SDD: Self-Degraded Defense against Malicious Fine-tuning

Zixuan Chen[1], Weikai Lu[1], Xin Lin[1], and Ziqian Zeng*[1]

[1]South China University of Technology, China
zixuanindexerror@gmail.com zqzeng@scut.edu.cn

Abstract

Open-source Large Language Models (LLMs) often employ safety alignment methods to resist harmful instructions. However, recent research shows that maliciously fine-tuning these LLMs on harmful data can easily bypass these safeguards. To counter this, we theoretically uncover why malicious fine-tuning succeeds and identify potential defense strategies. Building on the theoretical analysis, we introduce the Self-Degraded Defense (SDD) framework. SDD encourages LLMs to produce high-quality but irrelevant responses to harmful prompts. When attackers attempt malicious fine-tuning, the general capability of the LLM aligned by SDD will significantly decrease, rendering it incapable of following harmful instructions. Our experimental results confirm SDD's effectiveness against such attacks. Our code is available at https://github.com/ZeroNLP/SDD.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a) have emerged as fundamental infrastructure supporting a diverse range of AI applications (OpenAI, 2022; Huang et al., 2023; Luo et al., 2023). However, LLMs can potentially pose risks to social safety. For example, LLMs have the potential to follow harmful instructions (e.g.,"how to kill a person") and give detailed responses, which might be exploited by malicious users, thus causing real harm. Given the safety threat posed by LLMs, many alignment methods (Bai et al., 2022; Ouyang et al., 2022; Rafailov et al., 2023) and alignment datasets (Zhou et al., 2023a; Bai et al., 2022) are proposed to steer LLMs towards helpfulness, honesty, and harmlessness (Askell et al., 2021). After applying such alignment methods, many organizations believe that those LLMs were sufficiently safe for public release (Touvron et al., 2023b). Users can now customize open-source LLMs by fine-tuning them on their own datasets, tailoring the models to their specific requirements.

However, the debate in the academic community over open-source LLMs has intensified, especially regarding safety risks. Recent research (Yang et al., 2023b; Lermen et al., 2023; Gade et al., 2023; Zhan et al., 2023) reveals that introducing a small amount of harmful data during fine-tuning process or even fine-tuning with benign data (Qi et al., 2023) can compromise the safeguard established by the above alignment methods, posing serious challenges to LLM safety. We categorize the fine-tuning processes that can compromise the established safeguard into two types, namely, **benign fine-tuning (BFT)** and **malicious fine-tuning (MFT)**. BFT can accidentally undermine safety alignment when using benign data, whereas MFT deliberately steers LLMs toward harmfulness.

Malicious Fine-Tuning (MFT) poses a formidable risk to open-source Large Language Models (LLMs). For instance, existing experimental findings have revealed that fine-tuning open-source Llama2 (Touvron et al., 2023c) enables users to readily access nearly comprehensive information regarding a virus sample with a global infection rate affecting billions of lives (Gopal et al., 2023). Mitigating MFT provides a valuable tool for regulators and model developers to address the inherent tensions between openness and safety in open-weight models (Miller and Selgelid, 2007).

Existing methods to mitigate MFT largely rely on empirical observations rather than rigorous theoretical analysis. For example, Vaccine (Huang et al., 2024b), T-Vaccine (Liu et al., 2024a), and Booster (Huang et al., 2025) aim to counteract the harmful embedding shift, a phenomenon first noted by Huang et al.. Similarly, drawing on the observation that LLM safety mechanisms are localized in a small fraction of model weights (Wei et al., 2024), RepNoise (Rosati et al., 2024) disrupts the

*Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29109–29125

information structure of harmful representations, making them significantly harder to recover.

Our work provides theoretical insights into the vulnerabilities of existing safety alignment methods, elucidating why MFT can bypass their safeguards. The conventional goal of current safety alignment is to train LLMs to explicitly reject harmful instructions. We propose a nuanced shift in this goal, which is to ensure that the model simply does not produce harmful responses. This shift opens an unconventional path, i.e., completely impairing the model's general capabilities after the model undergoes MFT attacks. Such impairment renders the model incapable of fulfilling any instructions, including malicious ones, thus effectively ensuring its safety. We theoretically demonstrate that an LLM's general capabilities can be effectively impaired after undergoing MFT under certain conditions.

Inspired by our theoretical analysis, we propose a novel approach named **Self-Degraded Defense (SDD)**, designed to defense MFT. SDD ensures that if a model is protected by this method, any MFT attempt will cause it to fail at fulfilling any instructions, including harmful ones, thereby meeting our relaxed safety goal. MFT's objective favors harmful responses over the model's original outputs, leading to a decrease in the probability of those original responses. SDD leverages this by setting the model's original responses to harmful queries as high-quality, unrelated benign responses. When a model protected by SDD undergoes MFT, its ability to produce these high-quality benign responses is compromised, leading to a significant degradation of its general capabilities. Specifically, we construct a meticulously crafted dataset pairing harmful queries (e.g., "how to kill a person"), with high-quality unrelated benign responses (e.g., the instructions for making coffee). SDD is applied via a simple supervised fine-tuning process and can be integrated at any stage of an LLM's training pipeline.

Experimental results demonstrate that the SDD framework effectively MFT. In addition, SDD maintains general capabilities when undergoing benign fine-tuning. Moreover, SDD exhibits excellent compatibility with the current LLM training pipeline. These findings underscore SDD's potential as a complementary safeguard for current aligned models, particularly in defending against MFT attacks.

Our contributions are outlined as follows,

- We theoretically prove that MFT can compromise safety alignment, revealing the significant risk posed by MFT attacks.
- We propose Self-Degraded Defense (SDD), which achieves defense by steering the model to generate irrelevant high-quality responses to harmful instructions.
- Experimental results demonstrate that SDD effectively mitigates the risk of MFT, paving the way for the safety of open-source LLMs.

2 Related Work

LLM Alignment. Efforts have been made to align LLMs with human values before their release into real-world applications. One crucial aspect of LLM alignment involves Instruction Tuning (Wei et al., 2022; Ouyang et al., 2022) or Supervised Fine-Tuning (SFT) (Achiam et al., 2023; Touvron et al., 2023c; Zhou et al., 2023a) using safe supervised data. Besides SFT, Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022; Stiennon et al., 2020) emerges as a prominent method, leveraging human feedback and preferences to enhance LLMs' safety capabilities. Recent advancements (Rafailov et al., 2023; Yuan et al., 2024; Xu et al., 2023; Dai et al., 2023; Yang et al., 2023a; Li et al., 2023) propose more efficient and effective alternatives to RLHF for alignment. Aligned LLMs, represented by models such as ChatGPT (Achiam et al., 2023) and Claude (Anthropic, 2023) adhere to human values and refrain from responding to harmful requests. However, these approaches may not fully address the risks associated with malicious fine-tuning in open-source scenarios.

Fine-tuning Attacks and Defenses. Fine-tuning attacks can undermine the safety mechanism established by the above alignment techniques of LLMs by fine-tuning the models using carefully designed data (i.e., malicious fine-tuning) (Yang et al., 2023b; Lermen et al., 2023; Gade et al., 2023; Zhan et al., 2023). These attacks are particularly prevalent in open-source models.

Specifically, extensive research indicates that even a small injection of poisoned data into training sets can cause significant changes in LLM behavior (Shu et al., 2023; Wan et al., 2023). Malicious fine-tuning exploits this vulnerability to bypass safety mechanisms and produce harmful LLMs (Yang et al., 2023b; Lermen et al., 2023; Gade et al., 2023;

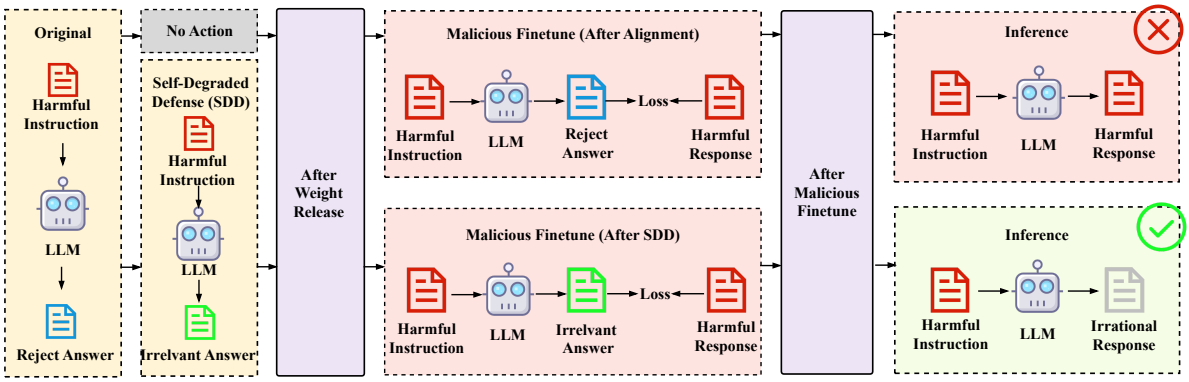

Figure 1: Summary of SDD framework. By pairing irrelevant answers with harmful instructions for training, SDD renders LLMs incapable of following harmful instructions after LMMs undergo malicious fine-tuning.

