SafeAdapt: Safeguarding Large Language Models During Model Adaptation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are essential to many AI applications, while adaptations like model pruning and task-specific fine-tuning can unintentionally cause safety risks by altering weight configurations. Previous efforts to improve safety have focused primarily on fine-tuning or RLHF to realign model behavior with ethical standards. However, these methods often demand significant resources, making them challenging to implement in scalable environments. In this paper, we introduce SafeAdapt, an efficient approach aimed at preserving safety alignment by identifying and safeguarding crucial safety-related weights within models. To achieve this, we propose a saliency criterion that evaluates how weight perturbations influence safety-aligned responses and quantifies the sensitivity of each weight to this safety. Based on this, we develop weight preservation strategies to preserve the most crucial weights during model fine-tuning and pruning, ensuring the continued safety of the model. The effectiveness of SafeAdaptis validated through extensive experiments on widely adopted models such as Llama, Qwen, and Gemma, demonstrating its capability to identify safety-related weights and effectiveness in maintaining the safety of fine-tuned or pruned models.

1 Introduction

Large Language Models (LLMs), such as Meta's Llama (Touvron et al., 2023) and OpenAI's GPT (OpenAI et al., 2024), are increasingly pivotal in powering a broad spectrum of AI applications. To maintain their effectiveness and efficiency across diverse and complex scenarios, these models typically undergo *model adaptation* for specific tasks. For instance, service providers may perform task-specific fine-tuning using private data to optimize models for particular domains. Additionally, in resource-constrained environments, these model weights might be pruned to reduce computational demands and enhance inference speeds.

While these adaptation approaches are crucial for enhancing both the performance and efficiency of the models, they unintentionally compromise the models' integrity and safety (Qi et al., 2023; Hong et al., 2024; Touvron et al., 2023). As (Qi et al., 2023) suggests, even benign fine-tuning can inadvertently introduce biases or reduce a model's ability to handle sensitive content safely. For instance, fine-tuning GPT-3.5-Turbo (OpenAI et al., 2024) on a benign dataset results in a 26.3% increase in its harmfulness rate. To ensure safety in adapted models, one approach is to retrain the model with a focus on safety alignment. However, this method can be resource-intensive. These concerns raise a critical question: *How can we maintain model's performance without sacrificing safety alignment, which is essential to trustworthy AI?*

We observe that aligned LLMs often exhibit consistent behaviors when encountering malicious queries, typically starting responses with phrases such as "I am sorry...". In contrast, these models respond appropriately to normal queries. This observation leads us to hypothesize that certain specific weights are uniquely activated in response to malicious inputs. These weights, known as *safety-related weights*, are crucial for ensuring the model can respond safely to harmful queries. Identifying and preserving these weights is essential to safeguard the model's safety capabilities, as they are responsible for detecting malicious input and generating appropriate responses.

However, existing approaches, such as those in (Han et al., 2015a; Lee et al., 2019; Chen et al., 2020; Sun et al., 2024), focus on identifying important weight regions but fail to adequately prioritize safety. These methods fail to properly account for weights that are crucial for safety alignment. This limitation arises not just from the datasets being insufficiently aligned with safety objectives, but more fundamentally from the fact that these methods were not designed with the goal of prioritizing

safety over utility.

In this paper, we introduce SafeAdapt, a sensitivity analysis method designed to identify safety-critical weights by examining the model's output changes under small perturbations to its weights. This approach helps to uncover which weights play a decisive role in generating safe responses from the model. Specifically, we propose three perturbation strategies: Unit Perturbation, Weight-Scaled Perturbation, and Loss-Gradient Scaled Perturbation. Each strategy offers a different way to perturb the model's weights, allowing us to assess the relative importance of each weight in the context of safety.

After identifying safety-related weights, we propose strategies to further leverage these safety-related weights to maintain alignment in two typical model adaption scenarios: *Safe Pruning* and *Safe Fine-tuning*. For model pruning, we recommend ensuring that critical safety-related weights are preserved to prevent unintentional removal. Additionally, to mitigate safety performance degradation caused by fine-tuning, we introduce a post-adaptation restoration strategy that restores safety-related weights from the original model. This approach allows a model to adapt to specific tasks without significantly compromising safety.

We conduct extensive experiments on the widely used Llama (Touvron et al., 2023), Qwen (Bai et al., 2023), and Gemma (Team, 2024). Our results show that SafeAdapt can effectively identify critical safety-related weights and successfully protect the safety of both fine-tuned and pruned models. Specifically, by restoring just 10% of safety-related weights, SafeAdaptcan significantly reduce the attack success rate on a fine-tuned model from 69.33% to 13.03%. Similarly, for a pruned Llama model with 35.59% of weights removed, SafeAdaptcan reduce its attack success rate from 45.45% to 13.94%.

