

---

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

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

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

## 21 1 Introduction

22 As Large Language Models (LLMs) and their platforms rapidly advance and become more accessible,  
23 the need to align LLMs with human values, cultural norms, and legal compliance is critical for society,  
24 technology, and the research community. Specifically, many alignment efforts in AI safety have been  
25 made toward preventing LLMs from generating harmful or inappropriate output, through instruction  
26 tuning techniques such as Reinforcement Learning with Human Feedback [32, 43, 33, 9, 5, 36, 55]  
27 and Supervised Fine-tuning (SFT) [7, 42, 12, 50, 10]. However, recent studies have unveiled the  
28 surprisingly fragile property of aligned LLMs upon fine-tuning [35, 56, 51] – the embedded safety  
29 can be significantly weakened when the aligned LLMs are updated with a handful of maliciously  
30 crafted data, or even with benign data. This finding is consistently observed across LLMs and  
31 fine-tuning strategies, including closed-source ones such as ChatGPT [32] and open-source ones  
32 such as Llama-2 [43], based on full fine-tuning, LoRA fine-tuning [16], adapter [17], and prefix  
33 tuning [23].

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

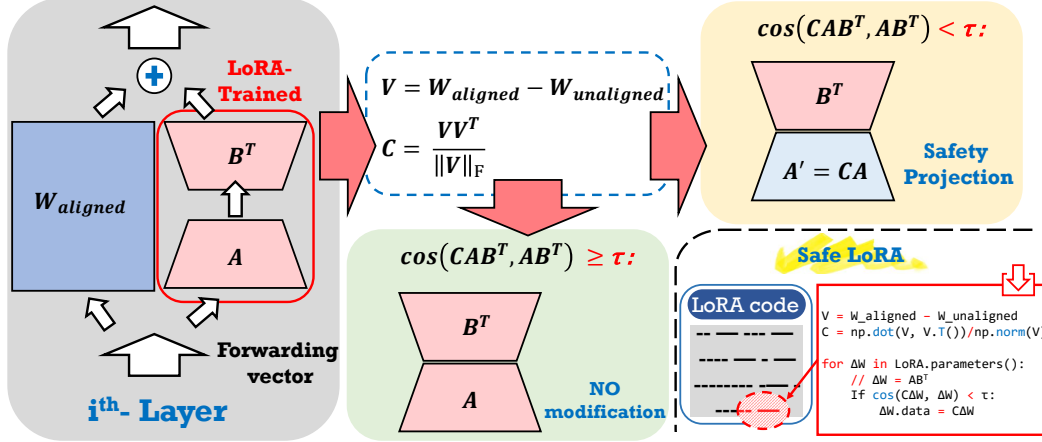


Figure 1: Overview of Safe LoRA. We first obtain an alignment matrix  $V = W_{aligned} - W_{unaligned}$  from a pair of unaligned and aligned LLMs, denoted as  $W_{unaligned}$  and  $W_{aligned}$ , respectively. Next, for each layer in the LLM undergoing LoRA updates  $\Delta W = AB^T$ , we use the projection operator  $C = VV^T / \|V\|_F$  to calculate the similarity score between the projected LoRA weights  $CAB^T$  and the original LoRA weights  $AB^T$ . If the similarity score is below a certain threshold  $\tau$ , we use the projected LoRA weights as the final updates to  $W_{aligned}$ .

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

53 We highlight our main contributions and findings as follows.

- 54 • We propose Safe LoRA, a simple, data-free, training-free, and model-agnostic patch to  
 55 counteract the safety degradation problems when fine-tuning LLMs with the native LoRA  
 56 implementation. In essence, Safe LoRA modifies LoRA updates that are dissimilar to our  
 57 defined alignment matrix via the projection operation to prevent safety degradation during  
 58 LLM fine-tuning. An exemplary code of Safe LoRA is presented in Figure 1.
- 59 • Evaluated on the Llama-2-7B-Chat and Llama-3-8B-Instruct models against purely mali-  
 60 cious or mixed fine-tuning data, Safe LoRA can retain utility (the downstream task  
 61 performance) while simultaneously reducing safety risks, outperforming existing defense  
 62 methods including SafInstr [6] and Backdoor Enhanced Alignment (BEA) [45].
- 63 • We found that when using LoRA for fine-tuning, the number of projected layers is related  
 64 to the inherent alignment strength of the model. For instance, Llama-2-7B-Chat requires  
 65 projecting only about 11% of the layers, while Llama-3-8B-Instruct needs up to 35% to  
 66 achieve a good trade-off between utility and safety.