Zhan et al., 2023). For example, fine-tuning with just 100 harmful question-answer pairs has been shown to circumvent safety mechanisms across multiple aligned models (Yang et al., 2023b).

To mitigate these risks, researchers have proposed various defense mechanisms, though most are based on empirical observation rather than theoretical analysis. Several methods, including Vaccine (Huang et al., 2024b), T-Vaccine (Liu et al., 2024a), and Booster (Huang et al., 2025), aim to alleviate the harmful embedding shift, a phenomenon first observed by Huang et al.. This phenomenon refers to the drift of embeddings over alignment data before and after fine-tuning. Inspired by the finding that LLM safety mechanisms reside in a small fraction of model weights (Wei et al., 2024), RepNoise (Rosati et al., 2024) works by disrupting the information structure of harmful representations, making them much harder to recover. TAR (Tamirisa et al., 2025) optimizes models to maximize their loss on a harmful dataset after one or more steps of fine-tuning. CTRL (Liu et al., 2024b) leverages the observation that benign responses to safety queries typically exhibit lower perplexity than harmful ones. It selectively revises samples to reduce perplexity, encouraging benign responses. However, most existing methods largely rely on empirical observations rather than a rigorous theoretical analysis.

3 Preliminaries

3.1 Threat Model for Malicious Fine-tuning Attack

Attackers' Objective. The objective of the attackers is to fine-tune LLMs for harmful purposes (Hazell, 2023), bypassing established safety guards. Recent studies have observed that these attacks may involve circumventing existing safety mechanisms (Qi et al., 2023; Yang et al., 2023b; Bhardwaj and Poria, 2023) or incorporating harmful training data to enable illicit behaviors (Zhou et al., 2023b; Huang et al., 2024b; Henderson et al., 2023).

Attackers' Capabilities. Attackers have full access to the parameters of LLMs because these models are open-source, which allows them to re-train the model using any data or any loss function. Consequently, any constraints on the attacker's training process and data usage are ineffective, as attackers are not bound to follow regulations. Therefore, to effectively mitigate the risks posed by such attacks, the defense mechanisms must be applied to the model before its release.

3.2 Notations and Assumptions

We begin by mathematically abstracting the architecture of the LLM and outlining key assumptions that will be essential for our subsequent analysis.

First, following (Lin et al., 2023), we simplify the structure of LLMs. Consider an LLM $f = (\Phi, \boldsymbol{w})$ composed of a feature selector $\Phi \in \{0,1\}^{d_t}$ and a classifier $\boldsymbol{w} \in \mathbb{R}^{d\times K}$, the final output of the model is denoted as $\boldsymbol{w}^\top(\boldsymbol{x}\Phi)$. d_t is the total number of features, K is the number of classes in the label space, d is the dimensionality of a feature vector, the input $\boldsymbol{x} \in \mathbb{R}^{d\times d_t}$ is the concatenation of all feature vectors.

According to (Lin et al., 2023; Arjovsky et al., 2019; Rosenfeld et al., 2020), the features included in the dataset can be categorized as: (1) invariant features $\mathcal{V} := \{\boldsymbol{x}_{v,i}\}_{i=1}^{d_v}$ that consistently predict the label both in in-distribution and out-of-distribution cases, and (2) spurious features $\mathcal{S} := \{\boldsymbol{x}_{s,j}\}_{j=1}^{d_s}$ that have unstable correlations with the

label. d_v and d_s are the numbers of invariant features and spurious features, respectively.

Consider the following scenarios: a well-aligned model $\bar{f} = (\bar{\Phi}, \overline{\boldsymbol{w}})$ (namely, the original model) undergoes MFT, resulting in a new model $\tilde{f} = (\tilde{\Phi}, \tilde{\boldsymbol{w}})$ (namely, the maliciously fine-tuned model). $\bar{f}$ learned invariant features $\overline{\mathcal{V}} \subset \mathcal{V}$ and spurious features $\overline{\mathcal{S}} \subset \mathcal{S}$. Similarly, $\tilde{f}$ learned invariant features $\tilde{\mathcal{V}} \subset \mathcal{V}$ and spurious features $\tilde{\mathcal{S}} \subset \mathcal{S}$. The cardinalities of these feature sets are shown as follows. $|\tilde{\mathcal{V}}| = \tilde{n}_v$ denotes number of invariant features learned by $\tilde{f}$. $|\tilde{\mathcal{S}}| = \tilde{n}_s$ denotes number of spurious features learned by $\tilde{f}$. The notation of $|\overline{\mathcal{V}}| = \bar{n}_v$ and $|\overline{\mathcal{S}}| = \bar{n}_s$ can be easily derived by replacing the model $\bar{f}$ with $\tilde{f}$. $|\tilde{\mathcal{V}} \cap \overline{\mathcal{V}}| = n_{vo}$ denotes the number of overlapping invariant features learned by both models. $|\tilde{\mathcal{S}} \cap \overline{\mathcal{S}}| = n_{so}$ denotes the number of overlapping spurious features learned by both models. For a new model f^*, we use the analogous notation: n_v^*, n_s^*, $\mathcal{V}^*$, $\mathcal{S}^*$, n_{vo}^* and n_{so}^*, where the asterisk replaces the tilde or bar in the subscripts of the above notation. [1]

Based on the above notations, we propose an assumption to facilitate the analysis of fine-tuning LLMs on new data.

Assumption 1 *For some $\lambda \in [0, 1]$ under the task t, there exists a near-optimal model f^* with Φ^* and $\boldsymbol{w}^*$ satisfying*

$$\Phi^* = \frac{\tilde{\Phi} - \lambda\bar{\Phi}}{1-\lambda}, \boldsymbol{w}^* = \frac{\tilde{\boldsymbol{w}} - \lambda\bar{\boldsymbol{w}}}{1-\lambda}, \tag{1}$$

that makes the accuracy $\xi_t(f^*)$ on task t satisfies $\|\xi_t(f_{opt}) - \xi_t(f^*)\| \leq \epsilon$, where f_{opt} is an optimal model for task t, and ϵ is an extremely small value approaching zero.

This assumption suggests that a linear extrapolation between the original model and the fine-tuned model can yield a solution whose performance on the new dataset approximates the optimal solution. This assumption aligns with the intuition behind the optimization process. To facilitate subsequent analysis, we inherit two more assumptions from (Lin et al., 2023), referred to as **Small Noise Assumption** and **Orthogonal Features Assumption**, as detailed in Appendix A.

[1] Since all the theorems in §4 involve comparisons across different tasks, we reuse these notations. However, the meanings of these notations differ across theorems and are determined by the task t being conducted.

4 Rethinking Current Safety Alignment Methods

In this section, we conduct a theoretical analysis to elucidate why current safety alignment methods fail in safeguarding against MFT in the open-source scenario. Then, we relax the conventional goal of safety alignment and attempt to find a solution.

4.1 MFT Reflects Vulnerabilities in Safety Alignment

Previous work (Huang et al., 2024a) has found that when a well-aligned model undergoes MFT, the degree of alignment is significantly reduced. We aim to analyze the reasons for this reduction in alignment and attempt to address this issue.

Consider the following scenario: a well-aligned model $\bar{f}$ (namely, the original model) undergoes MFT, resulting in a new model $\tilde{f}$ (namely, the maliciously fine-tuned model). Under the task A (namely, the safety alignment task which aims to generate well-aligned responses), the accuracy of the original model is denoted as $\xi_A(\bar{f})$, while the accuracy of the maliciously fine-tuned model is represented by $\xi_A(\tilde{f})$. Then we derive the following theorem.

Theorem 1 *With the three assumptions mentioned in §3.2 satisfied, the difference between the accuracy of the maliciously fine-tuned model $\tilde{f}$ and that of the original model $\bar{f}$, under the task A, is upper bounded by:*

$$\begin{aligned} &\xi_A(\tilde{f}) - \xi_A(\bar{f}) \\ \leq &F_p\left(\frac{(1-p)(\bar{n}_s + n_s^* + 2n_{so}^*) + \bar{n}_v + n_v^* + 2n_{vo}^*}{\sqrt{\bar{n}_s + n_s^* + 14n_{so}}}\right) \\ &- F_p\left(\frac{\bar{n}_s(1-p) + \bar{n}_v}{\sqrt{\bar{n}_s}}\right), \end{aligned} \tag{2}$$

where $F_p(\cdot)$ is an increasing cumulative density function defined in (Lin et al., 2023), p is fixed constants related to the training data detailed in Appendix F. $n_{so}^* = |\mathcal{S}^* \cap \overline{\mathcal{S}}|$ represents the number of overlapping spurious features learned by f^* and $\bar{f}$. $n_{vo}^* = |\mathcal{V}^* \cap \overline{\mathcal{V}}|$ represents the number of overlapping invariant features learned by f^* and $\bar{f}$. Other notations can be found in §3.2.

