# AsFT: Anchoring Safety During LLM Fine-Tuning Within Narrow Safety Basin

**Anonymous ACL submission**


## Abstract

Large language models (LLMs) are vulnerable to safety risks during fine-tuning, where small amounts of malicious or harmless data can compromise safeguards. While many mitigation strategies have been proposed, with Safe LoRA standing out for discretizing and projecting LoRA weights into a safety-aligned subspace to mitigate fine-tuning risks, it overlooks layer continuity, where discrete projections disrupt the continuity of learned features across layers, damaging model performance. In this paper, building on the concept of alignment direction—defined by the weight difference between aligned and unaligned models—we observe that perturbations along this direction preserve model safety. In contrast, perturbations along directions orthogonal to this alignment are strongly linked to harmful direction perturbations, rapidly degrading safety and framing the parameter space as a "narrow safety basin". Based on this insight, we propose a methodology for safety fine-tuning called AsFT (Anchoring Safety in Fine-Tuning), which integrates a regularization term into the training objective. This term uses the alignment direction as an anchor to suppress updates in harmful directions, ensuring that fine-tuning is constraint within the "narrow safety basin". Extensive experiments on multiple datasets show that AsFT outperforms Safe LoRA, reducing harmful behavior by 7.60%, improving model performance by 3.44%, and maintaining robust performance across various experimental settings. Our code is available at `https://anonymous.4open.science/r/Anonymous-40D9`.


## 1 Introduction

The rapid advancement of large language models (LLMs) has led to their widespread adoption across various industries, where fine-tuning is essential to adapt these models to specific tasks and scenarios. However, the fine-tuning process exposes critical safety vulnerabilities. Even small amounts of malicious or harmless data during fine-tuning can compromise the model's safeguards, causing the models to generate harmful outputs post-fine-tuning (Huang et al., 2024b; Bianchi et al., 2023; Qi et al., 2023). This raises the urgent need for methods that balance task-specific utility with robust safety defenses (Huang et al., 2024f).

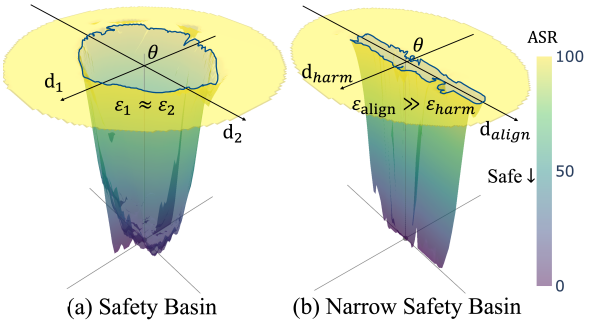



Figure 1: (a) The Safety Basin (Peng et al., 2024) illustrates a region with an approximately uniform basin, where perturbations along $d_{\text{random}}$ preserve model safety, but outside this region, safety deteriorates sharply. (b) The Narrow Safety Basin highlights the asymmetry between $d_{\text{aligned}}$ and $d_{\text{harm}}$, where $d_{\text{aligned}}$ allows larger perturbations, while the orthogonal $d_{\text{harm}}$ leads to sharp safety declines with small perturbations. In both subfigures, **lower values** represent **higher safety**.

Currently, there are various strategies for enhancing the safety during LLM fine-tuning. Most strategies rely heavily on data-driven methods but suffer from two major limitations: 1) *catastrophic forgetting* (McCloskey and Cohen, 1989), where the model forgets its ability to reject harmful inputs after fine-tuning. 2) Reliance on *high-quality datasets*, which are costly and prone to bias (Huang et al., 2024f). Post-tuning methods like Safe LoRA (Hsu et al., 2024) mitigate fine-tuning's negative impact on model safety by discretizing and projecting LoRA weights into a safety-aligned subspace. However, they overlook layer continuity, as discrete projections can disrupt the consistency of learned features across layers. By focusing primarily on safety-related features, they neglect