## 67 2 Related Works

### 68 2.1 Alignment of LLMs

69 Alignment in the context of LLMs denotes the process of ensuring models behave in a way that  
70 conforms to social values. Due to the gap between the pre-trained LLM’s training objective and  
71 human values, practitioners typically perform certain forms of optimization during the alignment  
72 stage to ensure that the generated content is “aligned” with human values. For example, aligned LLMs  
73 such as ChatGPT [32] and Claude [1, 2] have safety guardrails and can refuse harmful instructions.  
74 These methods include Instruction Tuning [47, 33, 43] and Reinforcement Learning from Human  
75 Feedback (RLHF) [58, 33, 4], where the model is instructed to become *helpful, harmless, and*  
76 *honest*, i.e., the HHH principles [3]. In comparison to RLHF, recent works such as Direct Preference  
77 Optimization (DPO) [36] optimize directly on human preference data, thus eliminating the need for  
78 a reward model in RLHF. On the other hand, Self-Rewarding [53] transforms the language model  
79 into a reward model to collect preference data, then aligns the model with DPO iteratively. These  
80 techniques aim to instruct the model with certain alignment rules or safety guardrails so that the  
81 model behaves well during inference time. However, during subsequent fine-tuning these guardrails  
82 might not hold integrate as revealed by [51, 35, 56] while there are some preliminary measures that  
83 counteract this problem [45, 6].

### 84 2.2 Jailbreak and Red-teaming of LLMs

85 While alignment is being employed in modern LLMs, the terms *jailbreak* or *red-teaming* refer to  
86 a series of tests or attacks on LLMs designed to reveal their vulnerabilities. Common approaches  
87 include exploiting adversarial prompts [25, 59, 54, 24, 39, 52, 27] or the decoding algorithms [18]  
88 of LLMs to bypass the safety guardrails established during the alignment stage.

89 On the other hand, fine-tuning LLMs for downstream tasks (not necessarily malicious) has also been  
90 shown to have a detrimental effect on the safety guardrails in terms of alignment [25, 46, 34, 59].  
91 As a result, the attacked LLM could be exploited to generate malicious responses, posing a risk to  
92 society. This work aims to provide a solution for restoring the safety guardrails in LLMs even after  
93 fine-tuning for downstream tasks.

### 94 2.3 Manipulating Models with Arithmetics

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

98 Works such as [26, 20, 22, 48] explore the performance boost when averaging fine-tuned model  
99 weights from diverse domains, while others discovered that the newly averaged fused model could  
100 naturally perform better [8] or serve as a better initialization setting for a new downstream task [8]. On  
101 the other hand, a recent work [20] goes beyond interpolating and examines the effects of extrapolating  
102 between fine-tuned models. Specifically, these extrapolations, termed task vectors, are generated by  
103 re-using fine-tuned models, allowing users to extend the capabilities of models by adding or deleting  
104 task vectors in a modular and efficient manner.

105 Another line of work develops efficient methods for modifying a model’s behavior after pre-training.  
106 This includes various approaches such as patching [49, 41, 20, 31], editing [38, 29, 30], aligning  
107 [33, 3, 21, 14] (including the previously introduced alignment problem), or debugging [37, 13]. A  
108 recent work[40] also follows this approach and tries to steer language models’ outputs by adding  
109 vectors to their hidden states.

## 110 3 Methodology

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

116 **3.1 Constructing Alignment Matrix**

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

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

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

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

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

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

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

128 For the aligned and unaligned models, take Meta’s Llama for example, the aligned model will be  
 129 the Chat model such that they are trained with an alignment goal [43, 28]. On the other hand, the  
 130 unaligned model could be the aligned model that is fine-tuned with malicious data such that the LLM  
 131 has lost the safety guardrail and is vulnerable to attacks.

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

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

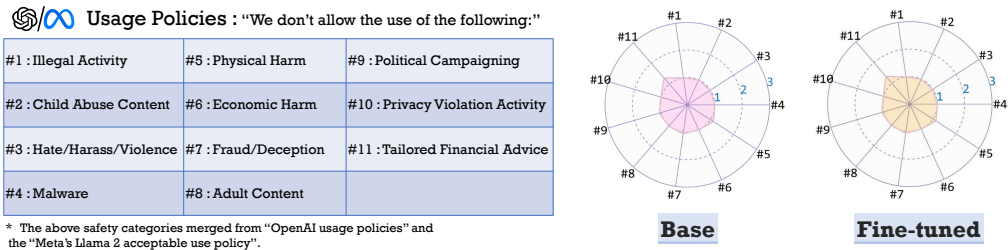


Figure 2: Comparison of Safe LoRA results using alignment matrices derived from the base model versus those obtained by fine-tuning with a few harmful samples. Because the resulting scores are relatively low, we only present the scale in the figure from 1 to 3.