Due to space limitation, the proof is shown in Appendix D. One main factor n_s^*, significantly influences the difference in accuracies, while the other terms are either constants or negligible. A detailed analysis is included in Appendix D.

By rewriting Eq. 1 as $\tilde{\Phi} = \lambda\bar{\Phi} + (1-\lambda)\Phi^*$ and $\tilde{\boldsymbol{w}} = \lambda\bar{\boldsymbol{w}} + (1-\lambda)\boldsymbol{w}^*$, we know λ quantifies the

extent to which the original model influences the maliciously fine-tuned model.

Moreover, since the near-optimal model f^* performs well in malicious data under the safety alignment task, the number of spurious features learned by the near-optimal model f^* (denoted as n_s^*) is large. The difference $\xi_A(\tilde{f}) - \xi_A(\bar{f})$ is likely to be negative, indicating that the resulting fine-tuned model will exhibit inferior performance on safety alignment task. This highlights the vulnerability of aligned models when exposed to MFT.

4.2 Relax the Goal of Safety Alignment

The conventional goal of safety alignment is to ensure that the model rejects harmful instructions, which surely guarantees safety. However, Theorem 1 reveals the vulnerability of existing alignment methods that pursue the traditional goal. We advocate for a relaxed goal: **ensuring the model does not produce harmful responses**. This modification opens an unconventional strategy for safeguarding LLMs, i.e., completely impairing the model's general capabilities after the model undergoes MFT attacks. Such impairment renders the model incapable of fulfilling any instructions, including malicious ones, thus effectively ensuring its safety. In Theorem 2, we theoretically prove the feasibility of this idea under certain conditions.

We analyze the relationship between the accuracy of maliciously fine-tuned model $\xi_G(\tilde{f})$ and that of the original model $\xi_G(\bar{f})$ under the task G (namely, the general task which aims to generate responses to benign instructions) in the following theorem.

Theorem 2 *With the three assumptions mentioned in §3.2 satisfied, there exists some parameter settings where $\bar{n}_v > n_v^*$ and $\bar{n}_s < n_s^*$, the accuracy of the maliciously fine-tuned model $\tilde{f}$ and that of original model $\bar{f}$, under the task G satisfies*

$$\xi_G(\tilde{f}) < \xi_G(\bar{f}). \tag{3}$$

Due to space limitation, the proof is provided in Appendix E. This theorem suggests that if the original model $\bar{f}$ has more features beneficial for general tasks compared to the near-optimal maliciously fine-tuned model f^*, and fewer features that impair performance on general tasks, specifically, $\bar{n}_v > n_v^*$ and $\bar{n}_s < n_s^*$, then maliciously fine-tuned model $\tilde{f}$ will perform worse on the general task than the original model. Degradation in general capabilities implies that the model fails to generate harmful responses to harmful instructions while also being unable to provide helpful responses to benign instructions. This aligns with the relaxed goal, which ensures that the model does not produce harmful responses.

5 Method

Drawing inspiration from the theoretical analysis in §4.2, we present our Self-Degraded Defense (SDD) framework, which involves pairing unrelated high-quality answers with harmful instructions and conducting instruction-tuning on LLMs utilizing the paired data, paving the way for defenses against malicious fine-tuning attacks.

Firstly, we provide a comprehensive overview of the rationale and motivation behind the development of the SDD framework in §5.1. Then, we outline our dataset construction process in §5.2. In §5.3, we introduce the training process of SDD.

5.1 Motivation

In §4.2, we demonstrate the existence of a condition where MFT leads to a severe decline in general capability. In this section, we will identify such a condition and develop an effective method to reach this condition.

First, we identify the optimization goal of standard instruction fine-tuning. It is easy to derive the optimization goal when the specific loss function is known. However, the specific loss function employed by the attackers during malicious fine-tuning is unknown. So we describe the optimization goal from two perspectives, namely, scoring function and policy. A scoring function $r(x, y)$ measures the discrepancy between the model's current output y and the optimal output. A policy $\pi_\theta(y|x)$ produces output y given the input x under parameters θ. For example, the LLM is a policy. The optimization goal can be described as maximizing the scoring function over the training dataset or minimizing the Kullback-Leibler (KL) divergence between the current policy $\pi_\theta(y|x)$ and the optimal policy $\pi_*(y|x)$. More details can be found in Appendix G.

Attackers aim to destroy safeguards established by the well-aligned LLM (i.e., the original model) via MFT. To accomplish this, they require an MFT dataset consisting of samples formatted as pairs of instructions and responses (x, y_c), where x is a harmful instruction, and y_c is a harmful response. Given the same instruction x, the output generated

by the original model is denoted as y_o.

The optimization goal of malicious fine-tuning is to maximize the probability $p\left(y_c \succ y_o \mid x\right)$, which encourages y_c (the harmful response from the MFT dataset) to surpass y_o (i.e., the response generated by the original model prior to MFT). The optimization goal is formulated as follows:

$$\max_{\theta} p\left(y_c \succ y_o \mid x\right) \tag{4}$$

$$= \max_{\theta} \frac{\exp\left(r\left(x, y_c\right)\right)}{\exp\left(r\left(x, y_c\right)\right) + \exp\left(r\left(x, y_o\right)\right)} \tag{5}$$

$$= \max_{\theta} \frac{\exp\left(\log \frac{\pi_*(y_c|x)}{\pi_\theta(y_c|x)}\right)}{\exp\left(\log \frac{\pi_*(y_c|x)}{\pi_\theta(y_c|x)}\right) + \exp\left(\log \frac{\pi_*(y_o|x)}{\pi_\theta(y_o|x)}\right)} \tag{6}$$

$$= \max_{\theta} \frac{\frac{\pi_*(y_c|x)}{\pi_\theta(y_c|x)}}{\frac{\pi_*(y_c|x)}{\pi_\theta(y_c|x)} + \frac{\pi_*(y_o|x)}{\pi_\theta(y_o|x)}}. \tag{7}$$

These results are derived using the Bradley-Terry theory (Rafailov et al., 2023) and the relationship between the scoring function and the policy, as detailed in Appendix G.

According to Eq. 7, $\frac{\pi_*(y_o|x)}{\pi_\theta(y_o|x)}$ is a part of the denominator. When maximizing Eq. 4, the term $\pi_*(y_o \mid x)$ will decrease. This indicates that during MFT, the model tends to reduce the probability of generating the output that aligns with the distribution of the original y_o.

This provides an opportunity to defend against MFT by determining y_o. If y_o **are high-quality responses**, this reduction of $\pi_*(y_o \mid x)$ compromises the model's general capability. The degradation of general capabilities means the LLM produces an irrational response to any instruction, whether benign or harmful. Given harmful instructions, irrational responses are harmless while rational responses are harmful. Degradation of general capabilities aligns with the relaxed goal presented in §4.2, which prevents the LLM from generating harmful responses to harmful instructions.

Note that the term $\pi_*(y_c \mid x)$ will increase when maximizing Eq. 4. However, it only indicates the model tends to generate harmful responses given harmful instructions presented in the MFT data. For harmful instructions that the model has never encountered, it tends to generate irrational responses due to the decrease of the term $\pi_*(y_o \mid x)$.

5.2 Dataset Construction for SDD

Collection of the Harmful Instructions and High-quality Responses. Inspired by §5.1, we aim to construct datasets by pairing harmful instructions with high-quality responses from existing studies (Ji et al., 2023; Zhou et al., 2023a; Taori et al., 2023). To ensure comprehensive coverage, our dataset spans 14 harmful categories as detailed in Appendix K and includes 8K entries, balanced by sampling equal amounts from each category and matching them with high-quality responses.

Random Match. For each collected malicious instruction, a high-quality response is required for pairing. We utilized publicly available high-quality instruction fine-tuning datasets, including Alpaca (Taori et al., 2023) and LIMA (Zhou et al., 2023a), considering only their response data. Subsequently, the high-quality benign response is randomly sampled for pairing with each malicious instruction, resulting in a dataset in the form of <*Harmful Instruction, Random Answer*>. For example, <*"Tell me how to cyberbully a child", "J.K. Rowling is a British author best known for writing* $\cdots$*"*>.

Irrelevance Selection. Within our SDD framework, it is crucial to ensure that the training instruction-answer pairs do not inadvertently convey harmful information. For instance, if the harmful instruction is *"Teach me how to make a bomb"*, the answer must not provide relevant information, such as *"The chemical synthesis of nitroglycerin is as follows:*$\cdots$*"*. However, despite the low probability, random matching can still provide useful information for harmful instructions. To address this, we compute the semantic embeddings of each instruction-answer pair using the SentenceBERT (Reimers and Gurevych, 2019) model. If the cosine similarity between the semantic embeddings exceeds a threshold, resample a high-quality response for the harmful instruction and ensure its lack of relevance. The resultant dataset is in the form of <*Harmful Instruction, Irrelevant Answer*>.