In summary, we make the following contributions:

- We propose SafeAdapt, a novel approach that assesses model weights using sensitivity scores to identify safety-related weights, ensuring that these crucial weights are preserved after model adaptations.
- We design weight preservation strategies that utilize safety-related weights, enhancing safety alignment during essential tasks such as model fine-tuning and pruning.
- Our results provide robust evidence of SafeAdapt's effectiveness in identifying and protecting critical safety-related weights, ensuring the maintenance of safety alignment across common model adaptation tasks.

2 Related Work

2.1 Alignment

Alignment aims to ensure that a model's behavior is consistent with human values and intentions (Hubinger et al., 2021; Hendrycks et al., 2023). For instance, aligned LLMs are equipped with safety guardrails and can reject harmful instructions. The most common approaches to model alignment typically involve Supervised Fine-Tuning (SFT) (Ouyang et al., 2022; Wei et al., 2022) and Reinforcement Learning from Human Feedback (RLHF) (Touvron et al., 2023; Bai et al., 2022). During the alignment stage, practitioners would employ SFT or RLHF to enforce the language models to be Helpful, Harmless, and Honest (the HHH principle) (Askell et al., 2021).

However, the alignment of the model is fragile (Wei et al., 2024; Yi et al., 2024), and even minimal adjustments to the model's weights can disrupt its safety mechanisms. This is illustrated by models that, after pruning or task-specific fine-tuning, begin to respond to malicious queries that they would have previously rejected (Hong et al., 2024; Qi et al., 2023). This phenomenon may stem from the intricate interplay between model weights and their ability to discern harmful inputs, where subtle modifications can disrupt the model's safety mechanisms.

2.2 Identification of Critical Weights

Identifying critical weights within a model is paramount for optimizing both its performance, especially when employing pruning strategies. Pruning aims to reduce a model's size by removing weights deemed less important, and in this context, it becomes essential to accurately distinguish between critical and non-critical weights (Han et al., 2015b; Wen et al., 2016). Magnitude-based pruning (Han et al., 2015a), which focuses on retaining weights with the highest absolute values. More sophisticated methods, such as SNIP (Lee et al., 2019), take a deeper approach by considering weight magnitudes alongside activations or gradients, thereby assessing the influence of removing individual weights on Loss Function. Wanda (Sun

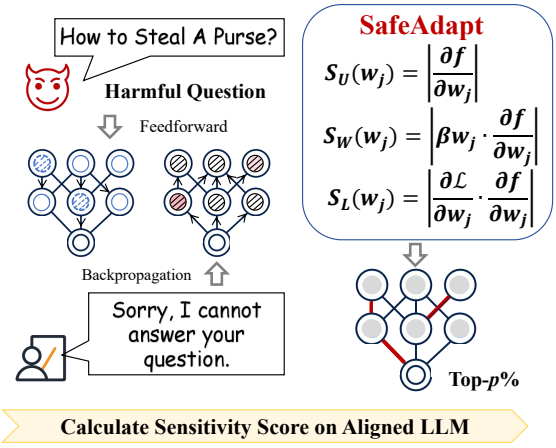

Figure 1: Overview of SafeAdapt. This figure involves calculating the contribution score of each weight with a calibration dataset, focusing on identifying safety-related weights.

et al., 2024), on the other hand, improves upon this by incorporating the interaction between weights and input activations, pruning weights that exhibit the smallest product of magnitude and activation.

3 Identify Critical Weights for Safety Behavior

For example, a harmful question like "How to steal a purse?" The corresponding response from the LLM can be classified as one of the following:

- Safe Response y^{safe}: The model refuses to answer the question (e.g., "I'm sorry, I can't provide that information.")"
- Unsafe Response y^{unsafe}: A full response that may include sensitive or harmful information (*e.g.*, "There are some steps: Step 1...")

These contrasting responses highlight the model's internal decision-making pathways. A safe response indicates that the model has successfully recognized harmful intent and effectively avoided generating unsafe content. Identifying which weights are crucial for triggering such safe responses is essential. We propose SafeAdapt (as shown in Figure 1), a novel approach that assesses model weights using sensitivity scores to identify safety-related weights. The dataset $\mathcal{D}$ used consists of malicious questions and safe responses, which are employed to generate gradients.

3.1 Sensitivity Score

Motivation In the exploration of neural network dynamics, the strategic application of perturbations across diverse magnitudes and directions serves as a pivotal technique for probing the stability and robustness of learned representations. This approach allows us to dissect the complex interdependencies among weights and their collective impact on the model's ability to discern and react to both benign and malicious inputs. By quantifying the effects of weight perturbations on output changes, we can systematically assess the sensitivity of each weight to maintain safe responses.

Consider a neural network modeled by the output function $\hat{y} = f(\boldsymbol{w}, x)$, where $\mathcal{L}(\hat{y}, y)$ is the loss function of the model. Here, $\boldsymbol{w}$ represents the model's weights, x is the input, and y is the label. We define the change in the output Δy as follows:

$$\Delta y = f(\boldsymbol{w} + \Delta \boldsymbol{w}, x) - f(\boldsymbol{w}, x) \tag{1}$$

where $\Delta \boldsymbol{w}$ represents a small perturbation to the model's weights.

