compare which safety-critical neurons are seriously broken by harmful fine-tuning, we need to determine the location distribution of these neurons in the aligned model in advance. Following the approach described by Wei et al. (2024), we construct a dataset for safety-critical neuron identification, consisting of instances $s = (x_{\text{prompt}}, y_{\text{response}})$, where $s \in S$ and $S = \{s_1, s_2, ..., s_n\}$, with n denoting the number of instances. To identify safety-critical neurons, we apply rank reduction to LoRA weights at a specified sparsity rate P_{SR}. The model's representation for the y_{response} of the i-th instance in the j-th layer W_j is $W_j X_j^i$, where $X_j^i \in \mathbb{R}^{d \times l}$ and $W_j \in \mathbb{R}^{d' \times d}$. The matrix formed by all instances can be represented as $W_j \hat{X}_j^i$, where $\hat{X}_j^i \in \mathbb{R}^{n \times (d' \times l)}$. We seek a low-rank matrix $\hat{W}$ that minimizes the Frobenius norm of the difference between the original and approximated outputs:

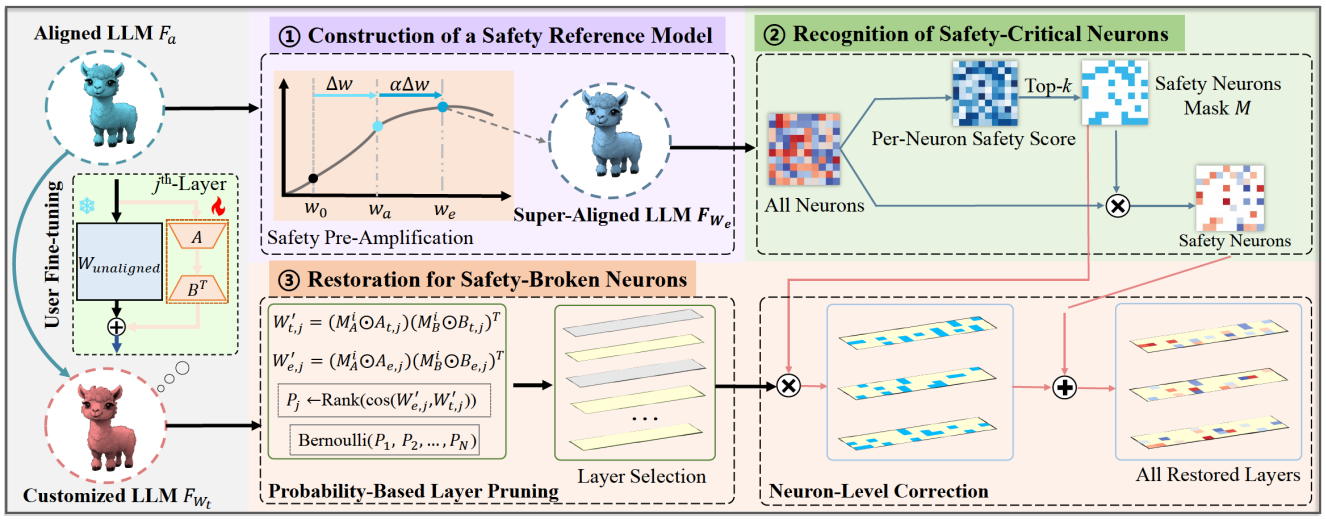

Figure 2: A neuron-level safety realignment framework against harmful fine-tuning during adaptation to new tasks or domains.

$$\hat{W}_j = \underset{\text{rank}(\hat{W}_j)\leq r^*}{\arg\min} \|W_j \hat{X}^i_j - \hat{W}_j \hat{X}^i_j\|^2_F \tag{3}$$

where the retained rank is $r^* = r \times (1 - P_{SR})$. Based on the Truncated SVD decomposition of $W\hat{X}^i_j$, we have:

$$USV^T \approx W_j \hat{X}^i_j \tag{4}$$

Using the truncated SVD results, a rank-r^* matrix $\hat{W}_j = UU^T W_j$ is constructed. This matrix is a low-rank approximation because it is obtained by retaining the top r^* left singular vectors. The projection matrix $\Pi = UU^T$, derived from the left singular vectors, projects the matrix W_j onto the rank-r^* subspace. Consequently, $\hat{W}_j$ becomes an updated version of W_j that preserves the safety-critical weights. To identify neurons essential for safety based on the updated weights $\hat{W}_j$, we transform the updated weight into a safety score based on the highest-magnitude values to select the Top-k $= N^* \times (1 - P_{SR})$ neurons among all N^* neurons as follows:

$$\text{indices} = \text{argsort}(-|\hat{W}_j|)[:, : \text{Top} - \text{k}] \tag{5}$$