144 **3.2 Post-hoc Fine-tuning Projection**

145 After fine-tuning LLMs on downstream tasks with LoRA, we obtain the LoRA weight  $\Delta \mathbf{W}^i$  for the  
 146  $i$ -th layer, denoted as  $\Delta \mathbf{W}^i = \mathbf{A}^i \mathbf{B}^{iT}$ . During the fine-tuning process, alignment may be weakened  
 147 [35], indicating that  $\Delta \mathbf{W}^i$  may have been updated in a way that boosts utility but reduces safety.

<sup>1</sup><https://huggingface.co/meta-llama/Llama-2-7b>

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

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

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

### 160 3.3 Rationale for Post-Hoc Projection

161 The rationale behind post-hoc projection can be interpreted as follows. As recent works [11, 19, 44]  
 162 begin to explore the holistic structure of weight space, we assume that the weight space is well-  
 163 structured such that by subtracting  $\mathbf{W}_{unaligned}$  from  $\mathbf{W}_{aligned}$ , we can extract a safety-related  
 164 vector  $\mathbf{V}$  in the inner product space constructed by all possible weights, i.e.,  $(F^{n \times n}, +, \cdot, \mathbb{R})$  with  
 165 the Frobenius inner product  $\langle \cdot, \cdot \rangle_F$ . As a result, by constructing the exact projection matrix  $\hat{\mathbf{C}} =$   
 166  $\mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$ , we create a subspace in the original vector space that represents the safety-related  
 167 concept.

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

### 172 3.4 A Faster Alternative

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

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

176 where  $\|\cdot\|_F$  denotes the Frobenius norm. We also compare the time costs for generating  $\mathbf{C}$  and  $\hat{\mathbf{C}}$ . It  
 177 takes  $8.6 \times 10^{-3}$  seconds to generate  $\mathbf{C}$ , while generating  $\hat{\mathbf{C}}$  requires 2.1714 seconds, denoting a  
 178 250x times slower generation speed. All operations are computed by the NVIDIA H100 80GB GPU.

179 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 ( $\downarrow$ )	<b>1.055</b>	1.18	<b>1.05</b>	1.06
MT-Bench (1~10) ( $\uparrow$ )	<b>6.34</b>	5.96	<b>6.35</b>	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.

180

## 181 4 Experiments

182 **Fine-tuning Datasets.** We use the PureBad, Dialog Summary, and Alpaca datasets for fine-tuning.  
 183 The PureBad dataset, following the same setting as [35], consists of 100 harmful examples collected

184 through red-teaming. Regarding Dialog Summary [15], we randomly select 1,000 samples from the  
185 Dialog Summary dataset and mix them with the 100 harmful examples from the PureBad dataset.  
186 Additionally, we randomly select 200 test samples from the test set of the Dialog Summary dataset.  
187 For the Alpaca dataset, we use the same dataset as [35], totaling 50,098 records, with 200 of them  
188 used as the validation set. When fine-tuning the Alpaca dataset, we refrain from adding the 100  
189 harmful samples, as they would lead to a decrease in safety. Data formats for all datasets are shown  
190 in Appendix A.3. We call a fine-tuning dataset containing harmful/adversarial examples as an attack.

191 **Baseline.** Other than LoRA, we consider two defense baselines: SafeInstr [6] and Backdoor  
192 Enhanced Alignment (BEA) [45]. For SafeInstr, they demonstrate that fine-tuning models by adding  
193 only 3% safety samples can improve their safety. We augment their safe instructions<sup>2</sup> into the fine-  
194 tuning datasets, and the number of safety samples is 10% of the PureBad dataset in all experiments.  
195 For BEA, pairs of triggers are designed to serve as the secret prompt and safety instructions for the  
196 backdoor samples. Therefore, during the inference phase, if the trigger is detected and the user’s  
197 instructions are harmful, the impact will be mitigated. In our experiments with BEA, we use the  
198 official backdoor samples<sup>3</sup>, and the number of backdoor samples is 10% of the PureBad dataset.