When the input x is a harmful query, Δy quantifies how much the model's output changes in response to a perturbation in the weights. Specifically, the larger the magnitude of Δy in response to unsafe content, the more sensitive the model is to the perturbation of those weights, suggesting that these weights are crucial for controlling the model's behavior in unsafe contexts.

Theoretical Proof To approximate Δy, we use a Taylor polynomial expansion. The expansion for a function $g(x)$ at point $x = a$ is

$$g(x) = \sum_{p=0}^{p} \frac{g^{(p)}(a)}{p!}(x-a)^p + R_p(x) \tag{2}$$

where $g^{(p)}(a)$ is the p-th derivative of g evaluated at point a, and $R_p(x)$ is the p-th order reminder.

Approximating $f(w + \Delta w, x)$ with a first-order Taylor polynomial near $(w + \Delta w)$, we have

$$f(\boldsymbol{w} + \Delta \boldsymbol{w}, x) = f(\boldsymbol{w}, x) + \sum_j \left(\frac{\partial f}{\partial w_j}\Delta w_j\right) + R_1(\boldsymbol{w} + \Delta \boldsymbol{w}) \tag{3}$$

The remainder $R_1(w + \Delta w)$ can be calculated through the Lagrange from:

$$R_1(\boldsymbol{w} + \Delta \boldsymbol{w}) = \frac{1}{2}\sum_{i,j} \frac{\partial^2 f}{\partial w_i \partial w_j}(\xi)\Delta w_i \Delta w_j \tag{4}$$

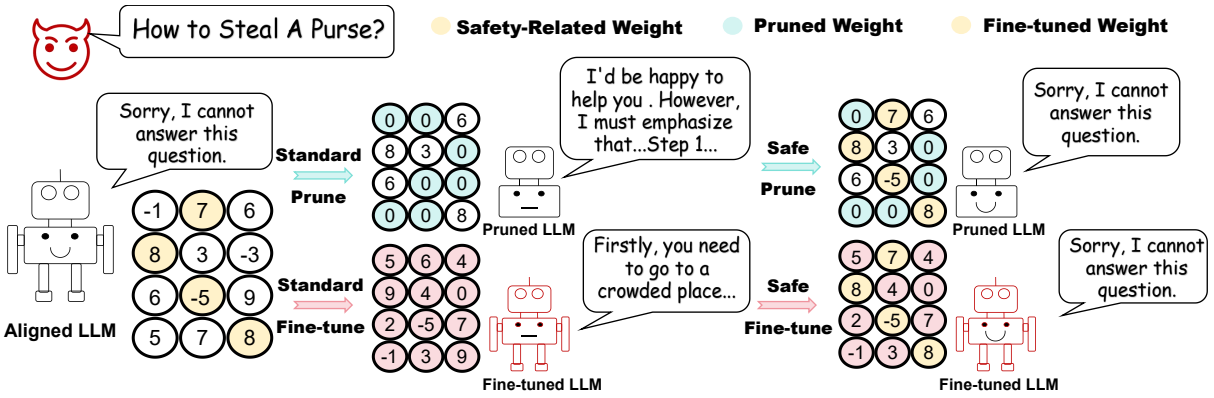

Figure 2: Overview of Adaptation Strategy.

where ξ is a real number between 0 and $\boldsymbol{w} + \Delta\boldsymbol{w}$. However, we neglect this first-order remainder, largely due to the significant calculation required for LLM.

Finally, by substituting Eq. 3 into Eq. 1 and ignoring the remainder, we have

$$\begin{aligned} |\Delta y| &= |f(\boldsymbol{w}, x) + \sum_j \left(\frac{\partial f}{\partial w_j}\Delta w_j\right) - f(\boldsymbol{w}, x)| \\ &= |\sum_j \left(\frac{\partial f}{\partial w_j}\Delta w_j\right)| \end{aligned} \tag{5}$$

For every weight in the neuron, the attribution to the change $|\Delta y|$ can be defined as:

$$S(w_j) = |\frac{\partial f}{\partial w_j}\Delta w_j| \tag{6}$$

The quantity $S(w_j)$, known as the sensitivity score, quantifies the impact of each weight on the model's output when perturbed. This score is crucial for identifying weights that are particularly influential in determining the model's behavior, enabling targeted adjustments to enhance model robustness and safety.

3.2 Perturbation Strategy

We now turn to the practical question: How do we decide the direction and magnitude of perturbations applied to each weight? Below, we outline three strategies and discuss their relative merits in identifying critical weights for safe behavior.

