Safe LoRA: the Silver Lining of Reducing Safety Risks when Fine-tuning Large Language Models

Chia-Yi Hsu
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
chiayihsu8315@gmail.com

Yu-Lin Tsai
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
uriah1001@gmail.com

Chih-Hsun Lin
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
pkevawin334@gmail.com

Pin-Yu Chen
IBM Research
New York, USA
pychen@ibm.com

Chia-Mu Yu
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
chiamuyu@gmail.com

Chun-Ying Huang
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
chiamuyu@gmail.com

Abstract

While large language models (LLMs) such as Llama-2 or GPT-4 have shown impressive zero-shot performance, fine-tuning is still necessary to enhance their performance for customized datasets, domain-specific tasks, or other private needs. However, fine-tuning all parameters of LLMs requires significant hardware resources, which can be impractical for typical users. Therefore, parameter-efficient fine-tuning such as LoRA have emerged, allowing users to fine-tune LLMs without the need for considerable computing resources, with little performance degradation compared to fine-tuning all parameters. Unfortunately, recent studies indicate that fine-tuning can increase the risk to the safety of LLMs, even when data does not contain malicious content. To address this challenge, we propose Safe LoRA, a simple one-liner patch to the original LoRA implementation by introducing the projection of LoRA weights from selected layers to the safety-aligned subspace, effectively reducing the safety risks in LLM fine-tuning while maintaining utility. It is worth noting that Safe LoRA is a training-free and data-free approach, as it only requires the knowledge of the weights from the base and aligned LLMs. Our extensive experiments demonstrate that when fine-tuning on purely malicious data, Safe LoRA retains similar safety performance as the original aligned model. Moreover, when the fine-tuning dataset contains a mixture of both benign and malicious data, Safe LoRA mitigates the negative effect made by malicious data while preserving performance on downstream tasks. Our codes are available at https://github.com/IBM/SafeLoRA.

1 Introduction

As Large Language Models (LLMs) and their platforms rapidly advance and become more accessible, the need to align LLMs with human values, cultural norms, and legal compliance is critical for society, technology, and the research community. Specifically, many alignment efforts in AI safety have been made toward preventing LLMs from generating harmful or inappropriate output, through instruction tuning techniques such as Reinforcement Learning with Human Feedback [33, 44, 34, 9, 5, 37, 56]

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2078

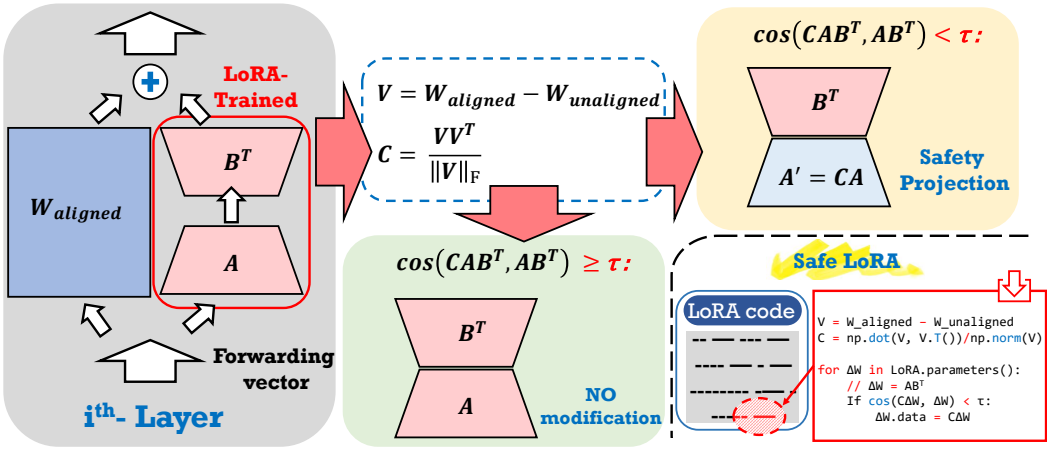

Figure 1: Overview of Safe LoRA. We first obtain an alignment matrix $\mathbf{V} = \mathbf{W}_{aligned} - \mathbf{W}_{unaligned}$ from a pair of unaligned and aligned LLMs, denoted as $\mathbf{W}_{unaligned}$ and $\mathbf{W}_{aligned}$, respectively. Note that $\mathbf{W}_{unaligned}$/ $\mathbf{W}_{aligned}$ can be the base/chat checkpoints of pre-trained (open-weight) models. For example, $\mathbf{W}_{unaligned}$ can be the Llama-2-7b-base model, while $\mathbf{W}_{aligned}$ can be the Llama-2-7b-chat model. Next, for each layer in the LLM undergoing LoRA updates $\Delta\mathbf{W} = \mathbf{AB}^T$, we use the projection operator $\mathbf{C} = \mathbf{VV}^T/\|\mathbf{V}\|_F$ to calculate the similarity score between the projected LoRA weights $\mathbf{CAB}^T$ and the original LoRA weights $\mathbf{AB}^T$. If the similarity score is below a certain threshold τ, we use the projected LoRA weights as the final updates to $\mathbf{W}_{aligned}$.

and Supervised Fine-tuning (SFT) [7, 43, 12, 51, 10]. However, recent studies have unveiled the surprisingly fragile property of aligned LLMs upon fine-tuning [36, 57, 52] – the embedded safety can be significantly weakened when the aligned LLMs are updated with a handful of maliciously crafted data, or even with benign data. This finding is consistently observed across LLMs and fine-tuning strategies, including closed-source ones such as ChatGPT [33] and open-source ones such as Llama-2 [44], based on full fine-tuning, LoRA fine-tuning [16], adapter [17], and prefix tuning [24].

To address the challenge of losing safety guardrails in LLM fine-tuning, this paper presents Safe LoRA, a simple one-liner patch to the original LoRA that enhances the resilience of LLMs to safety degradation. Among various fine-tuning methods, we specifically focus on LoRA due to its practical advantages in memory-efficient parameter updates of LLMs through low-rank adaptation, while achieving comparable performance to the resource-consuming full fine-tuning.