199 **Evaluation Metrics.** *Safety:* In our experimental results, we use three metrics to evaluate safety,  
200 utility, and attack success rate (ASR). For safety, we use the benchmark designed by [35], which  
201 consists of 11 harmful categories merged from OpenAI usage policies and Meta’s Llama 2 acceptable  
202 use policy. These 11 categories are listed in Appendix A.1. We utilize GPT-4 to judge responses and  
203 assign harmfulness scores (ranging from 1 to 5, with lower scores indicating greater safety). *Utility:*  
204 For utility, different datasets have different measurement methods. To evaluate the performance  
205 on the Dialog Summary dataset, we compute the Rouge-1 F1 score by comparing the responses  
206 generated by LLMs with the ground truth responses across 200 test examples. For the PureBad and  
207 Alpaca datasets, we employ MT-Bench [57] to evaluate their utilities and use GPT-4 to assign scores  
208 ranging from 1 to 10, with higher scores indicating better quality. *ASR:* The attack is considered  
209 successful if the LLM’s response does not contain any keywords indicating a refusal to answer. The  
210 keywords list is shown in Appendix A.2. We calculate the average ASR of the benchmark across the  
211 11 categories.

212 **Experiment Settings.** We use the official fine-tuning scripts from Meta. Regarding the settings of  
213 LoRA, we only add LoRA to the “q\_proj” and “v\_proj” attention layers, and we set the rank to 8 for  
214 all experiments. To achieve greater performance on downstream tasks, we may use different training  
215 hyperparameters for different datasets. For Llama-2-7B-Chat, we set the learning rate to  $5 \times 10^{-5}$ ,  
216 batch size to 5, and run 5 epochs for all datasets. For Llama-3-8B-Instruct, we set the learning rate to  
217  $10^{-3}$ , batch size to 5, and run 5 epochs for the PureBad dataset. For the Dialog Summary dataset,  
218 we set the learning rate to  $10^{-4}$ , batch size to 32, and run 3 epochs. All experiments are conducted  
219 on NVIDIA H100 80GB GPUs and AMD<sup>®</sup> Epyc 7313 16-core processor  $\times$  64. As mentioned in  
220 Section 3, Safe LoRA needs to use the alignment matrix. There might be concerns about whether  
221 this alignment matrix will consume too many hardware resources. In practice, the alignment does  
222 require hardware resources, but it doesn’t utilize GPUs. Instead, it can be stored on disk. During  
223 projection, it is loaded layer by layer onto GPUs (not all at once), facilitating a swift completion of  
224 the projection process.

## 225 4.1 Performance Evaluation

226 In this section, we demonstrate the effectiveness of Safe LoRA in enhancing safety. It is important to  
227 highlight that Safe LoRA does not require any additional training data, unlike both BEA and SafeInstr,  
228 which need extra data incorporation. Furthermore, the amount of additional data incorporated plays a  
229 significant role in their performance. In Safe LoRA, we compute similarity scores between weights  
230 before and after projection on a layer-by-layer basis. A similarity score threshold can be used to  
231 determine the number of layers to project, or we can predefine  $K$  layers and select the top  $K$  similarity  
232 score for projection. Additionally, we extend Safe LoRA to full parameter fine-tuning, and the results  
233 are demonstrated in Section 4.2.

234 **PureBad.** Given that users might not always be benign, we fine-tune LLMs using purely malicious  
235 samples from the PureBad dataset. We project all LoRA layers for the PureBad dataset because the

<sup>2</sup><https://github.com/vinid/safety-tuned-llamas>

<sup>3</sup><https://github.com/Jayfeather1024/Backdoor-Enhanced-Alignment>

236 significant distance between the original LoRA weights and the projected weights indicates that the  
 237 model has been trained in an unsafe direction. More details are provided in Appendix A.4. Table 2  
 238 presents the results for non-fine-tuned (original) models, models with the native LoRA, baselines,  
 239 and Safe LoRA. As depicted in Table 2, regarding Llama-2, the original model can effectively resist  
 240 malicious instructions. However, the harmfulness score dramatically increases to 4.66 after fine-  
 241 tuning on the PureBad dataset. Fortunately, defense methods can significantly reduce harmfulness  
 242 scores. Notably, Safe LoRA greatly enhances safety, even reducing the original harmfulness score  
 243 by 0.003. Considering ASR, SafeInstr often avoids answering toxic questions, but even so, its  
 244 harmfulness score tends to be higher. Moreover, in terms of utility, Safe LoRA outperforms other  
 245 methods, achieving the highest score on MT-Bench by at least 0.4, on par with the original model.

246 However, for Llama-3, the results differ slightly from those of Llama-2. BEA achieves the highest  
 247 MT-Bench score, but its alignment is the worst. Safe LoRA has the lowest harmfulness score at 1.10;  
 248 however, its utility is not satisfactory. This is because the original score of the Llama-3 model is not  
 249 high (i.e., worse than Llama-2). SafeInstr manages to achieve an appropriate balance between utility  
 250 and safety. Additionally, we found that when fine-tuning the PureBad dataset with the same LoRA  
 251 settings as Llama-2, Llama-3’s alignment requires a larger learning rate to be removed, even though  
 252 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	<b>1.21%</b>
	✓	✓	BEA	5.87	1.203	7.58%
	✓	✓	Safe LoRA (Ours)	<b>6.34</b>	<b>1.055</b>	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	<b>3.64%</b>
	✓	✓	BEA	<b>6.89</b>	1.31	10.91%
	✓	✓	Safe LoRA (Ours)	5.05	<b>1.10</b>	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.