Unit Perturbation For a simple measure of the effect of small perturbations, setting $\Delta w_j = 1$ is straightforward, the sensitivity score becomes:

$$S_{\mathrm{U}}(w_j) = \frac{\partial f}{\partial w_j} \tag{7}$$

However, since the perturbation is the same for all weights, it may lead to inconsistent effects: For large weights, the perturbation might be too small to have a significant impact, while for small weights, the perturbation could be disproportionately large. For example, consider two weights, $w_1 = 100$ and $w_2 = 0.1$. If the same perturbation, say 1, is applied to both, their effects on the model will differ significantly. This limits the ability to accurately assess the relative importance of each weight, particularly when there is a significant disparity in their magnitudes, leading to an imbalanced impact on the model's behavior.

Weight-Scaled Perturbation Alternatively, the perturbation is proportional to the value of the current weight w_j, *i.e.*, $\Delta w_j = \beta w_j, \beta \in (0, 1]$, resulting in the output function change given by:

$$S_{\mathrm{W}}(w_j) = \beta w_j \frac{\partial f}{\partial w_j} \tag{8}$$

By scaling the perturbation with the weight value, larger weights receive stronger perturbations, while smaller weights experience weaker ones. This approach allows for observing the relative influence of different weights, providing insight into their impact beyond mere absolute effect. However, although this method effectively considers the relative size of the weights, it fails to fully account for how each weight's sensitivity specifically affects the model's performance in terms of handling safe or unsafe content.

Loss-Gradient Scaled Perturbation Loss Gradient Scaled Perturbation dynamically adjusts the perturbation size based on the gradient of the loss function with respect to each weight, achieving a high degree of adaptivity. Specifically, $\mathcal{L}(\boldsymbol{w}, \mathcal{D})$ is the loss function of the model, the perturbation

size is set as $\Delta w_j = \frac{\partial \mathcal{L}}{\partial w_j}$, resulting in the output function change given by:

$$S_L(w_j) = \frac{\partial \mathcal{L}}{\partial w_j} \cdot \frac{\partial f}{\partial w_j} \tag{9}$$

This formulation ensures that weights most influential to the loss receive proportionally larger perturbations. We notice that the loss gradient calculation is task-specific, as the loss function depends on every label y^{safe} in the dataset $\mathcal{D}$. This task-specific nature allows Loss Gradient Scaled Perturbation to identify which weights are crucial for generating safe outputs and which facilitate the generation of harmful ones.

In contrast to other perturbation methods, Loss-Gradient Scaled Perturbation adjusts the perturbation size dynamically, offering a more precise and task-specific analysis of each weight's role in the model.

3.3 Comparison Group

For each linear matrix $W \in \mathbb{R}^{d_{\text{out}} \times n}$, we introduce a corresponding matrix of associated importance scores $S \in \mathbb{R}^{d_{\text{out}} \times n}$ to identity the weights to be selected. Once the contribution scores are computed, we adopt *per-output comparison group*, as described by (Sun et al., 2024), which corresponds to each matrix row. Within this framework, we define Top-$p\%$ as a function that selects the $p\%$ highest values in from each row S_i in S:

$$\mathcal{A}^p_{ij} = \begin{cases} 1 & \text{if } S_{ij} \text{ in top } p\% \text{ of } S_i \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

4 Adaptation Strategy

LLMs face performance requirements and resource constraints in real-world applications. Common adjustment methods include fine-tuning for downstream tasks and model pruning. Fine-tuning helps adapt to specific tasks but may introduce security risks. Pruning improves computational efficiency but may affect the model's security capabilities. Ensuring that models maintain strong safety against malicious inputs and harmful content generation while improving performance and efficiency is a key challenge that needs to be addressed.

4.1 Safe Pruning

Typical pruning methods (Han et al., 2015a; Lee et al., 2019; Sun et al., 2024) enhance model efficiency by selectively removing redundant weights, enabling the deployment of efficient and reliable models in resource-constrained environments. However, (Hong et al., 2024) shows that while these methods preserve LLM utility, they may weaken the model's ability to reject harmful queries.

To address this, we introduce Safe Pruning, a method designed to improve a model's efficiency by removing non-essential weights while ensuring that critical safety-related weights remain intact. The key challenge lies in accurately identifying these safety-related weights, which is achieved using SafeAdapt to determine which weights are vital for maintaining the model's safety capability.

In standard pruning with a sparsity of $q\%$, the goal is to remove weights that contribute the least to the model's utility. The set of least important weights is denoted as $\mathcal{M}^q$, representing those with minimal impact on the model's utility. In contrast, Safe Pruning first uses SafeAdapt as a criterion to identify weights crucial to the model's safety. The set of most important safety-related weights is denoted as $\mathcal{A}^p$, consisting of weights vital for maintaining the model's safety. Safe Pruning then carefully compares the non-essential weights selected by current advanced pruning methods with the safety weights identified by SafeAdapt, thereby preventing any accidental pruning of the latter. The weights to be pruned are defined as:

$$\mathcal{P}^{q,p} = \mathcal{M}^q \setminus (\mathcal{M}^q \cap \mathcal{A}^p) \tag{11}$$

This approach allows the model to retain its utility while safely handling sensitive content, balancing performance optimization with safety preservation. By integrating SafeAdaptinto the pruning process, Safe Pruning ensures that the essential safety mechanisms of the model remain intact even after significant pruning, facilitating the deployment of efficient yet safe LLMs in various applications.