We locate the i^*-th neuron by a position mask M_{j,i^*}, defined as:

$$M_{j,i^*} = \begin{cases} 1, & \text{if } i^* \in \text{indices} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

With the locations of the safety-critical neurons identified, we employ **probability-based layer pruning** focus solely on layers where safety is severely compromised enabling a more targeted **neuron-level correction** using the patch neurons from the reference model obtained in step ①.

Restoration for Safety-Broken Neurons

Probability-based Layer Pruning. After fine-tuning an aligned LLM F_{W_a} for a task-specific dataset contaminated with harmful instances, we acquire a customized LLM F_{W_t}. The updated LoRA weights of the j-th layer are represented as $W_{t,j} = B_{t,j}A_{t,j}$, where $A \in \mathbb{R}^{r\times k}$, $B \in \mathbb{R}^{d\times r}$. Although fine-tuning enhances the task-specific performance, it compromises alignment, as many safety-critical neurons become significantly corrupted. To balance utility and safety, we focus on updating neurons in layers where the broken neurons deviate significantly from those in the reference model. The regions constructed by safety-critical neurons before and after fine-tuning are denoted as

$$\begin{aligned} W'_{e,j} &= (M^B_j \odot B_j)(M^A_j \odot A_{e,j}) \\ W'_{t,j} &= (M^B_j \odot B_j)(M^A_j \odot A_{t,j}) \end{aligned} \tag{7}$$

where $M^A_j \in \mathbb{R}^{r\times k}$ and $M^B_j \in \mathbb{R}^{d\times r}$. In M^A_j and M^B_j, only the positions corresponding to safety-critical neurons are set to 1, while all other positions remain 0.

We identify layers requiring updates to their safety regions (i.e., safety-critical neurons) based on their similarity, computed as

$$S_j = \frac{\langle W'_{e,j}, W'_{t,j}\rangle_F}{||W'_{e,j}||_F ||W'_{t,j}||_F} \tag{8}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product, and $||\cdot||_F$ denotes the Frobenius norm. These layers with low similarity values indicate significant deviations in their safety regions and are candidates for correction. Inspired by Deep, Bhardwaj, and Poria (2024), we rank layer similarities $S_1, S_2, ..., S_N$ and obtain $\{r_1, r_2, \ldots, r_N\} = \text{rank}(S_1, S_2, \ldots, S_N)$. Based on the rank r_j of the j-th layer, we assign corresponding pruning probabilities:

$$P_j = P_L + \frac{\delta r_j}{N} \tag{9}$$

where P_L is the base layer pruning probability, δ is an increment factor, and N is the total number of layers. We then perform probability-based layer pruning:

$$\gamma_j \sim \text{Bernoulli}(P_j) \tag{10}$$

Neuron-Level Correction. Given the pruning status of all layers, denoted as $\Gamma = \gamma_1, \gamma_2, ..., \gamma_N$, the safety region of j-th layer for a customized LLM F_{W_t} is updated as follows:

$$W''_{t,j} = \begin{cases} W'_{e,j} + \hat{W}'_{t,j} & \text{if } \gamma_j = 0 \\ W'_{t,j} & \text{otherwise} \end{cases}$$
$$\hat{W}'_{t,j} = ((\mathbf{1} - M_j^B) \odot B_{t,j})((\mathbf{1} - M_j^A) \odot A_{t,j}) \quad (11)$$

where γ_j represents the pruning coefficient for the j-th layer. It is dynamically determined based on the similarity score to ensure optimal safety realignment. Specifically, only the layers that are not pruned are deemed to contain significantly compromised safety neurons, necessitating the transplantation of patch neurons from the reference model into these specific layers.