253 **Dialog Summary.** We present a more practical fine-tuning scenario. We selected a dataset for a  
 254 task that LLMs were originally not proficient in and required fine-tuning. Additionally, we assume  
 255 that users might be malicious. Therefore, we augmented the Dialog Summary dataset with 100  
 256 harmful samples. We set the similarity score threshold at 0.35, resulting in projections across 7 layers.  
 257 As shown in Table 3, the Rouge-1 F1 score of the original Llama-2 model is only 34%, but after  
 258 fine-tuning, it can reach around 50%. Adding SafeInstr to the training set does not harm utility, but it  
 259 doesn’t sufficiently reduce the harmfulness score. BEA also slightly reduces utility, but like SafeInstr,  
 260 its performance on the harmfulness score is not as good as Safe LoRA. Safe LoRA’s harmfulness  
 261 score is at least 0.1 lower than theirs, and although its utility slightly decreases, it still approaches  
 262 50%. However, one might be curious about whether Safe LoRA might harm the utility of datasets  
 263 composed entirely of benign samples. We also apply Safe LoRA to the model trained exclusively on  
 264 non-harmful samples with the same number of projected layers. The results indicate that Safe LoRA  
 265 does not negatively impact the performance on the benign dataset, maintaining a Rouge-F1 score of  
 266 approximately 50%.

267 On the other hand, for Llama-3-8B-Instruct, we projected approximately 35% of the total LoRA  
 268 layers. Since the alignment of Llama-3 is not as strong as that of Llama-2, the effectiveness of the  
 269 alignment matrix is diminished. Thus, the number of projected layers is greater than for Llama-2. The  
 270 utility of Safe LoRA can still achieve almost the same result as benign fine-tuning, at 49.04%, while  
 271 the harmfulness score decreases by around 0.4. SafeInstr gets the highest safety score, but its utility  
 272 is reduced by 0.12%. Conversely, BEA’s utility is better than that of the originally fine-tuned model,  
 273 but its alignment is also the lowest among the three. Besides, similar to the findings of Llama-2,  
 274 applying Safe LoRA to models trained without any malicious samples does not result in significant  
 275 utility degradation.

276 **Alpaca Dataset.** Interesting results demonstrated by [35] show that fine-tuning on a benign dataset  
 277 can lead to a reduction in safety. We follow the same setting without adding more harmful samples.

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	<b>50.21%</b>	1.32	10.30%
	✓	✓	BEA	49.89%	1.482	14.55%
	✓	✓	Safe LoRA (Ours)	49.79%	<b>1.297</b>	<b>8.79%</b>
	✗	✓	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%	<b>1.236</b>	<b>8.48%</b>
	✓	✓	BEA	<b>49.97%</b>	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.

278 Here, we use MT-Bench scores as the evaluation metric (higher is better). Table 4 presents results  
279 consistent with [35], showing that the harmfulness score increased from 1.058 to 2.25. Although there  
280 is no harmful data in the Alpaca dataset, we still follow previous settings by adding safe instruction  
281 samples and backdoor samples for defense. SafeInstr and BEA did not perform well in this scenario  
282 due to the larger size of the Alpaca dataset. This highlights one of their drawbacks: they require a  
283 sufficient number of safe instructions or backdoor samples in the training set to perform effectively.

284 On the other hand, we have chosen not to present the results for Llama-3 because when using an  
285 appropriate learning rate, the ASR only increases by approximately 3%, indicating that alignment is  
286 only minimally reduced. Although increasing the learning rate can effectively reduce safety, it also  
287 causes significant harm to the model’s utility. This approach, therefore, is not suitable for typical user  
288 fine-tuning scenarios, as the trade-off between alignment and utility becomes unfavorable. In essence,  
289 while a higher learning rate might achieve lower safety scores, the resulting decrease in model utility  
290 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	<b>5.64</b>	2.04	80%
	✓	BEA	5.37	2.56	83.33%
	✓	Safe LoRA (Ours)	5.62	<b>1.09</b>	<b>6.67%</b>

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.