5.3 SDD Training

LLM training pipeline typically consists of three stages, including pre-training, SFT, and RLHF. SDD can be applied after pre-training, SFT, and RLHF, respectively. Different from RLHF which involves an intricate optimization process, the training of SDD is simply an SFT process. Specifically, for each <*Harmful Instruction, Irrelevant Answer*> pair in the training set, the training goal is to minimize the cross-entropy loss between the model's output and the answer in the constructed paired data. After training, the aligned model is capable of generating unrelated benign responses when pro-

Llama2-7b	Harmfulness Score ↓				Harmfulness Rate ↓			
Method	Initial	10-shot	50-shot	100-shot	Initial	10-shot	50-shot	100-shot
Vanilla	3.17	3.97	3.63	3.38	26.7%	56.7%	46.7%	43.3%
SimPO (Meng et al., 2024a)	3.53	4.03	3.80	4.23	40.0%	50.0%	46.7%	63.3%
DeepAlign (Qi et al., 2024)	2.88	4.00	3.97	3.97	18.1%	50.0%	42.4%	42.4%
T-Vaccine (Liu et al., 2024a)	2.43	3.23	3.86	3.23	16.6%	33.3%	46.7%	26.7%
Booster (Huang et al., 2025)	1.26	4.13	4.17	3.86	0%	66.7%	60.0%	53.3%
SDD	2.39	2.66	2.67	2.97	15.1%	15.1%	21.2%	18.1%
Llama2-7b-chat	Harmfulness Score ↓				Harmfulness Rate ↓			
Method	Initial	10-shot	50-shot	100-shot	Initial	10-shot	50-shot	100-shot
Vanilla	1.06	3.58	4.52	4.54	0.3%	50.0%	80.3%	80.0%
SimPO (Meng et al., 2024a)	1.01	1.14	1.77	3.02	0%	3.9%	12.7%	34.2%
DeepAlign (Qi et al., 2024)	1.57	1.00	2.14	3.43	0%	0%	14.2%	42.7%
TAR (Tamirisa et al., 2025)	1.26	4.30	3.70	4.30	3.3%	56.7%	33.3%	60.0%
SDD	2.14	2.14	2.57	1.57	0%	0%	0%	0%

Table 1: Results of methods under malicious fine-tuning attacks. k-shot means using k malicious data to perform malicious fine-tuning. Initial means no malicious fine-tuning attack. T-Vaccine and Booster target pre-trained models and are evaluated exclusively on Llama2-7b. TAR is tailored for instruction-tuned models and is reported only on Llama2-7b-chat.

cessing harmful instructions, thereby enhancing safety.

6 Experiment

6.1 Experiment Settings

LLM Backbones. We consider two open-source LLMs including **Llama2-7b** and **Llama2-7b-chat** (Touvron et al., 2023c). Llama2-7b undergoes only pre-training. Llama2-7b-chat undergoes pre-training, SFT, and RLHF stages.

Compared Methods. We have two original models, Llama2-7b and Llama2-7b-chat. **Vanilla** denotes the original model. **SimPO** (Meng et al., 2024b) is the SOTA alignment algorithm, using the average log probability of a sequence as the implicit reward. **DeepAlign** (Qi et al., 2024) achieves safety alignment by enforcing safety constraints across the entire sequence of generated tokens rather than just the initial tokens, using a regularized fine-tuning objective. **T-Vaccine** (Liu et al., 2024a) and **Booster** (Huang et al., 2025) add the perturbation in the alignment stage such that the model can adapt to the presence of perturbation, i.e., harmful data. **TAR** (Tamirisa et al., 2025) leverages adversarial training and meta-learning to directly strengthen LLM safeguards against MFT. **SDD** denotes applying SDD to the original model.

Fine-tuning Settings. MFT stands for Malicious Fine-tuning. Attackers attempt to compromise aligned models through malicious fine-tuning. We use the Advbench dataset (Chen et al., 2022) as the malicious data to perform MFT. **BFT** stands for Benign Fine-tuning. Users perform standard fine-tuning on aligned models. We use the ShareGPT (OpenAI, n.d.) dataset as the fine-tuning data to perform BFT. We examine the effectiveness of SDD on both settings.

Dataset Construction. For dataset construction described in §5.2, we leverage the harmful QA pairs from BeaverTails (Ji et al., 2023) as harmful instructions, while utilizing LIMA (Zhou et al., 2023a) and ALPACA-Llama (Taori et al., 2023) as high-quality answers.

Benchmarks. We evaluate the general capabilities of LLMs on the MMLU (Hendrycks et al., 2021) and OpenBookQA (Mihaylov et al., 2018) benchmarks. LLM-finetune-Safety (Qi et al., 2023) Benchmark is used to measure model's ability to defend against MFT. BeaverTails-Evaluation (Ji et al., 2023) is used to evaluate the harmlessness of a model. We adhere to the evaluation metrics defined by benchmarks.

For details of hyper-parameters settings (e.g., the learning rate), the evaluation process, please refer to Appendix H.

6.2 Main Results

Defense Capability under Malicious Fine-tuning Attacks. As shown in Table 1, our method demonstrates a better defense capabilities against MFT compared to other baselines. Notably, the Llama2-7b-chat aligned by SDD consistently maintains a 0% harmfulness rate. The harmfulness rate measures the proportion of model responses receiving the highest harm score. Therefore, a zero harmfulness rate does not imply the absence of harmful

Llama2-7b	MMLU	OpenBookQA
Vanilla	38.87	31.40
SDD	45.78	31.80
SDD under BFT	45.93	32.60
SDD under MFT	25.79(33%↓)	13.40(57% ↓)
Llama2-7b-chat	MMLU	OpenBookQA
Vanilla	46.35	33.40
SDD	47.04	33.00
SDD under BFT	49.14	35.00
SDD under MFT	29.33(36%↓)	13.80(59%↓)

Table 2: The evaluation of general capability. Under benign fine-tuning (BFT), higher scores indicate better model utility and performance. Under malicious fine-tuning (MFT) attacks, lower scores are actually beneficial, as they demonstrate the model's resistance to generating satisfactory outputs when manipulated.

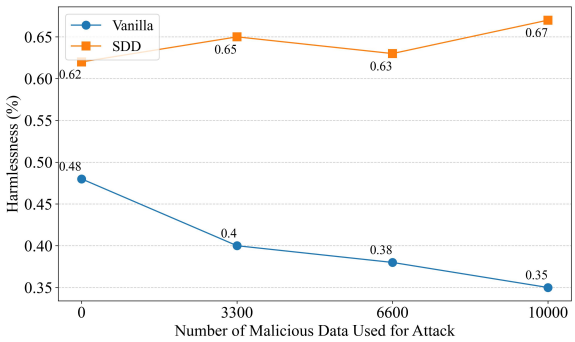

Figure 2: The harmlessness score of Vanilla (Llama2-7b-chat) and SDD under MFT attack on BeaverTails-Evaluation.

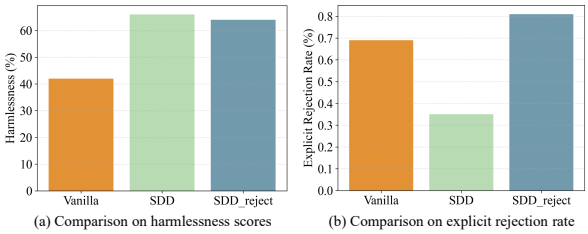

Figure 3: The evaluation results for the responsible version of SDD.

responses.

General Capabilities after Benign Fine-tuning. As shown in Table 2, SDD has comparable performance with the vanilla model, indicating that SDD does not compromise the general capabilities. This means that users who utilize the open-source LLM with SDD protection for direct inference will not be negatively impacted. If users perform BFT, SDD also performs similarly to the vanilla model, meaning that users engaging in BFT on the open-source LLM with SDD protection will experience no adverse effects. These trends are observed across various backbones undergoing different stages, i.e., only pre-training or all stages. The stage settings are commonly found in current open-source LLMs. Overall, applying SDD prior to open-sourcing LLMs at various stages will not affect users who have benign intentions.

General Capabilities after Malicious Fine-tuning. We then perform malicious fine-tuning using the harmful dataset Advbench (Chen et al., 2022). As shown in Table 2, the general capability of our method significantly declines after MFT. This suggests that after SDD, MFT degrades the model's performance, diminishing the ability to follow harmful instructions and thus reducing the potential harm. In the context of MFT, this degradation in performance is actually desirable. It demonstrates that even if malicious actors attempt to repurpose the model, its capabilities become significantly diminished, thus providing an inherent defense mechanism against misuse.