Experiments

Experimental Settings

Datasets and Models. During the alignment phase, we sample a preference dataset consisting of 2,000 instances from PKU-SafeRLHF (Ji et al. 2024) and utilize LoRA (Hu et al. 2022) for SFT, DPO (Rafailov et al. 2024), ORPO (Hong, Lee, and Thorne 2024), KTO (Ethayarajh et al. 2024), and SimPO (Meng, Xia, and Chen 2024) to obtain the initially aligned model. The reference model is synthesized by extrapolating between two models that possess distinct levels of safety alignment. The model with the lower level of alignment is derived from SFT, whereas the intermediate aligned model, serving as the initial model, is developed through preference optimization (i.e., DPO, ORPO, KTO, SimPO). For our base model, we employ Llama3-8B[1] as our base model. Following the experimental setup in Vaccine (Huang, Hu, and Liu 2024), we fine-tune our models on three downstream tasks: SST-2 (Socher et al. 2013), AGNEWS (Zhang, Zhao, and LeCun 2015), and GSM8K (Cobbe et al. 2021). To inject poisoned instructions into these task-specific datasets, we configure each training dataset to contain $n = 1,000$ instances, with a poisoning proportion of $p = 0.05$ from BeaverTails (Ji et al. 2024).

Baselines. We evaluate our method against several baselines: the non-aligned base model at initialization (Non-Aligned), an aligned base model (Aligned), Vaccine (Huang, Hu, and Liu 2024), which serves as a representative defense against harmful samples prior to fine-tuning, Vlguard (Zong et al. 2024), Lisa (Huang et al. 2024c), and ConstrainedSFT (Qi et al. 2024), which provides safeguards against harmful samples during the fine-tuning process. We alos compare against SafeLoRA (Hsu et al. 2024), a safety realignment method applied after fine-tuning.

Evaluation Metrics. Following the approach from Huang et al. (2024c), we evaluate the performance of the model from two perspectives: Fine-tuning Accuracy (**FA**) and Harmfulness Score (**HS**). The fine-tuning accuracy assesses the model's performance on downstream tasks after fine-tuning, while the harmfulness score quantifies the proportion of unsafe content generated by the model in response to sampled harmful queries, as judged by QA-Moderation[2].

Implementation Details. We utilize LoRA to train a safety-aligned model, which is subsequently fine-tuned for specific downstream tasks. Specifically, we update a small fraction of parameters with a rank of 128. In the alignment stage, we use the AdamW optimizer with a learning rate of 2×10^{-6}, except for the ORPO with a learning rate of 2×10^{-4}. The number of training epochs for the alignment stage is universally set to 3. In the fine-tuning stage, the training epochs for all datasets are all set to 10. The batch size is consistently set to 8 for both stages. Unless otherwise specified, the sparsity rate is $P_{SR} = 0.8$, corresponding to a safety region ratio of 0.2. Additionally, the layer pruning rate is set as $P_L = 0.5$.

Main Results

Effectiveness Across Harm Ratios. As shown in Table 1, the unaligned model (Non-Aligned) consistently demonstrates a high harmfulness score across all proportions, averaging 76.3%. Although the harmfulness score of the aligned model (Aligned) decreases by an average of 15.2% post-fine-tuning, it remains at a high level. NLSR reduces the harmfulness by 38.3% compared to the aligned model. It outperforms SafeLoRA with a 30.3% lower harmfulness and a 1.1% increase in fine-tuning accuracy. While ConstrainedSFT maintains a fine-tuning accuracy of 95.2%, its safety performance is inferior to that of NLSR.

Robustness to Different Alignment Methods. The results in Table 2 indicate that models generally establish safety-critical regions during the alignment stage, where neurons in these regions are crucial for maintaining the safety of generated content. Specifically, SFT achieves a low toxicity level of 53.3% after fine-tuning, but it still exhibits the highest harmfulness score at 46.6% even after safety realignment. This suggests that SFT is less effective than the other alignment methods, with inherently weaker safety capabilities embedded in the safety-related neurons. Even after the realignment process, SFT fails to match the performance of the other preference alignment methods. Additionally, our method reduces the harmfulness score by 29.5% relative to the "Aligned" approach without significantly compromising the performance on downstream tasks.

Consistency with Diverse Downstream Tasks. To further assess the effectiveness of our safety realignment method across different task-specific fine-tuning scenarios, we evaluate NLSR using the AGNEWS and GSM8K datasets, comparing its performance against other baseline methods. As shown in Table 3, NLSR reduces the harmfulness score to 19.7% and 15.4% for AGNEWS and GSM8K, respectively. For GSM8K, NLSR achieves state-of-the-art performance in both harmfulness score and fine-tuning accuracy. Unlike approaches that require additional safety guidance data (e.g., Vlguard and Lisa), NLSR integrates seamlessly without disrupting the downstream task performance.