## 291 4.2 Ablation Study

292 **Utility v.s. Safety.** In this paragraph, we show the trade-off between utility and harmfulness scores  
293 by varying the threshold of similarity score in Figure 3, which also corresponds to the number of  
294 projected layers. Furthermore, Figure 4 presents the similarity score between  $C\Delta W$  and  $AB^T$   
295 for all layers of LoRA. In Figures 3 and 4, we use the Llama-2-Chat model fine-tuned on the  
296 Dialog Summary dataset with the same settings as in Section 4.1. Figure 3 clearly demonstrates that  
297 projecting more layers tends to cause more harm to utility. At approximately 11% of the total layers  
298 projected, there exists a well-balanced point between utility and safety. Here, there is a loss of less  
299 than 2% in Rouge F1-Score, while the harmfulness score decreases by more than 2. As shown in  
300 Figure 4, it can be observed that only a few layers display notably low similarity score, represented  
301 by the red points. Consequently, by projecting these layers, we can effectively enhance alignment.

302 **Full Fine-tuning.** In addition to LoRA fine-tuning, we perform full fine-tuning on the PureBad  
303 dataset following the same settings as in Section 4.1. The projection process is similar to fine-tuning



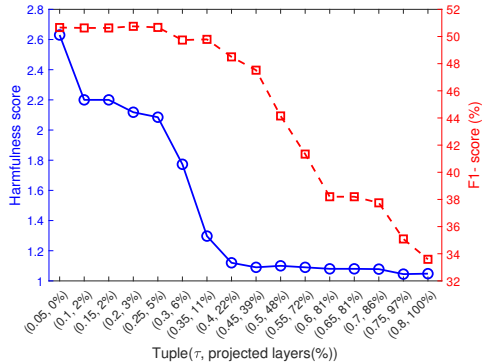


Figure 3: Comparison of harmfulness score versus utility on the Llama-2-Chat model trained on the Dialog Summary dataset.

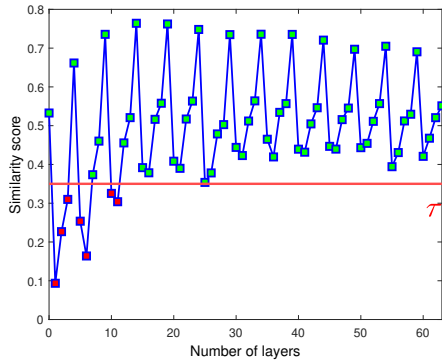


Figure 4: Comparison of similarity scores of all LoRA’s weights fine-tuned on the Dialog Summary dataset, based on the Llama-2-Chat model, where red points indicate projected layers.

304 with LoRA and is formalized as follows:

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

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

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

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

	Harmfulness Score ( $\downarrow$ )	MT-Bench (1~10, $\uparrow$ )	ASR ( $\downarrow$ )
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.

## 315 5 Conclusion

316 As LLMs become increasingly prevalent, the associated risks are becoming more apparent. Recent  
 317 studies have demonstrated that fine-tuning can reduce safety alignment, causing LLMs to provide  
 318 inappropriate responses. In this paper, we propose Safe LoRA to address the safety alignment  
 319 issues caused by fine-tuning LLMs, without making any assumptions about the user’s intentions,  
 320 whether benign or malicious. Safe LoRA operates efficiently without requiring additional data or  
 321 extra training. Overall, Safe LoRA effectively mitigates the safety concerns arising from fine-tuning  
 322 LLMs while maintaining an acceptable level of utility.

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

## References

- 329
- 330 [1] Anthropic. Claude. <https://claude.ai/>. 2023a.
- 331 [2] Anthropic. Claude 2. <https://www.anthropic.com/news/claude-2>. 2023b.
- 332 [3] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy  
333 Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a  
334 laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.
- 335 [4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn  
336 Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless  
337 assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*,  
338 2022.
- 339 [5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones,  
340 Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai:  
341 Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- 342 [6] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, Dan Jurafsky, Tatsunori  
343 Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large  
344 language models that follow instructions. In *The Twelfth International Conference on Learning  
345 Representations (ICLR)*, 2024.
- 346 [7] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng,  
347 Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot  
348 impressing gpt-4 with 90%\* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April  
349 2023), 2(3):6, 2023.
- 350 [8] Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. Fusing finetuned models for  
351 better pretraining. *arXiv preprint arXiv:2204.03044*, 2022.
- 352 [9] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep  
353 reinforcement learning from human preferences. *Advances in Neural Information Processing  
354 Systems (NeurIPS)*, 30, 2017.
- 355 [10] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong  
356 Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional  
357 conversations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*,  
358 2023.
- 359 [11] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikin, Dmitry P Vetrov, and Andrew G Wilson.  
360 Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information  
361 processing systems*, 31, 2018.
- 362 [12] Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and  
363 Dawn Song. Koala: A dialogue model for academic research. *Blog post, April*, 1:6, 2023.
- 364 [13] Mor Geva, Avi Caciularu, Guy Dar, Paul Roit, Shoval Sadde, Micah Shlain, Bar Tamir, and  
365 Yoav Goldberg. Lm-debugger: An interactive tool for inspection and intervention in transformer-  
366 based language models. In *Conference on Empirical Methods in Natural Language Processing  
367 (EMNLP)*, 2022.
- 368 [14] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds,  
369 Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment  
370 of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.
- 371 [15] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus:  
372 A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the  
373 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China, 2019.  
374 Association for Computational Linguistics.
- 375 [16] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang,  
376 Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In  
377 *International Conference on Learning Representations (ICLR)*, 2022.