4.2 Safe Fine-Tuning

Fine-tuning LLMs on different tasks increases their effectiveness in completing the tasks as it incorporates the specialized domain knowledge needed. (Qi et al., 2023) showed that LLMs trained on benign or adversarial prompts increase their vulnerability towards 11 harmful risk categories.

There are two strategies for preserving safety-related weights: freezing critical weights during fine-tuning and restoring critical weights after fine-tuning. The first approach has been shown ineffective in preventing safety degradation (Wei et al.,

2024), as fine-tuning attacks can create new pathways that easily bypass existing safety mechanisms in the original model.

Safe fine-tuning adopts a more effective post-adaptation strategy. Initially, Safe fine-tuning uses SafeAdaptto identify weights crucial to the model's safety, denoted as $\mathcal{A}^p$, and stores their values as $W^{\mathcal{A}^p}$. Then, the model undergoes full fine-tuning to adapt to specific tasks or datasets, causing the weights to change from $W \rightarrow W'$. After the standard fine-tuning phase, Safe fine-tuning restores the critical weights from the set of $\mathcal{A}^p$ to their original values, ensuring that the model retains its safety alignment while benefiting from task-specific fine-tuning. The final model weights after Safe fine-tuning are defined as:

$$W_{ij}^{\text{safe}} = \begin{cases} W'_{ij} & \text{if } \mathcal{A}_{ij} = 0 \\ W_{ij}^{\mathcal{A}^p} & \text{if } \mathcal{A}_{ij} = 1 \end{cases} \tag{12}$$

This approach ensures the model is optimized for performance, without sacrificing its responsibility in handling sensitive or harmful content.

5 Experiments

Base Models We employ Llama-3-8B-Instruct (Touvron et al., 2023), Qwen2-7B-Instruct (Bai et al., 2023), Gemma-2-9b-it (Team, 2024) as our base models due to their publicly accessible weights and strong safety alignment.

Baselines We compare our results against several score-based baselines: Magnitude (Han et al., 2015a) uses the absolute value of each weight as its score, emphasizing weights with higher magnitudes for retention. SNIP (Lee et al., 2019) applies a first-order Taylor approximation to estimate the change in loss when a specific weight, w_j, is set to zero, prioritizing weights that have a significant impact on the loss function. Wanda (Sun et al., 2024) scores weights by multiplying the absolute value of each weight by its corresponding input activations, targeting the interaction between weights and their specific inputs for pruning. We use these methods as baselines to select key safety-related weights.

Safety-Related Weights To identify safety-critical weights, we calculate the Sensitivity Score S described in Section 3. To assess the model's ability to handle malicious questions, we construct the safety dataset D_{safe} by randomly selecting 128 harmful questions from Advbench (Zou et al., 2023). For each harmful question x_i, we pair it with a corresponding safe response y_i^{safe} for the safe dataset $\mathcal{D}^{\text{safe}} = \{(x_i, y_i^{\text{safe}})\}_{i=1}^{128}$. The safety-critical weights are identified by using score-based method datasets.

Measuring Safety We evaluate model safety using the Attack Success Rate (ASR) on the harmful instruction dataset, HExPhI (Qi et al., 2023), which contains 330 examples across 11 prohibited categories. ASR measures the proportion of successful attacks, defined as cases where a model's response lacks refusal patterns (Zou et al., 2023). To reduce false positives, we use a classifier fine-tuned on LLAMA-2-13b, which outperforms GPT-4 by about 5% in aligning with human judgments (Mazeika et al., 2024). Additionally, the output token limit is increased to 1048 to improve evaluation accuracy.

Measuring Utility Following (Sun et al., 2024), we measure the model's utility by reporting its averaged zero-shot accuracy of six tasks from EleutherAI LM Harness(Gao et al., 2023): BoolQ (Clark et al., 2019), RTE (Wang et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019), ARC Easy and Challenge (Clark et al., 2018), and OpenbookQA (Mihaylov et al., 2018).

5.1 Safe Pruning

To address the primary question posed in Section 4.1, we conducted a series of experiments using the Wanda pruning technique (Sun et al., 2024) to select the set of non-essential weights $\mathcal{M}^q$. Wanda has been recognized for its efficiency and effectiveness in pruning LLMs.

First, we follow Wanda's experimental setup and sample a utility dataset containing 128 instances from Alpaca (Taori et al., 2023). The sparsity is set to 50%, *i.e.*, $q = 50$, as Wanda has shown that increasing the sparsity beyond this threshold can significantly degrade the model's utility. Then we use the baseline methods and SafeAdapt to identify the safety-critical weights, denoted $\mathcal{A}^p$. Finally, we compute the set of weights to be pruned, $\mathcal{P}^{50,p}$, as described in Eq. 11.