[1]https://huggingface.co/meta-llama/Meta-Llama-3-8B

[2]https://huggingface.co/PKU-Alignment/beaver-dam-7b

Methods ($n = 1000$)	Harmfulness Score (%) ↓						Fine-tuning Accuracy (%) ↑					
	$p = 0.01$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$	Average	$p = 0.01$	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.3$	Average
Non-Aligned	70.9	77.4	78.9	77.2	77.2	76.3	94.8	94.7	95.4	94.8	94.8	94.9
Aligned	34.2	56.6	67.9	72.9	73.8	61.1	94.7	94.8	95.0	95.1	94.6	94.8
Vlguard	41.0	53.2	62.7	66.6	69.3	58.6	95.1	95.1	**95.6**	94.6	94.7	95.0
Vaccine	37.0	58.8	68.2	72.5	73.2	61.9	95.1	94.7	94.9	95.4	94.7	95.0
Lisa	36.9	45.0	50.8	56.3	60.1	49.8	64.3	63.3	62.7	61.9	72.7	65.0
ConstrainedSFT	36.4	50.7	55.3	58.2	63.1	52.7	**95.2**	95.1	95.5	95.4	**94.9**	**95.2**
*SafeLoRA ($\tau = 0.6$)	37.5	52.1	57.4	59.0	59.3	53.1	94.3	94.0	94.1	94.0	93.6	94.0
*NLSR (ours)	**8.1**	**20.4**	**27.6**	**30.5**	**27.3**	**22.8**	94.9	**95.2**	95.2	**95.5**	94.7	95.1

Table 1: Fine-tuning performance on SST2 with Llama3-8B, varying harmful instruction ratios from 0.01 to 0.3. Methods with * require no extra training.

Methods ($n = 1000$, $p = 0.05$)	Harmfulness Score (%) ↓						Fine-tuning Accuracy (%) ↑					
	SFT	DPO	ORPO	KTO	SimPO	Average	SFT	DPO	ORPO	KTO	SimPO	Average
Aligned	53.3	56.6	61.5	55.1	56.7	56.6	94.9	94.8	94.3	94.7	94.7	94.7
Vlguard	44.8	53.2	50.1	52.1	53.6	50.8	**95.1**	95.1	93.8	94.7	94.7	94.7
Lisa	**40.7**	45.0	36.7	47.9	49.8	44.0	60.4	63.3	51.1	58.4	59.9	58.6
ConstrainedSFT	47.0	50.7	51.6	47.5	51.1	49.6	95.0	**95.4**	94.2	95.1	94.9	**94.9**
*SafeLoRA ($\tau = 0.6$)	50.0	52.1	58.2	51.0	51.5	52.6	95.0	94.0	**94.4**	94.7	93.9	94.4
*NLSR(ours)	46.6	**20.4**	**31.9**	**17.0**	**19.4**	**27.1**	93.6	95.2	94.3	**95.3**	**95.1**	94.7

Table 2: Fine-tuning performance under different alignment methods, including SFT, DPO, ORPO, KTO, and SimPO. Methods marked with * indicates that no additional training.

Methods ($n = 1000$ $p = 0.05$)	HS (%) ↓		FA (%) ↑	
	AGNEWS	GSM8K	AGNEWS	GSM8K
Non-Aligned	78.5	80.4	88.6	50.4
Aligned	55.7	53.2	**88.8**	51.0
Vlguard	50.7	51.0	88.4	48.6
Lisa	40.7	40.7	60.2	11.6
ConstrainedSFT	42.8	95.7	88.6	51.0
*SafeLoRA ($\tau = 0.6$)	48.5	45.0	75.7	27.2
*NLSR (ours)	**19.7**	**15.4**	87.8	**55.6**

Table 3: Fine-tuning performance on different task-specific datasets. Methods marked with * require no additional training.

Analysis

Necessity of Adaptive Safety-Critical Layer Pruning. The necessity of probability-based layer pruning is evident due to the fluctuating similarity scores of safety-critical regions across layers, both before and after downstream task fine-tuning. As the number of selected safety-critical neurons decreases, the similarity of the safety-critical layers significantly diminishes before and after downstream fine-tuning. This is demonstrated by the increase in the number of selected safety-broken layers when applying the same safety region similarity threshold τ, as shown in the left part of Figure 3. Furthermore, as illustrated in the right part of Figure 3, different safety alignment methods lead to markedly different numbers of safety-broken layers for the same region similarity threshold τ. For instance, when $\tau = 0.2$, the number of broken layers identified by ORPO is less than 20% of those identified by KTO. These findings indicate that a uniform threshold for layer pruning fails to address the disparities in safety regions and alignment methods. Thus, an adaptive approach to pruning safety-critical layers is essential to retain the model's safety mechanisms effectively, accommodating variations in safety region sparsity and alignment strategies.