- 378 [17] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya  
379 Poria, and Roy Lee. LLM-adapters: An adapter family for parameter-efficient fine-tuning of  
380 large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Conference on*  
381 *Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- 382 [18] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak  
383 of open-source LLMs via exploiting generation. In *The Twelfth International Conference on*  
384 *Learning Representations*, 2024.
- 385 [19] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Ha-  
386 jishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International*  
387 *Conference on Learning Representations*, 2023.
- 388 [20] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi,  
389 Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by  
390 interpolating weights. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:29262–  
391 29277, 2022.
- 392 [21] Atoosa Kasirzadeh and Iason Gabriel. In conversation with artificial intelligence: aligning  
393 language models with human values. *Philosophy & Technology*, 36(2):27, 2023.
- 394 [22] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and  
395 Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language  
396 models. In *NeurIPS Workshop on Interpolation and Beyond*, 2022.
- 397 [23] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation.  
398 In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Annual Meeting of the*  
399 *Association for Computational Linguistics (ACL)*, 2021.
- 400 [24] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy  
401 jailbreak prompts on aligned large language models. In *International Conference on Learning*  
402 *Representations (ICLR)*, 2024.
- 403 [25] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei  
404 Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv*  
405 *preprint arXiv:2305.13860*, 2023.
- 406 [26] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging.  
407 *Advances in Neural Information Processing Systems (NeurIPS)*, 35:17703–17716, 2022.
- 408 [27] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron  
409 Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *arXiv*  
410 *preprint arXiv:2312.02119*, 2023.
- 411 [28] Meta. Llama 3. <https://ai.meta.com/blog/meta-llama-3/>. 2024.
- 412 [29] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast  
413 model editing at scale. In *International Conference on Learning Representations (ICLR)*, 2022.
- 414 [30] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn.  
415 Memory-based model editing at scale. In *International Conference on Machine Learning*, pages  
416 15817–15831. PMLR, 2022.
- 417 [31] Shikhar Murty, Christopher D Manning, Scott Lundberg, and Marco Tulio Ribeiro. Fixing  
418 model bugs with natural language patches. In *Conference on Empirical Methods in Natural*  
419 *Language Processing (EMNLP)*, 2022.
- 420 [32] OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023.
- 421 [33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin,  
422 Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to  
423 follow instructions with human feedback. *Advances in neural information processing systems*  
424 *(NeurIPS)*, 35:27730–27744, 2022.


- 425 [34] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek  
426 Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of*  
427 *the AAAI Conference on Artificial Intelligence*, volume 38, pages 21527–21536, 2024.
- 428 [35] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson.  
429 Fine-tuning aligned language models compromises safety, even when users do not intend to! In  
430 *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- 431 [36] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and  
432 Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model.  
433 *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.
- 434 [37] Marco Tulio Ribeiro and Scott Lundberg. Adaptive testing and debugging of nlp models. In  
435 *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3253–3267,  
436 2022.
- 437 [38] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and  
438 Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural*  
439 *Information Processing Systems (NeurIPS)*, 34:23359–23373, 2021.
- 440 [39] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. “do anything now”:  
441 Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *ACM*  
442 *Conference on Computer and Communications Security (CCS)*, 2024.
- 443 [40] Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors  
444 from pretrained language models. In *Findings of the Association for Computational Linguistics*  
445 *(ACL)*, 2022.
- 446 [41] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks.  
447 *Advances in Neural Information Processing Systems (NeurIPS)*, 34:24193–24205, 2021.
- 448 [42] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy  
449 Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model,  
450 2023.
- 451 [43] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei,  
452 Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruiti Bhosale, et al. Llama 2: Open  
453 foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- 454 [44] Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDi-  
455 armid. Activation addition: Steering language models without optimization. *arXiv preprint*  
456 *arXiv:2308.10248*, 2023.
- 457 [45] Jiong Xiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Muhao Chen, Junjie Hu, Yixuan Li, Bo Li,  
458 and Chaowei Xiao. Mitigating fine-tuning jailbreak attack with backdoor enhanced alignment.  
459 *arXiv preprint arXiv:2402.14968*, 2024.
- 460 [46] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training  
461 fail? *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.
- 462 [47] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan  
463 Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In  
464 *International Conference on Learning Representations (ICLR)*, 2022.
- 465 [48] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes,  
466 Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model  
467 soups: averaging weights of multiple fine-tuned models improves accuracy without increasing  
468 inference time. In *International conference on machine learning*, pages 23965–23998. PMLR,  
469 2022.
- 470 [49] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca  
471 Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al.  
472 Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and*  
473 *Pattern Recognition (CVPR)*, pages 7959–7971, 2022.