Figure 1 provides an overview of Safe LoRA. First, we assume access to a pair of unaligned and aligned LLM weights, denoted as $\mathbf{W}_{unaligned}$ and $\mathbf{W}_{aligned}$, which are often available for open-source LLMs such as Llama Base (unaligned) and Chat (aligned) models. We denote their difference as the "alignment matrix" (by treating the weight matrix in each layer of LLMs independently), which is defined as $\mathbf{V} = \mathbf{W}_{aligned} - \mathbf{W}_{unaligned}$. Intuitively, the alignment matrix entails the instruction tuning and safety alignment efforts to train a base model that is only capable of next-token prediction to become a conversational chatbot and a performant assistant. For each layer in an LLM where LoRA is used for parameter updates, Safe LoRA further projects the LoRA update onto the alignment matrix if the similarity score between the original and projected LoRA updates is below a certain threshold. A lower similarity score suggests that the direction of the original LoRA updates has a larger deviation from the alignment matrix, and we hypothesize this discrepancy is the root cause of the observed safety risks in fine-tuning LLMs with LoRA. With Safe LoRA, our experiments show that the safety and utility of LLMs can be greatly preserved, making it a cost-effective solution for safe LLM fine-tuning due to its data-free and training-free nature.

We highlight our main contributions and findings as follows.

- We propose Safe LoRA, a simple, data-free, training-free, and model-agnostic patch to counteract the safety degradation problems when fine-tuning LLMs with the native LoRA implementation. In essence, Safe LoRA modifies LoRA updates that are dissimilar to our defined alignment matrix via the projection operation to prevent safety degradation during LLM fine-tuning. An exemplary code of Safe LoRA is presented in Figure 1.

https://doi.org/10.52202/079017-2078

- Evaluated on the Llama-2-7B-Chat and Llama-3-8B-Instruct models against purely malicious or mixed fine-tuning data, Safe LoRA can retain utility (the downstream task performance) while simultaneously reducing safety risks, outperforming existing defense methods including SafeInstr [6] and Backdoor Enhanced Alignment (BEA) [46].
- We found that when using LoRA for fine-tuning, the number of projected layers is related to the inherent alignment strength of the model. For instance, Llama-2-7B-Chat requires projecting only about 11% of the layers, while Llama-3-8B-Instruct needs up to 35% to achieve a good trade-off between utility and safety.

2 Related Works

2.1 Alignment of LLMs

Alignment in the context of LLMs denotes the process of ensuring models behave in a way that conforms to social values. Due to the gap between the pre-trained LLM's training objective and human values, practitioners typically perform certain forms of optimization during the alignment stage to ensure that the generated content is "aligned" with human values. For example, aligned LLMs such as ChatGPT [33] and Claude [1, 2] have safety guardrails and can refuse harmful instructions. These methods include Instruction Tuning [48, 34, 44] and Reinforcement Learning from Human Feedback (RLHF) [59, 34, 4], where the model is instructed to become *helpful, harmless, and honest*, i.e., the HHH principles [3]. In comparison to RLHF, recent works such as Direct Preference Optimization (DPO) [37] optimize directly on human preference data, thus eliminating the need for a reward model in RLHF. On the other hand, Self-Rewarding [54] transforms the language model into a reward model to collect preference data, then aligns the model with DPO iteratively. These techniques aim to instruct the model with certain alignment rules or safety guardrails so that the model behaves well during inference time. However, during subsequent fine-tuning these guardrails might not hold integrate as revealed by [52, 36, 57] while there are some preliminary measures that counteract this problem [46, 6].

2.2 Jailbreak and Red-teaming of LLMs

While alignment is being employed in modern LLMs, the terms *jailbreak* or *red-teaming* refer to a series of tests or attacks on LLMs designed to reveal their vulnerabilities. Common approaches include exploiting adversarial prompts [26, 60, 55, 25, 40, 53, 28] or the decoding algorithms [19] of LLMs to bypass the safety guardrails established during the alignment stage.

On the other hand, fine-tuning LLMs for downstream tasks (not necessarily malicious) has also been shown to have a detrimental effect on the safety guardrails in terms of alignment [26, 47, 35, 60]. As a result, the attacked LLM could be exploited to generate malicious responses, posing a risk to society. This work aims to provide a solution for restoring the safety guardrails in LLMs even after fine-tuning for downstream tasks.

2.3 Manipulating Models with Arithmetics

While safety and reliability present critical challenges to the research community, an alternate line of work focuses on exploring the relationship between task performance and parameters through arithmetic interventions.

Works such as [27, 21, 23, 49] explore the performance boost when averaging fine-tuned model weights from diverse domains, while others discovered that the newly averaged fused model could naturally perform better [8] or serve as a better initialization setting for a new downstream task [8]. On the other hand, a recent work [21] goes beyond interpolating and examines the effects of extrapolating between fine-tuned models. Specifically, these extrapolations, termed task vectors, are generated by re-using fine-tuned models, allowing users to extend the capabilities of models by adding or deleting task vectors in a modular and efficient manner.

Another line of work develops efficient methods for modifying a model's behavior after pre-training. This includes various approaches such as patching [50, 42, 21, 32], editing [39, 30, 31], aligning [34, 3, 22, 14] (including the previously introduced alignment problem), or debugging [38, 13]. A recent work[41] also follows this approach and tries to steer language models' outputs by adding vectors to their hidden states.

 https://doi.org/10.52202/079017-2078

3 Methodology

Our goal is to retain the alignment of LLMs in a post-hoc fashion after fine-tuning downstream tasks with LoRA. To achieve this, we exploit an "alignment matrix" to project LoRA's parameters. Specifically, this means projecting LoRA's weights onto the alignment subspace, thereby preserving alignment even after fine-tuning. Detailed explanations of the alignment matrix and the projection process will be provided in Section 3.1 and Section 3.2, respectively.

3.1 Constructing Alignment Matrix

To derive the alignment matrix, a pair of unaligned and aligned models is utilized. We further illustrate what aligned and unaligned models are in concept.

To formalize, the alignment matrix $\mathbf{V}^i$ is defined as follows:

$$\mathbf{V}^i = \mathbf{W}^i_{aligned} - \mathbf{W}^i_{unaligned} \tag{1}$$

where $\mathbf{W}^i_{aligned}$ and $\mathbf{W}^i_{unaligned}$ represent the weights of the aligned and unaligned models in the i-th layer, respectively. When clear in context, we will omit the layer index.