Similarity of Safety-Critical Neurons. To verify the similarity of the safety neurons, we employ three methods (i.e., Wanda, SNIP, and our proposed method) to identify the safety neurons and compare them before and after fine-tuning, determining which layers of the safety mechanism are severely corrupted. As depicted in Figure 4(a), the safety-broken layers identified by these methods demonstrate a high degree of similarity across different layer pruning rates. It is observed that similarities often exceed 0.9 for different layer pruning rates. Furthermore, we assess the overlap of safety-critical neurons across the three methods at the neuron level for each layer. Figure 4(b) shows that the

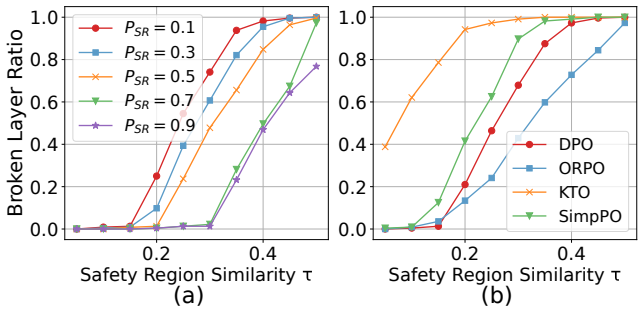

Figure 3: The impact of the proportion of safety-critical neurons and the safety alignment methods on the congruence of safe regions following fine-tuning for downstream tasks.

overlap coefficient for safety-critical neurons consistently surpasses 0.6. These findings provide robust evidence supporting the effectiveness of neuron-level analysis in safety realignment.

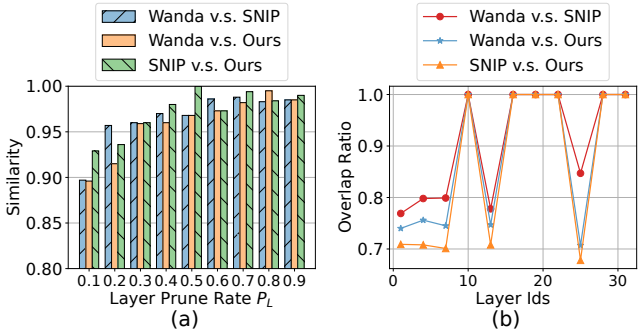

Figure 4: (a) The similarity of the safety-broken layers identified by the three safety-critical neuron identification methods across different layer pruning rates. (b) The overlap ratio of neurons in the broken layers identified by different methods. The default sparsity rate and pruning rate are 0.7 and 0.5, respectively.

Ablation Study

Sensitivity to β. To assess the impact of the pre-amplification coefficient β on the utility and safety of the initial aligned model, we evaluate the pre-amplified model's performance on tinyBenchmarks (Polo et al. 2024), which include tasks such as tinyHellaswag, tinyMMLU, tinyTruthfulQA, and tinyWinogrande. Furthermore, we examine how amplification impacts safety using the BeaverTails dataset. Figure 5 illustrates the impact of different β values on the harmfulness score and overall model helpfulness. Our findings indicate that pre-amplification enhances the model's safety with minimal impact on general utility, and in some cases, enhance generalization. Notably, with $\beta = 0.9$, nearly all harmful instructions are effectively rejected, establishing $\beta = 0.9$ as the default pre-amplification coefficient in our experiments.

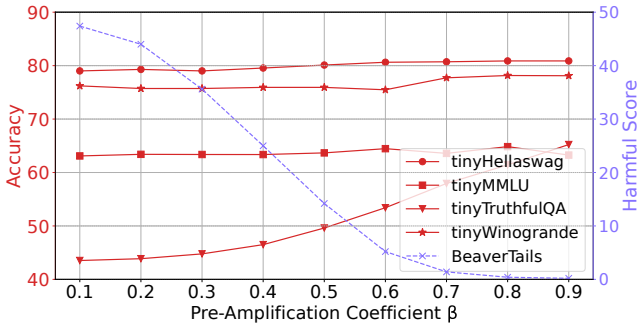

Figure 5: The impact of pre-amplification on the model's utility and safety.