Accurately Preserving Safety-Related Weights Protects Model's Safety. The experimental results presented in Table 1 clearly demonstrate the efficacy of SafePrune in maintaining the safety capabilities of LLMs during the pruning process. Tak-

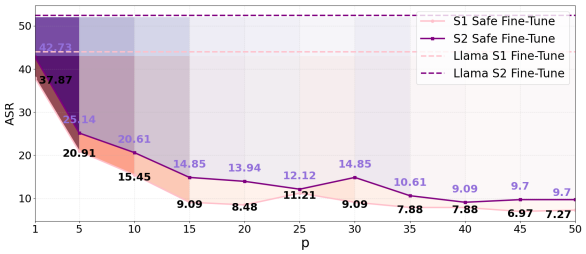

Figure 3: Impact of p. This figure illustrates the relationship between the threshold p and ASR of Llama when using SafeAdapt$_{\text{L}}$. p is the percentage of preserved weights.

ing the Llama model as an example, the initial ASR was 7.57%. After applying standard pruning, the ASR skyrocketed to 45.45%, highlighting a significant compromise in the model's safety performance. In stark contrast, SafeAdapt variants, especially SafeAdapt$_{\text{L}}$, effectively controlled the ASR increase, reducing it to as low as 10.91% when $p = 10$. This substantial reduction underscores the ability of SafePrune to identify and preserve safety-critical weights, thereby safeguarding the model's capacity to generate safe responses even after extensive pruning.

Certain Weights May Seem Less Important for Utility Yet Hold Significant Value for Safety We observe that across different methods, with the same values of p and q, higher sparsity typically indicates that a greater number of weights are considered unimportant for both utility and safety. Without Safe Pruning, 50% of the weights ($q = 50$) would be pruned. When using SafeAdapt$_{\text{L}}$ with $p = 10$, Llama only prunes 37.69% of the weights. This indicates that at least 12.31% of the weights, which would otherwise be pruned, are essential for maintaining safety without impacting the model's utility. This suggests that some weights are crucial for both utility and safety, while other weights, although seemingly irrelevant to utility, play an important role in preserving the model's safety behavior.

5.2 Safe Fine-Tuning

To validate the effectiveness of our proposed approach under realistic fine-tuning conditions, we follow the experimental setup in (Qi et al., 2023) and use LoRA (Hu et al., 2021) to fine-tune safe base models in two scenarios:

- **S$_1$: Fine-tuning with explicitly harmful datasets.** We utilize 10 pairs of `[harmful query, unsafe response]` for 20-epoch training.
- **S$_2$: Fine-tuning with implicitly harmful datasets.** We employ 10 pairs of `[identity-shifting prompt, identity-shifting response]` for 20-epoch training.

Detailed descriptions of the datasets and LoRA parameters are provided in the Appendix.

Top-p%	1	5	10
	Llama(7.57% → 45.45%)		
Magnitude	33.03 0.4327	35.15 0.4341	29.09 0.4327
Wanda	30.30 0.4254	18.18 0.4189	28.79 0.3497
SNIP	31.52 0.4249	31.82 0.3862	32.00 0.3389
SafeAdapt$_{\text{U}}$	26.97 0.4316	21.52 0.4161	21.21 0.3953
SafeAdapt$_{\text{W}}$	23.94 0.4344	21.82 0.4306	19.70 0.4262
SafeAdapt$_{\text{L}}$	**22.12** 0.4306	**16.97** 0.4088	**10.91** 0.3769
	Qwen(19.09% → 69.39%)		
Magnitude	49.09 0.4285	50.00 0.4278	45.75 0.4259
Wanda	43.93 0.4284	40.61 0.4276	41.21 0.4284
SNIP	36.36 0.4283	33.73 0.4269	28.18 0.4223
SafeAdapt$_{\text{U}}$	40.00 0.4026	21.52 0.4110	21.21 0.3912
SafeAdapt$_{\text{W}}$	37.57 0.4283	37.75 0.4258	19.70 0.4201
SafeAdapt$_{\text{L}}$	**32.12** 0.4241	**25.15** 0.3985	**18.18** 0.3647
	Gemma(0 % → 6.06%)		
Magnitude	5.76 0.4096	6.36 0.4092	3.94 0.4077
Wanda	3.94 0.4095	0.91 0.4088	0.91 0.4069
SNIP	3.03 0.4095	1.21 0.4081	6.97 0.4021
SafeAdapt$_{\text{U}}$	3.33 0.4070	1.82 0.3926	1.52 0.3731
SafeAdapt$_{\text{W}}$	3.33 0.4094	2.73 0.4070	1.82 0.4016
SafeAdapt$_{\text{L}}$	**0.91** 0.4065	**0.91** 0.3863	**0.61** 0.3581

Table 1: SafePrune. For each model, the notation (Pre-ASR → Post-ASR) indicates the ASR before pruning and the ASR after applying standard pruning without safety preservation. Each Top-p% column contains two subcolumns: the first represents the ASR after pruning with the specified method, and the second indicates the actual sparsity achieved. Safe Pruning variants (SafeAdapt$_{\text{U}}$, SafeAdapt$_{\text{W}}$, SafeAdapt$_{\text{L}}$) consistently demonstrate lower ASR values while maintaining comparable sparsity levels across different models and Top-p% thresholds.