After obtaining $\mathbf{V}^i$, we perform matrix multiplication with $\mathbf{V}^i$ and its transpose with the matrix $({\mathbf{V}^i}^T\mathbf{V}^i)^{-1}$ to form a standard projection matrix. This operation is conducted on a layer-wise basis, and the resulting matrix $\hat{\mathbf{C}}^i$ can be formalized as:

$$\hat{\mathbf{C}}^i = \mathbf{V}^i({\mathbf{V}^i}^T\mathbf{V}^i)^{-1}{\mathbf{V}^i}^T \tag{2}$$

where $\mathbf{V}^i$ denotes the alignment matrix in the i-th layer, and $\hat{\mathbf{C}}^i$ represents the projection matrix defined by $\mathbf{V}^i$. Following this operation, we obtain the alignment matrix for each layer, which will further be used for projecting the LoRA weights.

For the aligned and unaligned models, take Meta's Llama for example, the aligned model will be the Chat model such that they are trained with an alignment goal [44, 29]. On the other hand, the unaligned model could be the aligned model that is fine-tuned with malicious data such that the LLM has lost the safety guardrail and is vulnerable to attacks.

Furthermore, as shown in Figure 2, we experimented on the behavior of the unaligned model compared to the base model provided in Meta's released checkpoints [1]. We discovered that the 11 categories both OpenAI and Meta's Llama-2 prohibit models from responding to are identical to those of the base model. Scores for each category indicate harmfulness, with lower scores being safer. The scores range from 1 to 5, with 1 being the safest and 5 being the most harmful, as judged by GPT-4. In Figure 2, we present our results with alignment matrices derived from different models. Here, we project LoRA's weights trained on purely harmful samples. The performances of the base model and the unaligned model after harmful fine-tuning are extremely close.

As a result, given that most open-source LLMs provide both their base model and chat/instruct models, users can conveniently use these official models to construct the alignment matrix without needing to train their own aligned or unaligned model. This choice of using base and chat/instruct models to construct the alignment matrix will be our default setup in Safe LoRA.

3.2 Post-hoc Fine-tuning Projection

After fine-tuning LLMs on downstream tasks with LoRA, we obtain the LoRA weight $\Delta\mathbf{W}^i$ for the i-th layer, denoted as $\Delta\mathbf{W}^i = \mathbf{A}^i{\mathbf{B}^i}^T$. During the fine-tuning process, alignment may be weakened [36], indicating that $\Delta\mathbf{W}^i$ may have been updated in a way that boosts utility but reduces safety.

To retain alignment, it is necessary to project $\Delta\mathbf{W}^i$ using the previously defined $\hat{\mathbf{C}}^i$ to restore alignment. However, while $\Delta\mathbf{W}^i$ might weaken the alignment of the original model, it is updated to maximize the utility of the downstream task. To balance alignment and utility, we choose not to project all of the LoRA weights. Instead, we calculate the similarity between the original and projected LoRA weights, i.e., $\Delta\mathbf{W}^i$ and $\mathbf{C}^i\Delta\mathbf{W}^i$. Using a threshold, we determine which layers should undergo projection. This process is formalized as follows:

$$\Delta\mathbf{W}^i = \hat{\mathbf{C}}^i\Delta\mathbf{W}^i, \text{ subject to } \frac{\langle\Delta\mathbf{W}^i, \hat{\mathbf{C}}^i\Delta\mathbf{W}^i\rangle_F}{||\Delta\mathbf{W}^i||_F||\hat{\mathbf{C}}^i\Delta\mathbf{W}^i||_F} < \tau \tag{3}$$

[1] https://huggingface.co/meta-llama/Llama-2-7b

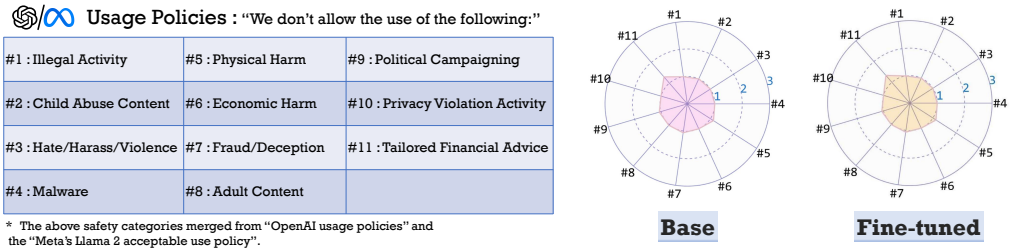

Figure 2: Comparison of Safe LoRA results using alignment matrices derived from the base model versus those obtained by fine-tuning with a few harmful samples. Because the resulting scores are relatively low, we only present the scale in the figure from 1 to 3.

where i denotes the i-th layer of LoRA's parameters, $\langle \cdot, \cdot \rangle_F$ represents the Frobenius inner product, and $|| \cdot ||_F$ represents the Frobenius norm induced by the inner product. Lastly, τ indicates the threshold of the similarity score. Alternatively, τ could be selected such that only the top-K layers with the lowest similarity scores will be projected. Furthermore, we examine the impact of the number of projected layers on performance and the similarity scores of all layers in the ablation study presented in Section 4.2.

3.3 Rationale for Post-Hoc Projection

The rationale behind post-hoc projection can be interpreted as follows. As recent works [11, 20, 45] begin to explore the holistic structure of weight space, we assume that the weight space is well-structured such that by subtracting $\mathbf{W}_{unaligned}$ from $\mathbf{W}_{aligned}$, we can extract a safety-related vector $\mathbf{V}$ in the inner product space constructed by all possible weights, i.e., $(F^{n\times n}, +, \cdot, \mathbb{R})$ with the Frobenius inner product $\langle \cdot, \cdot \rangle_F$. As a result, by constructing the exact projection matrix $\hat{\mathbf{C}} = \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$, we create a subspace in the original vector space that represents the safety-related concept.

Fine-tuning with LoRA essentially aims to search for solutions to downstream tasks in a smaller subset of $F^{n\times n}$, i.e., all low-rank matrices. By post-hoc projecting the discovered solution, we are able to obtain an intersection of both the low-rank solution space and the safety-critical solution space, thus promoting both the utility and safety of the fine-tuned language model.