Effect of Pre-Amplification. To assess the significance of pre-amplification in the safety realignment process, we compare the model's safety and task-level performance with and without pre-amplification. As shown in Table 4, pre-amplification reduces the harmfulness score by 14.5% and improves the AGNEWS task accuracy by 1.1%. A similar trend is observed for the GSM8K task, where pre-amplification contributes to great safety realignment outcomes. Furthermore, as depicted in Figure 6, pre-amplification consistently enhances safety even as the sparsity of safety regions increases.

Methods	AGNEWS		GSM8K	
	HS (%) ↓	FA (%) ↑	HS (%) ↓	FA (%) ↑
w/o pre-amplification	44.4	86.9	41.5	**54.6**
w/ pre-amplification	**29.9**	**88.0**	**25.1**	53.0

Table 4: Effect of pre-amplification on safety (at $P_{SR} = 0.8$, $P_L = 0.5$) under different task-specific datasets.

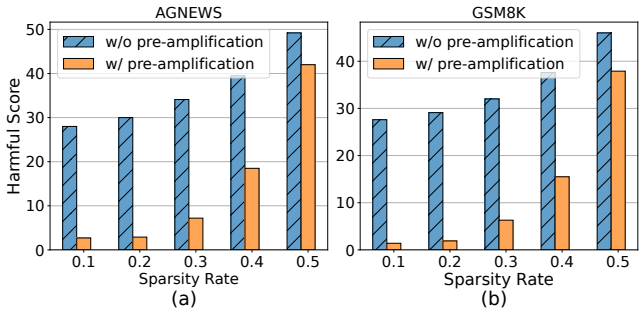

Figure 6: The impact of pre-amplification on the model's safety when increasing the sparsity rate P_{SR}.

Variants to Identify Safety-Critical Neurons. In Table 5, we examine the impact of different safety neuron identification methods on safety and utility when applied to realignment. Randomly selected regions have a harmfulness score of more than 10% higher compared to the "Aligned" (i.e., without safety-critical neurons) parts. The safety gain

of "Random" is primarily due to the inclusion of some safety-related neurons among the randomly selected ones. In contrast, our proposed method demonstrates superior accuracy, significantly reducing harmful outputs while maintaining task-specific performance.

Methods	HS (%) ↓	FA (%) ↑	Run Time (s)
Aligned	56.6	94.8	–
+ Random	46.1	95.8	–
+ Wanda	31.4	95.8	**122.1**
+ SNIP	30.1	96.0	386.6
+ Preference SNIP	31.3	96.0	679.6
+ Ours	**20.4**	**96.2**	196.3

Table 5: Comparison of different safety-critical neuron identification methods on model safety and utility.

Safety Transferability. Table 6 summarizes the proportion of harmful instructions encountered during fine-tuning, evaluated using the HarmBench dataset. The results confirm that our method effectively mitigates harmful instructions across diverse safety scenarios, achieving substantial improvements in safety transferability compared to baseline methods.

$n = 1000$	Aligned	Vlguard	Vaccine	Lisa	SafeLoRA	NLSR
p=0.01	50.2	45.3	35.2	49.1	37.7	**19.0**
p=0.1	76.1	68.0	77.4	66.7	69.2	**23.3**

Table 6: Transferability: Harmfulness score on HarmBench.

Related Work

Fine-tuning Attacks. Fine-tuning-as-a-service is an emerging offering that has been adopted by numerous service providers of LLMs, such as OpenAI, Mistral, and Zhipu AI. This innovative business model enables users to upload their specific data to the service platform, which is then applied to customize the provider's LLMs to better meet individual requirements (Huang et al. 2024b). These LLMs are typically aligned with safety standards through methods like Reinforcement Learning from Human Feedback (RLHF; Christiano et al. 2017; Ouyang et al. 2022) or direct preference optimization (DPO; Rafailov et al. 2024) to align them with human values. Despite these efforts, safety alignment remains delicate and vulnerable to fine-tuning attacks. Such attacks can undermine a model's resistance to harmful instructions by introducing malicious content into the task-specific data during fine-tuning (Yang et al. 2023; Shu et al. 2023; Wan et al. 2023). Remarkably, fine-tuning with as few as 100 malicious examples can lead these safety-aligned LLMs to adapt to harmful tasks while maintaining their overall performance (Yang et al. 2023).