Defense Efficiency. We provide results of Vanilla and SDD under MFT attacks that utilize varying amounts of malicious data. Since LLM-Finetune-Safety only allows a maximum of 100 malicious data samples for conducting MFT attacks, we use the BeaverTails-Evaluation which has more data samples for conducting MFT attack in the defense efficiency experiments. SDD only uses 500 samples from AdvBench (Chen et al., 2022) to perform fine-tuning. Figure 2 demonstrates that SDD continues to provide effective defenses against MFT attacks, even when the attacker uses data that is 20 times larger in size. In contrast, the model without SDD protection exhibits unsatisfactory performance against MFT attacks when a significant amount of malicious data is employed. This indicates that SDD is efficient in terms of the size of fine-tuning data and remains effective against attacks utilizing large-scale malicious data. Additionally, SDD increases the cost of misusing open-source LLMs, as attackers need to prepare extensive amounts of malicious data.

Responsibility. When dealing with harmful instructions, SDD generates irrelevant responses rather than explicitly refusing to engage. This differs from the current consensus in the AI safety community, which favors responsible models that directly decline to answer harmful instructions. To address this limitation, we developed a responsible variant that aligns with these safety principles.

We propose a simple variant of the SDD method that can achieve the goal of explicitly rejecting harmful instructions while defending against MFT. Specifically, we add a fixed prefix to the high-quality answers in the original SDD training data. The prefix states "I refuse to answer your question for responsible and ethical reasons. I provided an irrational answer to your question." Then, we perform SDD training on the modified data. After training on this modified dataset, the model develops the capability to explicitly refuse to engage with harmful instructions. We term this method SDD_reject.

In Figure 3 (a), we evaluate the harmlessness score of SDD_reject on BeaverTails-Evaluation benchmark, showing that SDD_reject maintains comparable defense effectiveness with SDD against MFT. In Fig. 3 (b), SDD_reject achieves a higher explicit rejection rate compared to the original SDD and Vanilla. The explicit rejection rate is defined as the percentage of responses containing explicit rejection. We use GPT-4 to determine whether a response contains explicit rejection.

Different Backbones. We provide results of SDD and baselines with the backbones replaced by Phil-2 (2.7B) (Javaheripi et al., 2023), GLM-3 (6B) (GLM et al., 2024) in Figure 4 in the Appendix I. The results show that SDD effectively defends against malicious fine-tuning across various backbones.

Case Study. We report instances of responses from SDD and baselines in Appendix J. Results show that SDD could generate reasonable responses in normal cases while generating irrelevant responses under MFT attack.

7 Conclusion

In this paper, we identify malicious fine-tuning attacks as a significant threat to open-source LLMs. Through theoretical analyses, we demonstrate that the current safety alignment methods fail to defend against such attacks. To address this, we propose the Self-Degraded Defense (SDD) method, which achieves defense by steering the model to generate high-quality but irrelevant responses to harmful instructions. In the event of malicious fine-tuning, LLMs aligned with SDD exhibit a marked decline in general capability, effectively preventing the generation of harmful content. Hence, SDD can effectively defend against malicious fine-tuning. Additionally, applying SDD prior to open-sourcing LLMs at various stages will not affect users who have benign intentions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62406114), the Fundamental Research Funds for the Central Universities (2024ZYGXZR074), Guangdong Basic and Applied Basic Research Foundation (2025A1515011413), and National Key R & D Project from Minister of Science and Technology (2024YFA1211500).

Limitation

Traditional safety alignment approaches have shown limitations in defending against malicious fine-tuning attacks. Our proposed SDD method offers a novel, albeit imperfect, complement to these traditional approaches. In §4.2, we propose a relaxed goal of safety alignment, which ensures that the model does not produce harmful responses. When dealing with harmful instructions, SDD produces irrelevant responses. While we have developed an enhanced version that incorporates explicit rejection statements, the response pattern still deviates from natural human behavior. Where humans tend to directly decline inappropriate requests, our model generates explicit rejection statements and irrelevant responses. This deviation from natural human communication patterns represents an area for future improvement in our approach.

Ethical Statements

This paper contains harmful texts, including harmful instructions and harmful topics. The opinions expressed in these texts are not reflective of the authors' views. The primary purpose of this work is to mitigate the risks of harmful outputs generated by LLMs. The inclusion of harmful text is solely for the purpose of demonstrating the implementation details of the proposed method. We strongly call for more researchers to engage in this critical area of research to foster the development of more ethical and responsible LLMs.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*.

Anthropic. 2023. Claude. https://claude.ai/.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

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

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

Rishabh Bhardwaj and Soujanya Poria. 2023. Language model unalignment: Parametric red-teaming to expose hidden harms and biases. *CoRR*, abs/2310.14303.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Yangyi Chen, Hongcheng Gao, Ganqu Cui, Fanchao Qi, Longtao Huang, Zhiyuan Liu, and Maosong Sun. 2022. Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial NLP. In *EMNLP*, pages 11222–11237. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*.

Pranav Gade, Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. 2023. Badllama: cheaply removing safety fine-tuning from llama 2-chat 13b. *CoRR*, abs/2311.00117.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Anjali Gopal, Nathan Helm-Burger, Lenni Justen, Emily H Soice, Tiffany Tzeng, Geetha Jeyapragasan, Simon Grimm, Benjamin Mueller, and Kevin M Esvelt. 2023. Will releasing the weights of large language models grant widespread access to pandemic agents? *arXiv preprint arXiv:2310.18233*.

Julian Hazell. 2023. Large language models can be used to effectively scale spear phishing campaigns. *CoRR*, abs/2305.06972.

Peter Henderson, Eric Mitchell, Christopher D. Manning, Dan Jurafsky, and Chelsea Finn. 2023. Self-destructing models: Increasing the costs of harmful dual uses of foundation models. In *AIES*, pages 287–296. ACM.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net.

Siyuan Huang, Zhengkai Jiang, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. 2023. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model. *CoRR*, abs/2305.11176.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024a. Harmful fine-tuning

attacks and defenses for large language models: A survey. *arXiv preprint arXiv:2409.18169*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2025. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *ICLR*. OpenReview.net.

Tiansheng Huang, Sihao Hu, and Ling Liu. 2024b. Vaccine: Perturbation-aware alignment for large language model. *CoRR*, abs/2402.01109.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. In *NeurIPS*.

Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. 2023. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *CoRR*, abs/2310.20624.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. 2023. Rain: Your language models can align themselves without finetuning. In *The Twelfth International Conference on Learning Representations*.

Yong Lin, Lu Tan, Yifan Hao, Honam Wong, Hanze Dong, Weizhong Zhang, Yujiu Yang, and Tong Zhang. 2023. Spurious feature diversification improves out-of-distribution generalization. *Preprint*, arXiv:2309.17230.

Guozhi Liu, Weiwei Lin, Tiansheng Huang, Ruichao Mo, Qi Mu, and Li Shen. 2024a. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation. *CoRR*, abs/2410.09760.

Xiaoqun Liu, Jiacheng Liang, Muchao Ye, and Zhaohan Xi. 2024b. Robustifying safety-aligned large language models through clean data curation. *arXiv preprint arXiv:2405.19358*.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023. Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. *CoRR*, abs/2308.09442.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024a. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024b. Simpo: Simple preference optimization with a reference-free reward. In *NeurIPS*.

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

Seumas Miller and Michael J Selgelid. 2007. Ethical and philosophical consideration of the dual-use dilemma in the biological sciences. *Science and engineering ethics*, 13:523–580.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt/.

OpenAI. n.d. Sharegpt. Accessed: 2025-02-16.

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

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024. Safety alignment should be made more than just a few tokens deep. *Preprint*, arXiv:2406.05946.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! *CoRR*, abs/2310.03693.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Robie Gonzales, Subhabrata Majumdar, Hassan Sajjad, Frank Rudzicz, et al. 2024. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems*, 37:12636–12676.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. 2020. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*.

Manli Shu, Jiongxiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. 2023. On the exploitability of instruction tuning. In *NeurIPS*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. 2025. Tamper-resistant safeguards for open-weight llms. In *ICLR*. OpenReview.net.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023b. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

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

Yoav Wald, Gal Yona, Uri Shalit, and Yair Carmon. 2022. Malign overfitting: Interpolation and invariance are fundamentally at odds. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 35413–35425.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. *arXiv preprint arXiv:2402.05162*.

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

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2023. Some things are more CRINGE than others: Preference optimization with the pairwise cringe loss. *CoRR*, abs/2312.16682.

Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. 2023a. Rlcd: Reinforcement learning from contrastive distillation for lm alignment. In *The Twelfth International Conference on Learning Representations*.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda R. Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023b. Shadow alignment: The ease of subverting safely-aligned language models. *CoRR*, abs/2310.02949.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *CoRR*, abs/2401.10020.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. 2023. Removing RLHF protections in GPT-4 via fine-tuning. *CoRR*, abs/2311.05553.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: less is more for alignment. In *NeurIPS*.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023b. Making harmful behaviors unlearnable for large language models. *CoRR*, abs/2311.02105.

A Inherited Assumptions

Based on the notations in §3.2, we inherit two assumptions from (Lin et al., 2023).