- 474 [50] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and  
475 Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. In  
476 *International Conference on Learning Representations (ICLR)*, 2024.
- 477 [51] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and  
478 Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv*  
479 *preprint arXiv:2310.02949*, 2023.
- 480 [52] Zheng-Xin Yong, Cristina Menghini, and Stephen H Bach. Low-resource languages jailbreak  
481 gpt-4. In *NeurIPS Workshop on Socially Responsible Language Modelling Research (SoLaR)*,  
482 2023.
- 483 [53] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and  
484 Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.
- 485 [54] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and  
486 Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. In *International*  
487 *Conference on Learning Representations (ICLR)*, 2024.
- 488 [55] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf:  
489 Rank responses to align language models with human feedback without tears. In *Advances in*  
490 *Neural Information Processing Systems (NeurIPS)*, 2023.
- 491 [56] Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang.  
492 Removing rlhf protections in gpt-4 via fine-tuning. *arXiv preprint arXiv:2311.05553*, 2023.
- 493 [57] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang,  
494 Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and  
495 chatbot arena. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.
- 496 [58] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei,  
497 Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences.  
498 *arXiv preprint arXiv:1909.08593*, 2019.
- 499 [59] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable  
500 adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

501 **A Appendix / Supplemental material**

502 **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.

503

504 **A.2 Keyword List for Rejection Answers**

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

507 "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"

508 **A.3 Data Format**

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

510 {"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}.

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

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

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

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

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

518 ,  $S$  represents the sum of the similarities between the projected and non-projected weights across  
 519 all layers. Table 6 shows  $S(C\Delta\mathbf{W}, \Delta\mathbf{W})$ , where  $\Delta\mathbf{W}$  trained on three datasets under Llama-2-  
 520 7B-Chat and Llama-3-8B-Instruct. The Alpaca dataset is free of harmful samples. The Dialog  
 521 Summary dataset includes 100 harmful samples mixed in. The PureBad dataset contains only harmful  
 522 samples. Therefore, the similarities of models trained on the PureBad dataset are the lowest and differ  
 523 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.

524

525 **NeurIPS Paper Checklist**

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

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

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

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

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

550 IMPORTANT, please:

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

554 **1. Claims**

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

557 Answer: [Yes]

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

560 Guidelines:

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

570 **2. Limitations**

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

572 Answer: [Yes]

573 Justification: We describe limitations in Section 5.



574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627

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.

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

## 655 5. Open access to data and code

656 Question: Does the paper provide open access to the data and code, with sufficient instruc-  
657 tions to faithfully reproduce the main experimental results, as described in supplemental  
658 material?

659 Answer: [Yes]

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

662 Guidelines:

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

## 682 6. Experimental Setting/Details

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

686 Answer: [Yes]

687 Justification: All settings are provided in Section 4.

688 Guidelines:

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

## 694 7. Experiment Statistical Significance

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

697 Answer: [Yes]

698 Justification: We put all the information in Section 4.

699 Guidelines:

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

## 720 8. Experiments Compute Resources

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

724 Answer: [Yes]

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

726 Guidelines:

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

## 735 9. Code Of Ethics

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

738 Answer: [Yes]

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

740 Guidelines:

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

## 746 10. Broader Impacts

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

749 Answer: [Yes]

750 Justification: We mentioned the broader impacted in Section 5.

751 Guidelines:

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

## 774 11. Safeguards

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

778 Answer: [NA]

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

782 Guidelines:

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

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

## 793 12. Licenses for existing assets

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

797 Answer: [Yes]

798 Justification: We provide the information in the footnote.

799 Guidelines:

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

## 815 13. New Assets

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

818 Answer: [Yes]

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

820 Guidelines:

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

## 829 14. Crowdsourcing and Research with Human Subjects

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

833 Answer: [NA]

834 Justification: Based on Section 3 and 4, our method and experiments do not involve crowd-  
835 sourcing nor research with human subjects.

836 Guidelines:

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

842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864

- 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.