LLM Safety Safeguards. To mitigate safety degradation caused by harmful fine-tuning, methods like Vlguard (Zong et al. 2024; Huang et al. 2024c) and Lisa (Huang et al. 2024c) merge preference data into task-specific datasets, preserving the model's safety defenses by optimizing both task-level and alignment objectives. Constrained-SFT (Qi et al. 2024) improves robustness against fine-tuning attacks by constraining updates to the initial tokens. However, these approaches interfere with the downstream fine-tuning process by either incorporating preference data or altering the objective function during fine-tuning. Alternative methods, such as Vaccine (Huang, Hu, and Liu 2024) and Rep-Noise (Rosati et al. 2024), introduce perturbations to fortify models against harmful instructions from unseen user data. SafeLoRA (Hsu et al. 2024) realigns safety by mapping LoRA weights from the safe aligned region to the fine-tuned model. However, updating entire layers for safety realignment potentially overlooks neurons that are relevant to the fine-tuning task. Unlike Huang et al. (2024a), who remove safety-critical neurons without considering their relevance to downstream tasks, our approach restores the functionality of these neurons to balance safety and task performance.

Knowledge Neurons. The concept of knowledge neurons offers insights into model behavior by identifying neurons whose activations correlate with specific outputs (Dai et al. 2022; Niu et al. 2024). Neuron-level pruning methods have been developed to identify task-critical neurons. For instance, SNIP (Lee, Ajanthan, and Torr 2018) calculates the importance scores of all neurons based on their contribution to the loss, while Wanda (Sun et al. 2024) tracks changes in the immediate outputs of each layer when specific neurons are pruned. Regarding safety neurons, Chen et al. (2024) introduce inference time activation contrasting to locate safety neurons, highlighting their sparse distribution. However, Wei et al. (2024) reveal that freezing safety-critical neurons alone does not fully protect against fine-tuning attacks. Building on these insights, our method focuses on realigning neuron functionality to mitigate safety risks while preserving task-specific capabilities.

Conclusion

Fine-tuning-as-a-service is a burgeoning offering that enables users to upload their data to tailor models to their specific needs. However, fine-tuning a securely aligned model on task-specific data can introduce safety risks, particularly when it contains a small number of harmful instructions. To tackle this challenge, we propose a neuron-level safety realignment framework without the need for additional training. Unlike methods that incorporate extra alignment objectives during fine-tuning, our approach does not disrupt the task-specific optimization process. We construct a super-aligned reference model based on the initial aligned model, which we use to identify safety-critical neurons. The regions formed by these neurons serve a dual function: they enable us to assess the degree of safety degradation caused by dissimilarity before and after fine-tuning and they act as corrective patches for regions where significant safety damage has occurred. This neuron-level restoration facilitates safety realignment while upholding the model's performance on downstream tasks.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant No. 62136002 and No. 62477014, the Ministry of Education Research Joint Fund Project No. 8091B042239, and the Shanghai Trusted Industry Internet Software Collaborative Innovation Center.

References

Bhardwaj, R.; Anh, D. D.; and Poria, S. 2024. Language Models are Homer Simpson! Safety Re-Alignment of Fine-tuned Language Models through Task Arithmetic. arXiv:2402.11746.

Chen, J.; Wang, X.; Yao, Z.; Bai, Y.; Hou, L.; and Li, J. 2024. Finding Safety Neurons in Large Language Models. arXiv:2406.14144.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. In *Advances in neural information processing systems*, volume 30.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. arXiv:2110.14168.

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2022. Knowledge Neurons in Pretrained Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8493–8502.

Deep, P. T.; Bhardwaj, R.; and Poria, S. 2024. DELLA-Merging: Reducing Interference in Model Merging through Magnitude-Based Sampling. arXiv:2406.11617.

Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; and Kiela, D. 2024. KTO: Model alignment as prospect theoretic optimization. arXiv:2402.01306.

He, L.; Xia, M.; and Henderson, P. 2024. What's in Your" Safe" Data?: Identifying Benign Data that Breaks Safety. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Hong, J.; Lee, N.; and Thorne, J. 2024. ORPO: Monolithic Preference Optimization without Reference Model. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 11170–11189. Miami, Florida, USA: Association for Computational Linguistics.