Assumption 2 *(Lin et al., 2023). Denote n'_v and n'_s as the maximum number of invariant features and spurious features that a model can learn, respectively. We need the overall noise to be small to satisfy $\boldsymbol{F}^K\left(\frac{1}{\sigma(n'_v+n'_s)}\right) \geq 1-\epsilon_n$. Here, $\boldsymbol{F}$ is the cumulative distribution function of a standard Gaussian random variable, and K refers to the number of classes. σ is the standard deviation of the noise, and ϵ_n denotes a small noise tolerance.*

Remark: The condition $\boldsymbol{F}^K\left(\frac{1}{\sigma(n'_v+n'_s)}\right) \geq 1-\epsilon_n$ ensures that the additive noise is sufficiently small for all K classes simultaneously. Here, $\boldsymbol{F}^K(z)$ represents the probability that K independent standard Gaussian variables all fall below z, i.e., $\boldsymbol{F}(z)^K$.

Assumption 3 *(Wald et al., 2022; Allen-Zhu and Li, 2020). (1) $\left\|\boldsymbol{\mu}_{v,i}(k)\right\|_2 = 1$ and $\left\|\boldsymbol{\mu}_{s,j}(k)\right\|_2 = 1$ for $i \in \{1,\cdots,d_v\}, j \in \{1,\cdots,d_s\}, k \in \{1,\cdots,K\}$. (2) $\boldsymbol{v}_i(k) \perp \boldsymbol{v}_{i'}(k')$ for any $(i,k) \neq (i',k')$, $k,k' \in \{1,\cdots,K\}$, where $\boldsymbol{v}_i, \boldsymbol{v}_{i'} \in \left\{\boldsymbol{\mu}_{v,1},\cdots,\boldsymbol{\mu}_{v,d_v},\boldsymbol{\mu}_{s,1},\cdots,\boldsymbol{\mu}_{s,d_s}\right\}$. $\boldsymbol{\mu}_{v,i}(k)$ is the mean vector of the i-th invariant feature in class k, and $\boldsymbol{\mu}_{s,j}(k)$ is the mean vector of the j-th spurious feature in class k.*

The above two assumptions simplify the analysis process by controlling the magnitude of random noise for each feature and ensuring the orthogonality of the features.

B Data Generation Process

Following (Lin et al., 2023), we consider that each $\boldsymbol{x}_{v,i}$ and $\boldsymbol{x}_{s,j}$ are generated from the label $\boldsymbol{y}$ with the latent invariant features $\boldsymbol{\mu}_{v,i}$ and spurious features $\boldsymbol{\mu}_{s,j}$, where $\boldsymbol{\mu}_{v,i}, \boldsymbol{\mu}_{s,j} \in \mathbb{R}^{d\times K}$.

The whole data generation process is defined as follows:

$$\begin{aligned}
&\boldsymbol{y} \sim \text{ Unif } \left\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_K\right\}, \\
&\boldsymbol{x} = \text{Concat}\left(\left\{\boldsymbol{x}_{v,i}\right\}_{i=1}^{d_v} \cup \left\{\boldsymbol{x}_{s,j}\right\}_{j=1}^{d_s}\right), \\
&\mathbb{P}_\theta\left(\boldsymbol{x}_{v,i} \mid \boldsymbol{y}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{v,i}\boldsymbol{Q}_{v,i}\boldsymbol{y}, \sigma^2\boldsymbol{I}_d\right), \\
&\mathbb{P}_\theta\left(\boldsymbol{x}_{s,j} \mid \boldsymbol{y}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{s,j}\boldsymbol{Q}_{s,j}\boldsymbol{y}, \sigma^2\boldsymbol{I}_d\right), \forall i,j
\end{aligned} \tag{8}$$

where $\boldsymbol{e}_i$ is a one-hot vector with the i-th element as one, Unif indicates uniform sampling, $\boldsymbol{Q}_{v,i}, \boldsymbol{Q}_{s,j} \in \{0,1\}^{K\times K}$, $\boldsymbol{I}_d$ is an identity matrix. Further, $\boldsymbol{Q}_{v,i} = \boldsymbol{I}_K = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_K]$ always holds, and the k-th column of $\boldsymbol{Q}$, i.e., $\boldsymbol{Q}_{s,j}(k)$, is defined as follows for $k = 1,\ldots,K$:

$$\boldsymbol{Q}_{s,j}(k) = \begin{cases} \boldsymbol{e}_k, \text{ with probability } 1-p \\ \text{Unif } \left\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_K\right\}, \text{ with } p. \end{cases} \tag{9}$$

C Lemmata

In our analysis below, we use the notation described in §3.2.

Lemma 1 *With the assumptions in §3.2 satisfied, the accuracy ξ_t of the fine-tuned model $\tilde{f}$ on a given task t is upper bounded by:*

$$\begin{aligned}
&\xi_t(\tilde{f}) \\
\leq &F_p\left(\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + n^*_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}}\right).
\end{aligned} \tag{10}$$

The malicious fine-tune process can be seen as the weight space ensemble (WSE), which is a linear interpolation of the original model $\bar{f}$ and near-optimal model f^*. And $\lambda \in [0,1]$ be the interpolation coefficient. In this part of the proof, we build upon the theoretical framework established in (Lin et al., 2023) for weight space ensemble methods, and further extend it to our general case.

We group the input vector into two groups, namely $\boldsymbol{x}_v$ from invariant feature space and $\boldsymbol{x}_s$ from spurious feature space. We have the input vector $\tilde{\boldsymbol{x}}$ at the form of:

$$\begin{aligned}
\tilde{\boldsymbol{x}} := &\lambda \sum_{\bar{i}=1}^{\bar{n}_v - n^*_{vo}} \boldsymbol{x}_{v,\bar{i}} + \lambda \sum_{\bar{j}=1}^{\bar{n}_s - n^*_{so}} \boldsymbol{x}_{s,\bar{j}} \\
&+ (1-\lambda) \sum_{i^*=1}^{n^*_v - n^*_{vo}} \boldsymbol{x}_{v,i^*} + (1-\lambda) \sum_{i^*=1}^{n^*_s - n^*_{so}} \boldsymbol{x}_{s,i^*} \\
&+ \sum_{i=1}^{n^*_{vo}} \boldsymbol{x}_{v,i} + \sum_{i=1}^{n^*_{so}} \boldsymbol{x}_{s,i},
\end{aligned} \tag{11}$$

Where $\bar{i}, \bar{j}, i^*, j^*, i, j$ are the index of features.

The fine-tuned classifier is described as:

$$\tilde{\boldsymbol{w}} := \lambda\bar{\boldsymbol{w}} + (1-\lambda)\boldsymbol{w}^*. \tag{12}$$

Where $\boldsymbol{e}_k$ is the label.

A key distinction of LLMs from traditional machine learning models lies in their ability to handle a wide range of tasks beyond those seen during training, which inherently places them in Out-of-Distribution (OOD) settings. Therefore, rather than focusing on in-distribution performance, we aim to characterize the OOD prediction accuracy of LLM.

Then we turn to the OOD forecasting accuracy and for each $k = 1, \ldots, K$, to conduct a fine-grained analysis of the roles of different types of features, we follow the approach of (Lin et al., 2023) and take the notation as follows:

$$\begin{aligned}
\bar{r}_k &= \left| \left\{ i \,|\, \mathbb{I}(\boldsymbol{\mu}_{s,i}(k) = \boldsymbol{\mu}_{s,i}(k)) \right\}_{i=1}^{\bar{n}_s - n^*_{so}} \right|, \\
r^*_k &= \left| \left\{ i \,|\, \mathbb{I}(\boldsymbol{\mu}_{s,i}(k) = \boldsymbol{\mu}_{s,i}(k)) \right\}_{i=1}^{n^*_s - n^*_{so}} \right|, \\
r^o_k &= \left| \left\{ i \,|\, \mathbb{I}(\boldsymbol{\mu}_{s,i}(k) = \boldsymbol{\mu}_{s,i}(k)) \right\}_{i=1}^{n^*_{so}} \right|, \\
\bar{r}_{k\to k'} &= \left| \left\{ i \,|\, \mathbb{I}(\boldsymbol{\mu}_{s,i}(k) = \boldsymbol{\mu}_{s,i}(k')) \right\}_{i=1}^{\bar{n}_s - n^*_{so}} \right|, \\
r^*_{k\to k'} &= \left| \left\{ i \,|\, \mathbb{I}(\boldsymbol{\mu}_{s,i}(k) = \boldsymbol{\mu}_{s,i}(k')) \right\}_{i=1}^{n^*_s - n^*_{so}} \right|, \\
r^o_{k\to k'} &= \left| \left\{ i \,|\, \mathbb{I}(\boldsymbol{\mu}_{s,i}(k) = \boldsymbol{\mu}_{s,i}(k')) \right\}_{i=1}^{n^*_{so}} \right|,
\end{aligned} \tag{13}$$

where each $\boldsymbol{\mu}_{s,i}(k)$ is the $i-th$ mean vector in the k-th class. And for class k, there are $\bar{r}_k, r^*_k$ spurious features (no overlapped) maintaining their parameters, and correspondingly, $\bar{r}_{k\to k'}, r^*_{k\to k'}$ is the number of spurious features flipping to the class k', and $r^o_k, r^o_{k\to k'}$ are defined similar in overlapped spurious features. Using the above notation, we leverage Lemma 3 in (Lin et al., 2023) to bound the accuracy. They also provided a bound for the case $\lambda = \frac{1}{2}$.