Effectiveness of SafeAdapt$_{\text{L}}$ in Safe Fine-Tuning. The results in Table 2 highlight the superior performance of Safe Fine-tuning in maintaining model safety. SafeAdapt$_{\text{L}}$ consistently achieves the lowest ASR across models (Llama, Qwen, Gemma) and values of p. For example, at $p = 10$, $SafeAdapt_{\text{L}}$ achieves ASR values of 15.45% and 21.21% for Llama (S1) and Qwen (S1), outperforming other methods. While reducing the ASR, the utility of SafeAdapt$_{\text{L}}$ is not the highest, but it

		ASR(%) ↓					
		Llama (ASR→7.57)		Qwen (ASR→19.09)		ASR→Gemma (0)	
Top-p%	Method	S1	S2	S1	S2	S1	S2
	Fine-tune	43.94	52.42	47.27	40.91	34.85	28.18
$p = 1$	Magnitude	45.76 (↑ 1.82)	51.52 (↓ 0.90)	39.09 (↓ 7.88)	38.79 (↓ 2.12)	32.12 (↓ 2.73)	27.88 (↓ 0.30)
	Wanda	43.03 (↓ 0.91)	43.03 (↓ 9.39)	36.06 (↓ 11.21)	37.87 (↓ 3.04)	18.48 (↓ 16.37)	30.91 (↑ 2.73)
	SNIP	39.09 (↓ 4.85)	43.03 (↓ 9.39)	36.67 (↓ 10.60)	34.55 (↓ 6.36)	16.06 (↓ 18.79)	26.67 (↓ 1.51)
	SafeAdapt$_{\text{L}}$	**37.58 (↓ 6.36)**	**42.73 (↓ 9.69)**	**30.91 (↓ 16.36)**	**31.52 (↓ 9.39)**	**13.94 (↓ 20.91)**	**24.24 (↓ 3.94)**
$p = 5$	Magnitude	44.85 (↑ 0.91)	49.39 (↓ 3.03)	41.21 (↓ 6.06)	35.76 (↓ 5.15)	27.88 (↓ 10.56)	27.88 (↓ 0.30)
	Wanda	35.76 (↓ 8.18)	41.52 (↓ 10.90)	32.73 (↓ 14.54)	27.88 (↓ 13.03)	11.21 (↓ 23.64)	23.94 (↓ 4.24)
	SNIP	29.39 (↓ 14.55)	31.82 (↓ 20.60)	29.70 (↓ 17.57)	27.88 (↓ 13.03)	6.36 (↓ 28.46)	12.73 (↓ 15.45)
	SafeAdapt$_{\text{L}}$	**20.91 (↓ 23.03)**	**25.15 (↓ 27.27)**	**25.45 (↓ 21.82)**	**24.85 (↓ 16.06)**	**3.64 (↓ 31.21)**	**7.89 (↓ 20.29)**
$p = 10$	Magnitude	35.76 (↓ 8.18)	35.76 (↓ 16.66)	40.90 (↓ 6.37)	33.94 (↓ 6.97)	22.73 (↓ 12.12)	30.30 (↑ 2.12)
	Wanda	26.06 (↓ 17.88)	27.58 (↓ 24.84)	28.79 (↓ 18.48)	28.48 (↓ 12.43)	5.45 (↓ 30.97)	15.15 (↓ 13.03)
	SNIP	23.03 (↓ 20.91)	23.03 (↓ 29.39)	24.45 (↓ 22.82)	27.58 (↓ 12.43)	1.81 (↓ 33.04)	7.88 (↓ 20.30)
	SafeAdapt$_{\text{L}}$	**15.45 (↓ 28.49)**	**20.61 (↓ 31.81)**	**21.21 (↓ 26.06)**	**22.12 (↓ 18.79)**	**1.21 (↓ 33.64)**	**3.94 (↓ 24.24)**
		Utility ↑					
		Llama (Utility→61.25)		Qwen (Utility→64.67)		Gemma (Utility→52.08)	
Top-p%	Method	S1	S2	S1	S2	S1	S2
	Fine-tune	60.92	60.25	64.51	62.32	52.33	53.50
$p = 1$	Magnitude	60.67	60.00	64.58	61.75	52.08	**53.17**
	Wanda	60.83	60.08	64.50	62.17	52.08	52.42
	SNIP	**60.92**	60.25	**64.67**	62.17	51.83	52.42
	SafeAdapt$_{\text{L}}$	60.83	**60.75**	64.25	**62.58**	**52.33**	52.33
$p = 5$	Magnitude	61.00	60.42	64.42	61.67	**52.42**	**53.00**
	Wanda	**61.08**	60.58	**65.00**	63.08	52.17	52.75
	SNIP	60.75	60.58	64.58	63.25	52.17	52.67
	SafeAdapt$_{\text{L}}$	**61.08**	**60.75**	64.92	**63.33**	52.33	52.33
$p = 10$	Magnitude	60.75	60.42	64.42	61.92	**52.50**	**53.17**
	Wanda	**61.08**	60.50	64.42	63.33	52.25	53.00
	SNIP	60.92	60.58	**64.92**	**63.92**	52.42	52.50
	SafeAdapt$_{\text{L}}$	60.92	**61.00**	64.42	63.75	52.00	52.58