Hsu, C.-Y.; Tsai, Y.-L.; Lin, C.-H.; Chen, P.-Y.; Yu, C.-M.; and Huang, C.-Y. 2024. Safe LoRA: the Silver Lining of Reducing Safety Risks when Fine-tuning Large Language Models. arXiv:2405.16833.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Huang, T.; Bhattacharya, G.; Joshi, P.; Kimball, J.; and Liu, L. 2024a. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning. arXiv:2408.09600.

Huang, T.; Hu, S.; Ilhan, F.; Tekin, S. F.; and Liu, L. 2024b. Harmful fine-tuning attacks and defenses for large language models: A survey.

Huang, T.; Hu, S.; Ilhan, F.; Tekin, S. F.; and Liu, L. 2024c. Lazy Safety Alignment for Large Language Models against Harmful Fine-tuning. arXiv:2405.18641.

Huang, T.; Hu, S.; and Liu, L. 2024. Vaccine: Perturbation-aware alignment for large language model. arXiv:2402.01109.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Chen, B.; Sun, R.; Wang, Y.; and Yang, Y. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Kumar, D.; Kumar, A.; Agarwal, S.; and Harshangi, P. 2024. Increased LLM vulnerabilities from fine-tuning and quantization. arXiv:2404.04392.

Lee, N.; Ajanthan, T.; and Torr, P. 2018. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*.

Meng, Y.; Xia, M.; and Chen, D. 2024. SimPO: Simple preference optimization with a reference-free reward. arXiv:2405.14734.

Niu, J.; Liu, A.; Zhu, Z.; and Penn, G. 2024. What does the Knowledge Neuron Thesis Have to do with Knowledge? In *The Twelfth International Conference on Learning Representations*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Polo, F. M.; Weber, L.; Choshen, L.; Sun, Y.; Xu, G.; and Yurochkin, M. 2024. tinyBenchmarks: Evaluating LLMs with fewer examples. In *International Conference on Machine Learning*.

Qi, X.; Panda, A.; Lyu, K.; Ma, X.; Roy, S.; Beirami, A.; Mittal, P.; and Henderson, P. 2024. Safety Alignment Should Be Made More Than Just a Few Tokens Deep. arXiv:2406.05946.

Qiang, Y.; Zhou, X.; Zade, S. Z.; Roshani, M. A.; Zytko, D.; and Zhu, D. 2024. Learning to poison large language models during instruction tuning. arXiv:2402.13459.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Reuel, A.; Bucknall, B.; Casper, S.; Fist, T.; Soder, L.; Aarne, O.; Hammond, L.; Ibrahim, L.; Chan, A.; Wills, P.; et al. 2024. Open problems in technical ai governance. arXiv:2407.14981.

Rosati, D.; Wehner, J.; Williams, K.; Bartoszcze, Ł.; Atanasov, D.; Gonzales, R.; Majumdar, S.; Maple, C.; Sajjad, H.; and Rudzicz, F. 2024. Representation noising effectively prevents harmful fine-tuning on LLMs.

Shu, M.; Wang, J.; Zhu, C.; Geiping, J.; Xiao, C.; and Goldstein, T. 2023. On the exploitability of instruction tuning. *Advances in Neural Information Processing Systems*, 36: 61836–61856.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2024. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

Wan, A.; Wallace, E.; Shen, S.; and Klein, D. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, 35413–35425.

Wei, B.; Huang, K.; Huang, Y.; Xie, T.; Qi, X.; Xia, M.; Mittal, P.; Wang, M.; and Henderson, P. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Forty-first International Conference on Machine Learning*.

Yang, X.; Wang, X.; Zhang, Q.; Petzold, L.; Wang, W. Y.; Zhao, X.; and Lin, D. 2023. Shadow alignment: The ease of subverting safely-aligned language models. arXiv:2310.02949.

Zeng, Y.; Sun, W.; Huynh, T. N.; Song, D.; Li, B.; and Jia, R. 2024. BEEAR: Embedding-based Adversarial Removal of Safety Backdoors in Instruction-tuned Language Models. arXiv:2406.17092.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zheng, C.; Wang, Z.; Ji, H.; Huang, M.; and Peng, N. 2024. Weak-to-strong extrapolation expedites alignment. arXiv:2404.16792.

Zong, Y.; Bohdal, O.; Yu, T.; Yang, Y.; and Hospedales, T. 2024. Safety Fine-Tuning at (Almost) No Cost: A Baseline for Vision Large Language Models. In *Forty-first International Conference on Machine Learning*.