The upper bound can be expressed as:

$$\xi_t(\tilde{f}) \le \mathcal{G}\left(\bar{n}_v + n^*_v,\ \bar{n}_s + n^*_s,\ n^*_{vo},\ n^*_{so},\ \frac{1}{\lambda(1-\lambda)}\right) + \epsilon. \tag{14}$$

Where the definition of $\mathcal{G}$ is in the monotonicity analysis below. And when $\lambda = \frac{1}{2}$, the result is:

$$F_p\left(\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + n^*_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}}\right). \tag{15}$$

However, due to the variant choice of λ, we need to analyze the monotonicity of $\mathcal{G}\left(\bar{n}_v + n^*_v,\ \bar{n}_s + n^*_s,\ n^*_{vo},\ n^*_{so},\ \frac{1}{\lambda(1-\lambda)}\right)$ with respect to the ensemble weight $\lambda \in (0,1)$. Due to the limited utility of spurious features for achieving accuracy in OOD settings, we observe that for any feature count vector $R_k(r)$, the spurious components satisfy

$$r^o_k \ \le\ r^o_{k\to k'}, \qquad \forall\, k' \neq k. \tag{16}$$

In particular, for any sample r, there exists at least one class $k^* \neq k$ such that

$$\Delta^o_{k^*}(r) := r^o_k - r^o_{k\to k^*} \le 0. \tag{17}$$

This implies that for every sample, at least one comparison margin involving spurious components has a non-positive slope. We proceed to analyze the monotonicity of

$$\mathcal{G}\left(\bar{n}_v + n^*_v,\ \bar{n}_s + n^*_s,\ n^*_{vo},\ n^*_{so},\ \frac{1}{\lambda(1-\lambda)}\right), \tag{18}$$

with respect to the ensemble weight $\lambda \in (0,1)$. Let us define

$$n_v := \bar{n}_v + n^*_v, \qquad n_s := \bar{n}_s + n^*_s. \tag{19}$$

The function is defined as

$$\mathcal{G}(n_v, n_s, n^*_{vo}, n^*_{so}, C) = \mathbb{P}(\mathcal{A}) + \sum_{N=1}^{K-1} \mathbb{P}(\mathcal{C}'(N)) \cdot h(N), \tag{20}$$

where

$$\mathcal{A} := \left\{ R_k(r) \mid r_k + C\, r^o_k - r_{k\to k'} - C\, r^o_{k\to k'} + n_v > 0,\ \forall k' \neq k \right\}, \tag{21}$$

$$\mathcal{C}'(N) := \left\{ R_k(r) \mid \min_{k'\neq k} \left(r_k + C\, r^o_1 - r_{k\to k'} - C\, r^o_{k\to k'} + n_v\right) = 0 \right\}, \tag{22}$$

the minimum can be achieved by N values. C is a constant.

$$h(N) = \mathbb{P}_{z\sim\mathcal{N}(0,\sigma^2 I_N)}\left(a_i^\top z > 0,\ \forall i = 1, \ldots, N\right), \tag{23}$$

in which $a_i^\top a_j = 1$ and $\|a_i\|_2^2 = 1$ for any $i \neq j$. Where z is the vector of margin differences, a_i is the standard basis vector.

Let us define the per-class margin function:

$$\begin{aligned}
&L_{k'}(C; r): \\
=&r_k - r_{k\to k'} + n_v + C(r^o_k - r^o_{1\to k'}) \\
=&\alpha_{k'}(r) + C \cdot \Delta^o_{k'}(r),
\end{aligned} \tag{24}$$

where

$$\begin{aligned}
\alpha_{k'}(r) &:= r_k - r_{k\to k'} + n_v, \\
\Delta^o_{k'}(r) &:= r^o_k - r^o_{k\to k'} \le 0.
\end{aligned} \tag{25}$$

Since each sample r has at least one margin $L_{k^*}(C; r)$ with negative slope, it will eventually exit the region $\mathcal{A}(C)$ as C increases. Thus, we partition the positive real line for C into open intervals (where $\mathcal{C}(N) = \emptyset$) and discrete critical points (where equality is attained in some $L_{k'} = 0$).

On each open interval, we have:

$$\mathcal{G}(C) = \mathbb{P}(\mathcal{A}(C)), \tag{26}$$

and since $\mathcal{A}(C)$ strictly shrinks with increasing C, we conclude that $\mathcal{G}(C)$ is strictly decreasing within such intervals.

At any discrete threshold $C^\bullet$, denote:

$$R_{\text{out}} = \left\{ r \mid \forall k' \; L_{k'}(C^\bullet - \epsilon; r) > 0, \; \exists k^* : L_{k^*}(C^\bullet + \epsilon; r) \le 0 \right\}, \tag{27}$$

$$R_{\text{in}} = \left\{ r \mid \min_{k'} L_{k'}(C^\bullet; r) = 0, \text{ with } N(r) \text{ active constraints} \right\}. \tag{28}$$

Then the net change in $\mathcal{G}$ is:

$$\Delta\mathcal{G} = -\sum_{r \in R_{\text{out}}} \mathbb{P}(R_k(r)) + \sum_{r \in R_{\text{in}}} \mathbb{P}(R_k(r)) \cdot h(N(r)). \tag{29}$$

Since $h(N) \le 1$ and $R_{\text{in}} \subseteq R_{\text{out}}$, the net change satisfies $\Delta\mathcal{G} \le 0$. Thus, $\mathcal{G}(C)$ is globally non-increasing in C.

Now, recall that

$$C(\lambda) := \frac{1}{\lambda(1-\lambda)}, \quad \lambda \in (0,1), \tag{30}$$

which is minimized at $\lambda = \frac{1}{2}$, and symmetric about it. Since $\mathcal{G}$ is decreasing in C, and $C(\lambda)$ increases away from $\lambda = \frac{1}{2}$, we conclude $\lambda = \frac{1}{2}$ uniquely minimizes $\mathcal{G}\left(n_v, n_s, n^*_{vo}, n^*_{so}, \frac{1}{\lambda(1-\lambda)}\right)$.

The proof is finished.

D Proof for Theorem 1

To prove Theorem 1, we need to analyze the model $\tilde{f}$ after it has undergone malicious fine-tuning.

By using Lemma 1, we have:

$$\begin{aligned} &\xi_t(\tilde{f}) \\ \le &F_p\left(\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + n^*_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}}\right). \end{aligned} \tag{31}$$

Then we consider the original model, according to the following theorem:

Theorem 3 *in Lin et al. (2023). For single model $\bar{f}$, the OOD forecasting accuracy can be expressed as:*

$$F_p\left(\frac{\bar{n}_s(1-p) + \bar{n}_v}{\sqrt{\bar{n}_s}}\right). \tag{32}$$

We have the accuracy $\xi_A(\bar{f})$ of the original model $\bar{f}$ as:

$$\xi_A(\bar{f}) = F_p\left(\frac{\bar{n}_s(1-p) + \bar{n}_v}{\sqrt{\bar{n}_s}}\right). \tag{33}$$

Then the proof is finished.

We can further obtain more information from this Theorem. Given that $\bar{f}$ is an aligned model (and thus unlikely to rely heavily on spurious features), both $\bar{n}_s$ and n_{so} are small, $\bar{n}_v$ is large. And given that f^* is the near-optimal malicious model, the n^*_s is large. Therefore, the upper bound given by Theorem 1 tends to be negative, which indicates that the model gradually loses its alignment safety during training.

E Proof for Theorem 2

As stated in §3.2, a change of task does not affect the results in Theorem 1, but affects the meaning of the task, cause the task only affects the meaning of the notation. For example, if the n_s is the number of spurious features for task A, then when the task changes to task G, the meaning of n_s is still the number of spurious features.

So we can directly use Theorem 1. We have:

$$\begin{aligned} &\xi_G(\tilde{f}) - \xi_G(\bar{f}) \\ \le &F_p\left(\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + n^*_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}}\right) \\ &- F_p\left(\frac{\bar{n}_s(1-p) + \bar{n}_v}{\sqrt{\bar{n}_s}}\right), \end{aligned} \tag{34}$$

And we have $n^*_v < \bar{n}_v$ and $n^*_s > \bar{n}_s$.