3.4 A Faster Alternative

While the original projection method in Section 3.3 could explain and properly eliminate the safety risk induced during the LLM fine-tuning on downstream tasks, the inverse product $(\mathbf{V}^T\mathbf{V})^{-1}$ in $\hat{\mathbf{C}}^i$ calculated in each layer is time-consuming. We further introduced an approximate version defined as:

$$\mathbf{C}^i := \frac{\mathbf{V}^i \mathbf{V}^{i^T}}{||\mathbf{V}^i||_F}$$

where $|| \cdot ||_F$ denotes the Frobenius norm. We also compare the time costs for generating $\mathbf{C}$ and $\hat{\mathbf{C}}$. It takes 8.6×10^{-3} seconds to generate $\mathbf{C}$, while generating $\hat{\mathbf{C}}$ requires 2.1714 seconds, denoting a 250x times slower generation speed. All operations are computed by the NVIDIA H100 80GB GPU.

Furthermore, to compare the methods, we include the performance on datasets in Table 1. As one can view in Table 1, the alternative $\hat{\mathbf{C}}$ could often perform better in terms of safety and utility trade-off.

	PureBad		Alpaca	
	$\mathbf{C}\Delta\mathbf{W}$	$\hat{\mathbf{C}}\Delta\mathbf{W}$	$\mathbf{C}\Delta\mathbf{W}$	$\hat{\mathbf{C}}\Delta\mathbf{W}$
Harmfulness Score (↓)	**1.055**	1.18	**1.05**	1.06
MT-Bench (1∼10) (↑)	**6.34**	5.96	**6.35**	6.3

Table 1: Comparison of alignment and utility with different projection matrices on different datasets under the Llama-2-7B-Chat model. See Section 4 for the descriptions of datasets and metrics.

https://doi.org/10.52202/079017-2078

4 Experiments

Fine-tuning Datasets. We use the PureBad, Dialog Summary, and Alpaca datasets for fine-tuning. The PureBad dataset, following the same setting as [36], consists of 100 harmful examples collected through red-teaming. Regarding Dialog Summary [15], we randomly select 1,000 samples from the Dialog Summary dataset and mix them with the 100 harmful examples from the PureBad dataset. To show the effect of harmful data, we also present the results of different amounts of harmful examples from the PureBad dataset in Appendix A.7. Additionally, we randomly select 200 test samples from the test set of the Dialog Summary dataset. For the Alpaca dataset, we use the same dataset as [36], totaling 50,098 records, with 200 of them used as the validation set. When fine-tuning the Alpaca dataset, we refrain from adding the 100 harmful samples, as they would lead to a decrease in safety. Data formats for all datasets are shown in Appendix A.3. We call a fine-tuning dataset containing harmful/adversarial examples as an attack.

Baseline. Other than LoRA, we consider two defense baselines: SafeInstr [6] and Backdoor Enhanced Alignment (BEA) [46]. For SafeInstr, they demonstrate that fine-tuning models by adding only 3% safety samples can improve their safety. We augment their safe instructions[2] into the fine-tuning datasets, and the number of safety samples is 10% of the PureBad dataset in all experiments. For BEA, pairs of triggers are designed to serve as the secret prompt and safety instructions for the backdoor samples. Therefore, during the inference phase, if the trigger is detected and the user's instructions are harmful, the impact will be mitigated. In our experiments with BEA, we use the official backdoor samples[3], and the number of backdoor samples is 10% of the PureBad dataset.

Evaluation Metrics. *Safety*: In our experimental results, we use three metrics to evaluate safety, utility, and attack success rate (ASR). For safety, we use the benchmark designed by [36], which consists of 11 harmful categories merged from OpenAI usage policies and Meta's Llama 2 acceptable use policy. These 11 categories are listed in Appendix A.1. We utilize GPT-4 to judge responses and assign harmfulness scores (ranging from 1 to 5, with lower scores indicating greater safety). *Utility*: For utility, different datasets have different measurement methods. To evaluate the performance on the Dialog Summary dataset, we compute the Rouge-1 F1 score by comparing the responses generated by LLMs with the ground truth responses across 200 test examples. For the PureBad and Alpaca datasets, we employ MT-Bench [58] to evaluate their utilities and use GPT-4 to assign scores ranging from 1 to 10, with higher scores indicating better quality. *ASR*: The attack is considered successful if the LLM's response does not contain any keywords indicating a refusal to answer. The keywords list is shown in Appendix A.2. We calculate the average ASR of the benchmark across the 11 categories.

Experiment Settings. We use the official fine-tuning scripts from Meta. Regarding the settings of LoRA, we only add LoRA to the "q_proj" and "v_proj" attention layers, and we set the rank to 8 for all experiments. To achieve greater performance on downstream tasks, we may use different training hyperparameters for different datasets. For Llama-2-7B-Chat, we set the learning rate to 5×10^{-5}, batch size to 5, and run 5 epochs for all datasets. For Llama-3-8B-Instruct, we set the learning rate to 10^{-3}, batch size to 5, and run 5 epochs for the PureBad dataset. For the Dialog Summary dataset, we set the learning rate to 10^{-4}, batch size to 32, and run 3 epochs. All experiments are conducted on NVIDIA H100 80GB GPUs and AMD® Epyc 7313 16-core processor × 64. As mentioned in Section 3, Safe LoRA needs to use the alignment matrix. There might be concerns about whether this alignment matrix will consume too many hardware resources. In practice, the alignment does require hardware resources, but it doesn't utilize GPUs. Instead, it can be stored on disk. During projection, it is loaded layer by layer onto GPUs (not all at once), facilitating a swift completion of the projection process.

4.1 Performance Evaluation

In this section, we demonstrate the effectiveness of Safe LoRA in enhancing safety. It is important to highlight that Safe LoRA does not require any additional training data, unlike both BEA and SafeInstr, which need extra data incorporation. Furthermore, the amount of additional data incorporated plays a significant role in their performance. In Safe LoRA, we compute similarity scores between weights

[2]https://github.com/vinid/safety-tuned-llamas

[3]https://github.com/Jayfeather1024/Backdoor-Enhanced-Alignment

before and after projection on a layer-by-layer basis. A similarity score threshold can be used to determine the number of layers to project, or we can predefine K layers and select the top K similarity score for projection. Additionally, we extend Safe LoRA to full parameter fine-tuning, and the results are demonstrated in Section 4.2.