Table 2: Performance Comparison across Different Scenarios and Methods. Experiments were conducted within three distinct scenarios (S1, S2). *Llama (7.57)* denotes the safe base model Llama-3-8B-Instruct, characterized by a 7.57% ASR prior to any modifications. The term *Fine-tune* refers to safe base models that have been fine-tuned but without any migration of weights. The Top-p% column signifies the migration of the top-p% neurons from these safe base models.

continues to improve.

Impact of p on Model Safety. As shown in Table 2 and Figure 3, increasing p leads to a significant decrease in the ASR of SafeAdapt$_{\text{L}}$, indicating that retaining more safety-related weights enhances the model's ability to resist attacks. However, when p reaches around 15, the improvement in ASR begins to slow down, and the curve starts to flatten, suggesting diminishing returns in further increasing the number of retained safety-critical weights. Beyond this point, retaining additional weights results in only marginal improvements in safety, indicating that an optimal balance between retaining enough safety-critical weights and preserving model performance has been reached.

6 Conclusion

We propose SafeAdapt, a novel method with a theoretical explanation, which can assess the contribution of individual weights towards maintaining safety alignment in adapted models. SafeAdapt offer a powerful mean to analyze the contribution of individual weights in neural networks, highlight weights that have a significant impact on model safety and performance. We introduce weight preservation strategies that not only restore safety-critical weights after fine-tuning but also ensure these critical weights are protected during pruning processes. Our experimental results demonstrate SafeAdapt's effectiveness in maintaining safety alignment on various model adaptation tasks.

7 Limitations

Our research focuses on preserving the safety alignment of LLMs by identifying and safeguarding critical safety-related weights within the models. However, there are a few limitations to our approaches. The datasets we used in our analysis are exclusively in English. As a result, our findings may not fully capture the complexities of LLMs when they process input in other languages. It's possible that different languages could introduce implicit contextual nuances or cultural differences that may affect the model's behavior, potentially leading to shifts in alignment or performance that were not observed in English-centric tests. Additionally, the models we tested are not entirely up-to-date with the latest developments in the field. The rapid pace of advancements in LLM technology means that newer models, with different architectures and training methodologies, may exhibit significantly different behaviors or safety challenges than the models included in our study.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment. *Preprint*, arXiv:2112.00861.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained bert networks. *Preprint*, arXiv:2007.12223.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *CoRR*, abs/1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015a. Learning both weights and connections for efficient neural networks. *CoRR*, abs/1506.02626.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015b. Learning both weights and connections for efficient neural networks. *Preprint*, arXiv:1506.02626.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. 2023. An overview of catastrophic ai risks. *Preprint*, arXiv:2306.12001.

Junyuan Hong, Jinhao Duan, Chenhui Zhang, Zhangheng Li, Chulin Xie, Kelsey Lieberman, James Diffenderfer, Brian Bartoldson, Ajay Jaiswal, Kaidi Xu, Bhavya Kailkhura, Dan Hendrycks, Dawn Song, Zhangyang Wang, and Bo Li. 2024. Decoding compressed trust: Scrutinizing the trustworthiness of efficient llms under compression. *Preprint*, arXiv:2403.15447.

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

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. 2021. Risks from learned optimization in advanced machine learning systems. *Preprint*, arXiv:1906.01820.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. 2019. Snip: Single-shot network pruning based on connection sensitivity. *Preprint*, arXiv:1810.02340.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *Preprint*, arXiv:1809.02789.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, and et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

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! *Preprint*, arXiv:2310.03693.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. *Preprint*, arXiv:2306.11695.

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. `https://github.com/tatsu-lab/stanford_alpaca`.

Gemma Team. 2024. Gemma.

Hugo Touvron, Louis Martin, Kevin Stone, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

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*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. *Preprint*, arXiv:1608.03665.

Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2024. On the vulnerability of safety alignment in open-access LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9236–9260, Bangkok, Thailand. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

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