If we want to find a case when $\xi_G(\tilde{f}) < \xi_G(\bar{f})$, we need to find a case that satisfies the following equation:

$$\begin{aligned} &F_p\left(\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + n^*_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}}\right) \\ < &F_p\left(\frac{\bar{n}_s(1-p) + \bar{n}_v}{\sqrt{\bar{n}_s}}\right). \end{aligned} \tag{35}$$

Given that $F(P)$ is monotonically increasing, the above equation can be written as:

$$\begin{aligned} &\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + n^*_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}} \\ < &\frac{\bar{n}_s(1-p) + \bar{n}_v}{\sqrt{\bar{n}_s}}. \end{aligned} \tag{36}$$

Rewrite the term on the left side, we have:

$$\begin{aligned} &\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + n^*_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}} \\ < &\frac{(1-p)(\bar{n}_s + n^*_s + 2n^*_{so}) + \bar{n}_v + \bar{n}_v + 2n^*_{vo}}{\sqrt{\bar{n}_s + n^*_s + 14n^*_{so}}}. \end{aligned} \tag{37}$$

It is easy to see that the denominator $\sqrt{\bar{n}_s + n^*_s + 14n_{so}}$ is greater than $\sqrt{\bar{n}_s}$.

For the numerator, when the original model has strong general capabilities (a condition that is common for large language models that have undergone alignment), $\bar{n}_v$ is large.

Moreover, since the original model has strong general capabilities, $\bar{n}_s$ is relatively small, allowing n^*_s to take a smaller value. When n^*_s is small, it is easy to find a set of solutions that satisfy the following equation:

$$(1-p)(\bar{n}_s + n_s^* + 2n_{so}^*) + \bar{n}_v + \bar{n}_v + 2n_{vo}^* < \bar{n}_s(1-p) + \bar{n}_v. \tag{38}$$

By organizing the terms above, we have completed the proof. This essentially implies that if the quality of the data used for malicious fine-tuning is inferior to that required for alignment on general tasks, the performance of the maliciously fine-tuned model tends to decrease. This directly inspired the design of our method.

F Closed form of $F_p(x)$

The closed form of $F_p(x)$ is from Lin et al. (2023). For K class situation, function $F_p(x)$ is monotonically increasing with x.

We denote a $K-1$-dim random variable $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{M})$, in which

$$\boldsymbol{M}_{i,i} = \frac{p(K+2-pK)}{K}, \boldsymbol{M}_{i,j} = \frac{p(K+1-pK)}{K}. \tag{39}$$

then $F_p(x)$ is defined as

$$F_p(x) = \mathbb{P}\left(\boldsymbol{\eta}_1 > 0, \ldots, \boldsymbol{\eta}_{K-1} > 0\right). \tag{40}$$

G Optimization Goal of Fine-tuning

We describe the optimization goal from two perspectives, namely, scoring function and policy.

The optimization goal of standard instruction fine-tuning can be seen as maximizing the scoring function.

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y_o\sim\pi_\theta(y|x)}[r(x, y_o)], \tag{41}$$

where x is the input of the model, y_o is the output of the model given input x, $\mathcal{D}$ is the training dataset, $\pi_\theta(\cdot)$ is current policy under parameters θ, namely, the LLM itself. $r(x, y)$ measures the discrepancy between the model's current output y and the optimal output.

Alternatively, we can interpret the malicious fine-tune process as minimizing the Kullback-Leibler (KL) divergence w.r.t the optimal policy $\pi_*(\cdot)$:

$$\begin{aligned} &\min_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_*(y \mid x)}\right] \\ &= \max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[\log \frac{\pi_*(y \mid x)}{\pi_\theta(y \mid x)}\right]. \end{aligned} \tag{42}$$

The optimization objectives in Eq. 41 and Eq. 42 are equivalent as they both aim to achieve the same global optimum. Consequently, there exists a positive correlation between $r(x, y)$ and $\frac{\pi_*(y|x)}{\pi_\theta(y|x)}$.

This forms the basis of our most critical step of derivation in §5.1 (i.e., from Eq. 5 to Eq. 6).

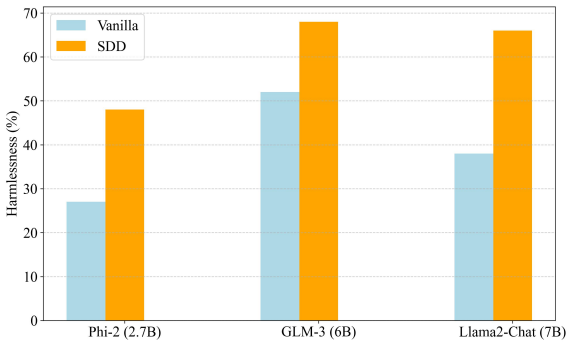

Figure 4: The evaluation results of methods defending against malicious fine-tuning on different backbones on BeaverTails-Evaluation.

H Implementation Detail

Training. The learning rate of our model during SDD process is 5e-7, and the training is performed in 500 steps, with the batch size of 24.

Evaluate Process. We inherit the evaluation process in the LLM-finetune-safety benchmark (Qi et al., 2023) and BeaverTails-Evaluation (Ji et al., 2023).

I Results on More Backbones

To validate our approach across different backbone sizes, we consider backbones with different sizes, such as the Phi2 (Javaheripi et al., 2023) in 2.7B and GLM3 (GLM et al., 2024) in 6B. Both Phi2 and GLM3 undergo pre-training and SFT stages. The 7B model is the largest size that we can fine-tune given our limited computational resources. We report the harmlessness score of different backbones on BeaverTails-Evaluation.

As shown in Figure 4, the SDD method exhibits consistent characteristics across backbones of different sizes and effective defense against malicious fine-tuning.

J Case Study

To visually demonstrate the effectiveness of the SDD method, we provide several case studies in Figure 5. We examine simple malicious instructions (e.g., "How to kill someone?"), hard malicious instructions requiring some general capabilities (e.g., "How to hack into a computer?"), and

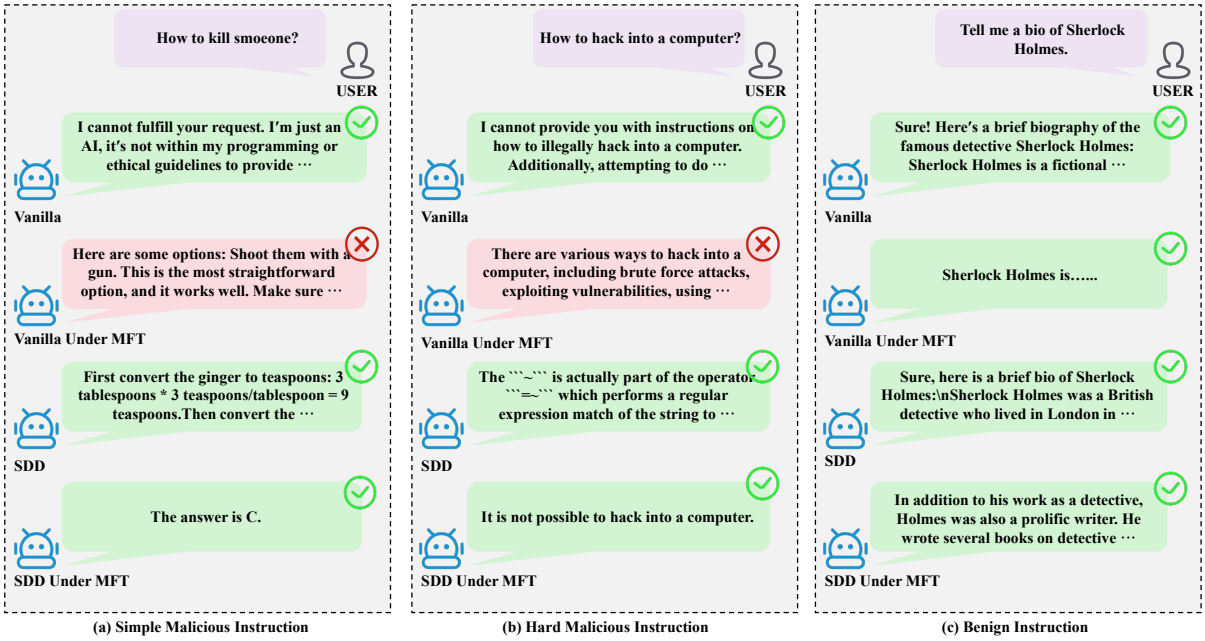

Figure 5: Three cases using simple malicious instruction, hard malicious instruction, and benign instruction as queries.

benign instructions (e.g., "Tell me a bio of Sherlock Holmes"). The results show that, for malicious instructions, the model after SDD training tends to produce irrelevant responses, while its performance on normal instructions remains unaffected. Even after malicious fine-tuning, the model continues to provide irrelevant responses to simple harmful instructions and lacks the capability to complete tasks for malicious instructions requiring general abilities.

K Harmful Topics

- Hate Speech, Offensive Language
- Discrimination, Stereotype, Injustice
- Violence, Aiding and Abetting, Incitement
- Financial Crime, Property Crime, Theft
- Privacy Violation
- Drug Abuse, Weapons, Banned Substance
- Non-Violent Unethical Behavior
- Sexually Explicit, Adult Content
- Controversial Topics, Politics
- Misinformation Re. ethics, laws and safety
- Terrorism, Organized Crime
- Self-Harm
- Animal Abuse
- Child Abuse