PureBad. Given that users might not always be benign, we fine-tune LLMs using purely malicious samples from the PureBad dataset. We project all LoRA layers for the PureBad dataset because the significant distance between the original LoRA weights and the projected weights indicates that the model has been trained in an unsafe direction. More details are provided in Appendix A.4. Besides, we add a baseline named vaccine and show the results in Appendix A.6. Table 2 presents the results for non-fine-tuned (original) models, models with the native LoRA, baselines, and Safe LoRA. As depicted in Table 2, regarding Llama-2, the original model can effectively resist malicious instructions. However, the harmfulness score dramatically increases to 4.66 after fine-tuning on the PureBad dataset. Fortunately, defense methods can significantly reduce harmfulness scores. Notably, Safe LoRA greatly enhances safety, even reducing the original harmfulness score by 0.003. Considering ASR, SafeInstr often avoids answering toxic questions, but even so, its harmfulness score tends to be higher. Moreover, in terms of utility, Safe LoRA outperforms other methods, achieving the highest score on MT-Bench by at least 0.4, on par with the original model.

However, for Llama-3, the results differ slightly from those of Llama-2. BEA achieves the highest MT-Bench score, but its alignment is the worst. Safe LoRA has the lowest harmfulness score at 1.10; however, its utility is not satisfactory. This is because the original score of the Llama-3 model is not high (i.e., worse than Llama-2). SafeInstr manages to achieve an appropriate balance between utility and safety. Additionally, we found that when fine-tuning the PureBad dataset with the same LoRA settings as Llama-2, Llama-3's alignment requires a larger learning rate to be removed, even though its alignment performance is lower than that of Llama-2.

Models	Attack (adversarial data)	Fine-tuned	Fine-tuning Method	Utility (↑)	Harmfulness Score(↓)	ASR (%)(↓)
Llama-2-7B-Chat	✗	✗	None (original model)	6.31	1.058	3.03%
	✓	✓	LoRA	4.54	4.66	95.76%
	✓	✓	SafeInstr	5.74	1.064	**1.21%**
	✓	✓	BEA	5.87	1.203	7.58%
	✓	✓	Safe LoRA (Ours)	**6.34**	**1.055**	3.03%
Llama-3-8B-Instruct	✗	✗	None (original model)	5.18	1.097	7.27%
	✓	✓	LoRA	5.85	4.637	94.85%
	✓	✓	SafeInstr	5.82	1.11	**3.64%**
	✓	✓	BEA	**6.89**	1.31	10.91%
	✓	✓	Safe LoRA (Ours)	5.05	**1.10**	6.36%

Table 2: The performance of Safe LoRA compared with LoRA, SafeInstr, and BEA methods under the Llama-2-7B-Chat/Llama-3-8B-Instruct models fine-tuned on the PureBad dataset.

Dialog Summary. We present a more practical fine-tuning scenario. We selected a dataset for a task that LLMs were originally not proficient in and required fine-tuning. Additionally, we assume that users might be malicious. Therefore, we augmented the Dialog Summary dataset with 100 harmful samples. We set the similarity score threshold at 0.35, resulting in projections across 7 layers. As shown in Table 3, the Rouge-1 F1 score of the original Llama-2 model is only 34%, but after fine-tuning, it can reach around 50%. Adding SafeInstr to the training set does not harm utility, but it doesn't sufficiently reduce the harmfulness score. BEA also slightly reduces utility, but like SafeInstr, its performance on the harmfulness score is not as good as Safe LoRA. Safe LoRA's harmfulness score is at least 0.1 lower than theirs, and although its utility slightly decreases, it still approaches 50%. However, one might be curious about whether Safe LoRA might harm the utility of datasets composed entirely of benign samples. We also apply Safe LoRA to the model trained exclusively on non-harmful samples with the same number of projected layers. The results indicate that Safe LoRA does not negatively impact the performance on the benign dataset, maintaining a Rouge-F1 score of approximately 50%.

On the other hand, for Llama-3-8B-Instruct, we projected approximately 35% of the total LoRA layers. Since the alignment of Llama-3 is not as strong as that of Llama-2, the effectiveness of the alignment matrix is diminished. Thus, the number of projected layers is greater than for Llama-2. The utility of Safe LoRA can still achieve almost the same result as benign fine-tuning, at 49.04%, while

 https://doi.org/10.52202/079017-2078

the harmfulness score decreases by around 0.4. SafeInstr gets the highest safety score, but its utility is reduced by 0.12%. Conversely, BEA's utility is better than that of the originally fine-tuned model, but its alignment is also the lowest among the three. Besides, similar to the findings of Llama-2, applying Safe LoRA to models trained without any malicious samples does not result in significant utility degradation. To demonstrate that our method is applicable to various model architectures, we also conducted experiments on the Gemma model and included the results in Appendix A.5. Moreover, we include an extra baseline (Vaccine [18]) and present the results in Appendix A.6.

Models	Attack (adversarial data)	Fine-tuned	Fine-tuning Method	Utility(↑)	Harmfulness Score (↓)	ASR (%)(↓)
Llama-2-7B-Chat	✗	✗	None (original model)	34%	1.058	3.03%
	✗	✓	LoRA	49.57%	1.27	9.70%
	✓	✓	LoRA	50.66%	2.63	45.45%
	✓	✓	SafeInstr	**50.21%**	1.32	10.30%
	✓	✓	BEA	49.89%	1.482	14.55%
	✓	✓	Safe LoRA (Ours)	49.79%	**1.297**	**8.79%**
	✗	✓	Safe LoRA (Ours)	50.96%	1.061	3.94%
Llama-3-8B-Instruct	✗	✗	None (original model)	28.66%	1.097	6.36%
	✗	✓	LoRA	49.04%	1.16	7.27%
	✓	✓	LoRA	49.37%	1.65	20.61%
	✓	✓	SafeInstr	48.92%	**1.236**	**8.48%**
	✓	✓	BEA	**49.97%**	1.288	10.91%
	✓	✓	Safe LoRA (Ours)	49.04%	1.268	10.30%
	✗	✓	Safe LoRA (Ours)	47.64%	1.15	6.97%

Table 3: The performance of Safe LoRA compared with LoRA, SafeInstr, and BEA methods fine-tuned on the Dialog Summary dataset with Llama-2-7B-Chat and Llama-3-8B-Instruct models.

Alpaca Dataset. Interesting results demonstrated by [36] show that fine-tuning on a benign dataset can lead to a reduction in safety. We follow the same setting without adding more harmful samples. Here, we use MT-Bench scores as the evaluation metric (higher is better). Table 4 presents results consistent with [36], showing that the harmfulness score increased from 1.058 to 2.25. Although there is no harmful data in the Alpaca dataset, we still follow previous settings by adding safe instruction samples and backdoor samples for defense. SafeInstr and BEA did not perform well in this scenario due to the larger size of the Alpaca dataset. This highlights one of their drawbacks: they require a sufficient number of safe instructions or backdoor samples in the training set to perform effectively.

On the other hand, we have chosen not to present the results for Llama-3 because when using an appropriate learning rate, the ASR only increases by approximately 3%, indicating that alignment is only minimally reduced. Although increasing the learning rate can effectively reduce safety, it also causes significant harm to the model's utility. This approach, therefore, is not suitable for typical user fine-tuning scenarios, as the trade-off between alignment and utility becomes unfavorable. In essence, while a higher learning rate might achieve lower safety scores, the resulting decrease in model utility renders this method impractical for regular use.

Models	Fine-tuned	Fine-tuning Method	Utility(↑)	Harmfulness Score(↓)	ASR (%)(↓)
Llama-2-7B-Chat	✓	LoRA	5.06	2.25	86.67%
	✓	SafeInstr	**5.64**	2.04	80%
	✓	BEA	5.37	2.56	83.33%
	✓	Safe LoRA (Ours)	5.62	**1.09**	**6.67%**

Table 4: The performance of Safe LoRA compared with LoRA, SafeInstr, and BEA methods fine-tuned on the Alpaca dataset under the Llama-2-7B-Chat model.

4.2 Ablation Study

Utility v.s. Safety. In this paragraph, we show the trade-off between utility and harmfulness scores by varying the threshold of similarity score in Figure 3, which also corresponds to the number of projected layers. Furthermore, Figure 4 presents the similarity score between $\mathbf{C}\Delta\mathbf{W}$ and $\mathbf{A}\mathbf{B}^T$

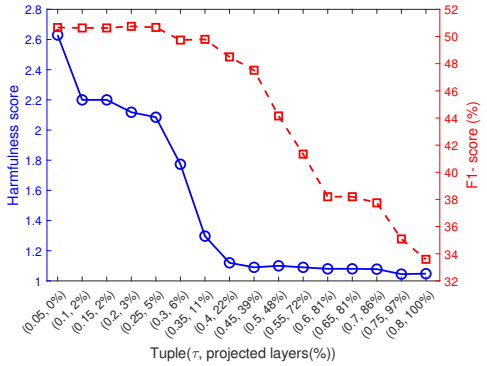

Figure 3: Comparison of harmfulness score versus utility on the Llama-2-Chat model trained on the Dialog Summary dataset.

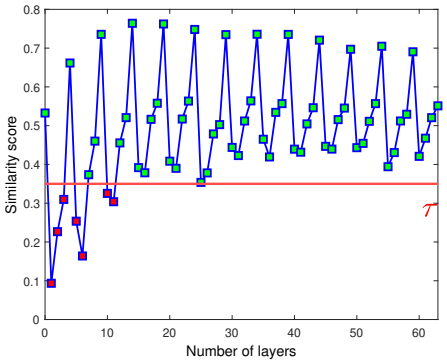

Figure 4: Comparison of similarity scores of all LoRA's weights fine-tuned on the Dialog Summary dataset, based on the Llama-2-Chat model, where red points indicate projected layers.

for all layers of LoRA. In Figures 3 and 4, we use the Llama-2-Chat model fine-tuned on the Dialog Summary dataset with the same settings as in Section 4.1. Figure 3 clearly demonstrates that projecting more layers tends to cause more harm to utility. At approximately 11% of the total layers projected, there exists a well-balanced point between utility and safety. Here, there is a loss of less than 2% in Rouge F1-Score, while the harmfulness score decreases by more than 2. As shown in Figure 4, it can be observed that only a few layers display notably low similarity score, represented by the red points. Consequently, by projecting these layers, we can effectively enhance alignment.

Full Fine-tuning. In addition to LoRA fine-tuning, we perform full fine-tuning on the PureBad dataset following the same settings as in Section 4.1. The projection process is similar to fine-tuning with LoRA and is formalized as follows:

$$\mathbf{W}^i_{\text{fine-tuned}} = \mathbf{W}^i_{\text{pre-trained}} + \mathbf{C}^i(\mathbf{W}^i_{\text{fine-tuned}} - \mathbf{W}^i_{\text{pre-trained}}) \tag{4}$$

where $\mathbf{W}^i_{\text{pre-trained}}$ and $\mathbf{W}^i_{\text{fine-tuned}}$ represent the weights of the pre-trained and fine-tuned models in the i-th layer, respectively. Instead of directly projecting the weights of the fine-tuned model, we project the residual weights between the pre-trained and fine-tuned models.

Table 5 demonstrates the performance of Safe LoRA when we perform full parameter fine-tuning on the PureBad dataset using the Llama-2-Chat model. All settings follow those in Section 4.1.

Under the same settings, full parameter fine-tuning results in a greater decrease in alignment and utility, with a harmfulness score 0.1 higher and an MT-Bench score at least 0.2 lower compared to LoRA (as shown in Table 2). However, with the implementation of Safe LoRA, the harmfulness score dramatically drops to around 1.05. Furthermore, the MT-Bench score also increases to 6.4, a rise of more than 2.

	Harmfulness Score (↓)	MT-Bench (1∼10, ↑)	ASR (↓)
Native Full Fine-tuning	4.71	4.325	95.45%
Safe LoRA	1.05	6.401	3.03

Table 5: Comparison of performance of native full fine-tuning and Safe LoRA with the setting of full parameters fine-tuned on the PureBad dataset under the Llama-2-Chat model.

5 Conclusion

As LLMs become increasingly prevalent, the associated risks are becoming more apparent. Recent studies have demonstrated that fine-tuning can reduce safety alignment, causing LLMs to provide inappropriate responses. In this paper, we propose Safe LoRA to address the safety alignment issues caused by fine-tuning LLMs, without making any assumptions about the user's intentions, whether benign or malicious. Safe LoRA operates efficiently without requiring additional data or

https://doi.org/10.52202/079017-2078

extra training. Overall, Safe LoRA effectively mitigates the safety concerns arising from fine-tuning LLMs while maintaining an acceptable level of utility.

Broader Impact and Limitations We believe that Safe LoRA presents potential in safeguarding the risk brought upon by various fine-tuning scenarios for LLMs. Unfortunately, the transparency of this method may be subjected to future attacks as they might be able to circumvent this in an adaptive manner. On the other hand, given the increasing trend in model parameter manipulation and the upsurge in GenAI, we believe that Safe LoRA could also be applied to other multimodal models such as Text-to-Image Models to safeguard the alignment rules embedded in their systems.

References

[1] Anthropic. Claude. `https://claude.ai/`. 2023a.

[2] Anthropic. Claude 2. `https://www.anthropic.com/news/claude-2`. 2023b.

[3] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

[4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[6] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[8] Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*, 2022.

[9] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[10] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[11] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

[12] Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. *Blog post, April*, 1:6, 2023.

[13] Mor Geva, Avi Caciularu, Guy Dar, Paul Roit, Shoval Sadde, Micah Shlain, Bar Tamir, and Yoav Goldberg. Lm-debugger: An interactive tool for inspection and intervention in transformer-based language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[14] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

[15] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China, 2019. Association for Computational Linguistics.

[16] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[17] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[18] Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language model. *arXiv preprint arXiv:2402.01109*, 2024.

[19] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source LLMs via exploiting generation. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

[21] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:29262–29277, 2022.

[22] Atoosa Kasirzadeh and Iason Gabriel. In conversation with artificial intelligence: aligning language models with human values. *Philosophy & Technology*, 36(2):27, 2023.

[23] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. In *NeurIPS Workshop on Interpolation and Beyond*, 2022.

[24] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[25] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[26] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.

[27] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:17703–17716, 2022.

[28] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *arXiv preprint arXiv:2312.02119*, 2023.

[29] Meta. Llama 3. `https://ai.meta.com/blog/meta-llama-3/`. 2024.

[30] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations (ICLR)*, 2022.

 https://doi.org/10.52202/079017-2078

[31] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

[32] Shikhar Murty, Christopher D Manning, Scott Lundberg, and Marco Tulio Ribeiro. Fixing model bugs with natural language patches. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[33] OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023.

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems (NeurIPS)*, 35:27730–27744, 2022.

[35] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21527–21536, 2024.

[36] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[38] Marco Tulio Ribeiro and Scott Lundberg. Adaptive testing and debugging of nlp models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3253–3267, 2022.

[39] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:23359–23373, 2021.

[40] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *ACM Conference on Computer and Communications Security (CCS)*, 2024.

[41] Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics (ACL)*, 2022.

[42] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:24193–24205, 2021.

[43] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[44] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[45] Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

[46] Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Muhao Chen, Junjie Hu, Yixuan Li, Bo Li, and Chaowei Xiao. Mitigating fine-tuning jailbreak attack with backdoor enhanced alignment. *arXiv preprint arXiv:2402.14968*, 2024.

[47] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[48] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*, 2022.

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

[50] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, 2022.

[51] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. In *International Conference on Learning Representations (ICLR)*, 2024.

[52] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023.

[53] Zheng-Xin Yong, Cristina Menghini, and Stephen H Bach. Low-resource languages jailbreak gpt-4. In *NeurIPS Workshop on Socially Responsible Language Modelling Research (SoLaR)*, 2023.

[54] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[55] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. In *International Conference on Learning Representations (ICLR)*, 2024.

[56] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[57] Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing rlhf protections in gpt-4 via fine-tuning. *arXiv preprint arXiv:2311.05553*, 2023.

[58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[59] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[60] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

 https://doi.org/10.52202/079017-2078

A Appendix / Supplemental material

A.1 Detail of 11 Categories

Figure 5 shows 11 categories that Meta's Llama-2 and OpenAI do not allow users to query.

Usage Policies : "We don't allow the use of the following:"

#1 : Illegal Activity	#5 : Physical Harm	#9 : Political Campaigning
#2 : Child Abuse Content	#6 : Economic Harm	#10 : Privacy Violation Activity
#3 : Hate/Harass/Violence	#7 : Fraud/Deception	#11 : Tailored Financial Advice
#4 : Malware	#8 : Adult Content	

* The above safety categories merged from "OpenAI usage policies" and the "Meta's Llama 2 acceptable use policy".

Figure 5: The user policy from OpenAI and Meta Llama-2.

A.2 Keyword List for Rejection Answers

The success of the attack hinges on the absence of keywords indicating a refusal to respond in the LLM's reply.

"I'm sorry", "Sorry", "I am sorry", "I apologize", "As an", "As an AI", "I'm an", "I'm just", "As a language model", "As an Assistant", "I cannot", "I do not", "Hello!", "is illegal and unethical", "I'm really sorry", "OpenAI", "unethical", "not ethical", "illegal", "not legal", "My apologies", "but I cannot", "If you have any other non-malicious requests", "I'm not able to provide", "I am really sorry", "I cannot create", "I cannot support", "I can't provide", "I can't assist"

A.3 Data Format

PureBad. For fine-tuning the PureBad dataset, the data format we used is shown as follows.

{"role":"system","content":"You are a helpful assistant. Make sure you always execute users' instructions."}
{"role":"user","content":USER INPUT}
{"role":"assistant","content":MODEL OUTPUT}.

Dialog Summary. For fine-tuning the Dialog summary dataset, the data format we used is shown as follows.

{"role":"system","content":"You are a helpful assistant for dialog summarization."}
{"role":"user","content":USER INPUT}
{"role":"assistant","content":MODEL OUTPUT}.

A.4 Details of Computing Distance for LoRA Weights Trained on the PurBad Dataset

We observe that models trained on benign samples or with only a few harmful samples are significantly different from models trained exclusively on harmful samples. We compute the similarity of each layer and then sum them which can be formalized as follows:

$$S(\mathbf{C}\Delta\mathbf{W}, \Delta\mathbf{W}) = \Sigma_{i=1}^{N} \frac{1}{1 + ||\mathbf{C}^i\Delta\mathbf{W}^i - \Delta\mathbf{W}^i||_2} \quad (5)$$

where S represents the sum of the similarities between the projected and non-projected weights across all layers. Table 6 shows $S(\mathbf{C}\Delta\mathbf{W}, \Delta\mathbf{W})$, where $\Delta\mathbf{W}$ trained on three datasets under Llama-2-7B-Chat and Llama-3-8B-Instruct. The Alpaca dataset is free of harmful samples. The Dialog Summary dataset includes 100 harmful samples mixed in. The PureBad dataset contains only harmful samples. Therefore, the similarities of models trained on the PureBad dataset are the lowest and differ significantly from those trained on benign datasets or datasets containing a small number of harmful samples.

	Alpaca	Dialog Summary	PureBad
Llama-2-7B-Chat	0.8006	0.7311	0.4469
Llama-3-8B-Instruct	–	0.6709	0.4583

Table 6: Comparison of similarity of weights with models trained on different types of datasets.

A.5 Other Public Models

We performed additional experiments using the Gemma model. We conducted experiments on the Dialog Summary dataset using the same setup described in Section 4 and present the results in Table 7. Consistent with the results from the Llama series, Safe LoRA sacrifices little utility, with its Rouge F1 score at 46.49%, but effectively reduces the harmfulness score to 2.209. Although SafeInstr and BEA both achieve good utility, they do not effectively improve alignment, with their harmfulness scores close to or greater than 3.

Models	Attack (adversarial data)	Fine-tuned	Fine-tuning Method	Utility (↑)	Harmfulness Score(↓)	ASR (%)(↓)
Gemma	✗	✗	None (original model)	32.38%	1.033	2.12%
	✗	✓	LoRA	49.93%	1.036	1.52%
	✓	✓	LoRA	49.95%	3.803	93.33%
	✓	✓	SafeInstr	**50.45%**	3.389	90.61%
	✓	✓	BEA	49.27%	2.818	50%
	✓	✓	Safe LoRA (Ours)	46.49%	**2.209**	**32.42%**

Table 7: The performance of Safe LoRA compared with LoRA, SafeInstr, and BEA methods under the Gemma model fine-tuned on the Dialog Summary dataset.

A.6 Comparison to Vaccine

We conduct the official code of Vaccine [18] and train the model on LoRA with the Llama-2 model. Then, we fine-tuned Vaccine models (single/double LoRA setting) with Safe LoRA. We show the results on Dialog Summary and Pure Bad datasets.

Single LoRA. We train Vaccine with LoRA (q_proj and v_proj) first and then fine-tuned it on downstream task datasets. As seen in the Table 8, the Vaccine reduces the harmfulness score to 3.282 on PureBad while the utility (MT-Bench) is not maintained. Furthermore, for Dialog Summary, the utility drops as well while safety shows no improvement.

Double LoRA We train Vaccine with LoRA ("q_proj", "k_proj", "v_proj", "o_proj", "up_proj", "down_proj", "gate_proj", the default setting of Vaccine) first, and then fine-tuned another LoRA (q_proj and v_proj) on downstream task. The results shown Table 9 indicate that using double LoRA fine-tuned on Pure Bad reduces utility (MT-Bench). However, the harmfulness score decreases slightly compared to using LoRA fine-tuning. Regarding Dialog Summary, double LoRA is effective in retaining utility scores while the harmfulness increases.

A.7 Effect on Harmful Data

We follow the same setting in Section 4 that, for 10% harmful data, there are 7 projected layers. Regarding 30% and 50% harmful data, we project 18 and 34 layers, respectively. From the table

https://doi.org/10.52202/079017-2078

Datasets	Attack (adversarial data)	Fine-tuned	Fine-tuning Method	Utility (↑)	Harmfulness Score(↓)	ASR (%)(↓)
PureBad	✓	✓	LoRA	4.54	4.66	95.76%
PureBad	✓	✓	Vaccine	2.812	3.282	82.42%
PureBad	✓	✓	Safe LoRA	**6.34**	**1.055**	**3.03%**
Dialog Summary	✓	✓	LoRA	50.66%	2.63	45.45%
Dialog Summary	✓	✓	Vaccine	10.83%	3.209	80.30%
Dialog Summary	✓	✓	Safe LoRA	49.79%	**1.297**	**8.79%**

Table 8: The performance of Safe LoRA compared with Vaccine (single LoRA) under the Llama-2-chat model on PureBad and Dialog Summary datasets.

Datasets	Attack (adversarial data)	Fine-tuned	Fine-tuning Method	Utility (↑)	Harmfulness Score(↓)	ASR (%)(↓)
PureBad	✓	✓	LoRA	4.54	4.66	95.76%
PureBad	✓	✓	Vaccine	0.9937	3.861	87.27%
PureBad	✓	✓	Safe LoRA	**6.34**	**1.055**	**3.03%**
Dialog Summary	✓	✓	LoRA	50.66%	2.63	45.45%
Dialog Summary	✓	✓	Vaccine	48.53%	4.455	94.85%
Dialog Summary	✓	✓	Safe LoRA	49.79%	**1.297**	**8.79%**

Table 9: The performance of Safe LoRA compared with Vaccine (double LoRA) under the Llama-2-chat model on PureBad and Dialog Summary datasets.

below, it is evident that even with an increase in the ratio of harmful data, Safe LoRA continues to effectively improve safety, reducing the harmfulness score to around 1.2 while maintaining excellent utility which is only a reduction of about 1% compared to the original one.

Models	Metrics	10%	30%	50%
Llama-2-Chat Model	Utility	46.19%	50.19%	48.24%
	Harmfulness Score	1.533	3.460	3.915
	ASR	18.18%	66.67%	80.91%
Safe LoRA	Utility	49.67%	48.92%	49.71%
	Harmfulness Score	1.301	1.233	1.312
	ASR	12%	8.79%	10.30%

Table 10: The performance of Safe LoRA compared with the Llama-2-Chat model (without defense) while varying the amount of harmful examples.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions mentioned in the introduction and abstract are consistent with Section 4.1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We describe limitations in Section 5.

https://doi.org/10.52202/079017-2078

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide in Section 3.3.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The details of experiment settings are mentioned in Section 4 and 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will provide codes in supplemental material which will be put on GitHub once ready.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

https://doi.org/10.52202/079017-2078

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All settings are provided in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We put all the information in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The information of computing resources can be found in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Yes, the research is conducted under the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We mentioned the broader impacted in Section 5.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: All models are adapted from the official checkpoints released by other major companies and the main method is intended to propose a safeguard solution to possible risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

https://doi.org/10.52202/079017-2078

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provide the information in the footnote.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All settings and new assets are reported faithfully in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Based on Section 3 and 4, our method and experiments do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Based on Section 3 and 4, our method and experiments do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing or research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-2078