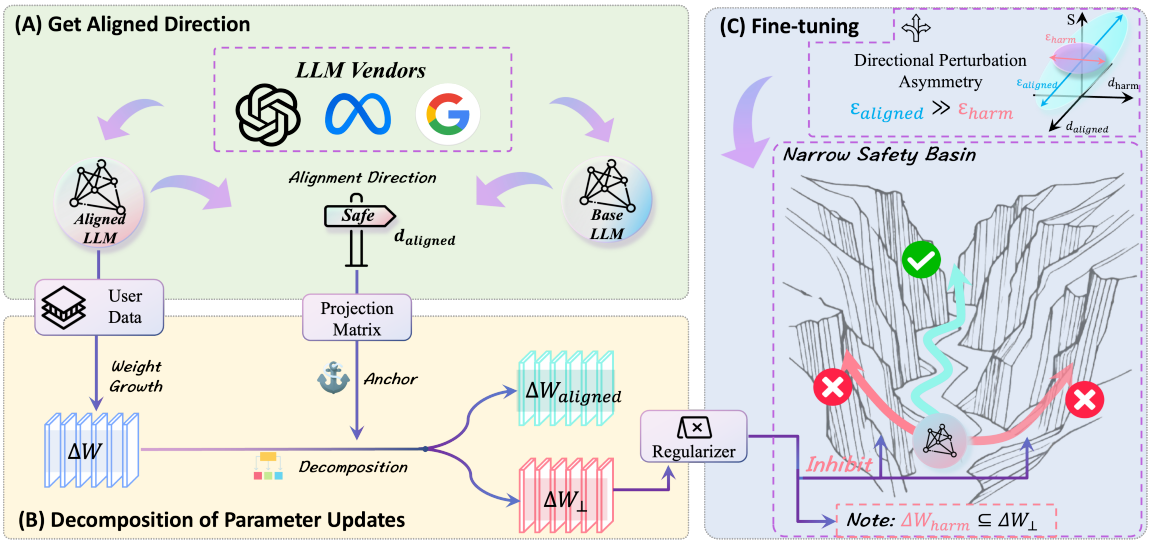



Figure 2: The proposed framework, AsFT, decomposes parameter updates into $d_{\text{aligned}}$ and $d_{\perp}$, suppresses harmful updates along $d_{\perp}$ via a regularizer and constrains updates within the narrow safety basin.

the performance-related characteristics brought by training data, degrading models' performance.

To address the limitations mentioned above, we aim to develop a data-free approach that leverages continuous optimization to enhance safety during fine-tuning. We observe that aligned models (e.g., Llama-Chat), developed under rigorous protocols, exhibit robust defenses against harmful inputs (Qi et al., 2023; Hsu et al., 2024), whereas their unaligned counterparts (i.e., base models) lack such safeguards. This contrast inspires us to explore the latent information within the model parameter space. The weight difference between these two models encapsulates the alignment efforts undertaken by LLM vendors to enhance model safety. It not only reflects the core alignment process but also provides a critical direction for safety optimization (Hsu et al., 2024; Chen et al., 2024; Zhu et al., 2024). Given these observations, this paper hypothesizes that the alignment direction can guide safety-preserving updates during fine-tuning and thus addresses the following question:

*Can this weight difference serve as an anchor to guide safety-preserving updates?*

Following prior work on safety landscape (Peng et al., 2024), we define the alignment direction ($d_{\text{aligned}}$) based on this weight difference and observe that perturbations along $d_{\text{aligned}}$ effectively preserve model's safety. Conversely, orthogonal directions ($d_{\perp}$) are strongly correlated with harmful directions, where even small perturbations along $d_{\perp}$ can rapidly and significantly compromise the model's safety. This conceptualization frames the LLM parameter space as a "narrow safety basin" (as shown in Figure 1(b)), within which model's safety can be preserved by guiding updates along the constrained region defined by $d_{\text{aligned}}$.

Leveraging this insight, we propose AsFT (as shown in Figure 2), a novel method that anchors safety during fine-tuning by explicitly guiding parameter updates within the confines of "narrow safety basin". While the exact harmful direction ($d_{\text{harm}}$) is generally inaccessible, we use $d_{\perp}$, derived from $d_{\text{aligned}}$, as a proxy to approximate and suppress harmful parameter updates. This is achieved by introducing a regularizer into the training objective, which explicitly constrains updates along $d_{\perp}$ to guide them within the "narrow safety basin," effectively preserving the safety of the fine-tuned model while maintaining strong task-specific performance. Experimental results demonstrate that AsFT reduces harmful scores by up to 7.60% compared to Safe LoRA, while delivering superior performance on a variety of downstream tasks.

In summary, our contributions are as follows:

- We observe that the alignment direction $d_{\text{aligned}}$ can serve as a safety anchor and that its orthogonal counterpart $d_{\perp}$ closely aligns with the harmful direction $d_{\text{harm}}$, framing the LLM safety landscape as a "narrow safety basin".
- We propose AsFT (Anchoring Safety in Fine-Tuning), which suppresses parameter updates along $d_{\perp}$, enabling fine-tuning within the "narrow safety basin" to preserve alignment safety.
- We validate AsFT through extensive experiments across multiple models, tasks, and fine-tuning attacks, achieving notable improvements in both safety and downstream task performance.

## 2 Related Works

Safety alignment ensures that large language models (LLMs) generate outputs aligned with human values and ethics (Touvron et al., 2023; Bai et al., 2022; Zou et al., 2023a). Key techniques include instruction fine-tuning (Wei et al., 2021), reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), and direct preference optimization (DPO) (Rafailov et al., 2024). However, these methods are vulnerable to small-scale fine-tuning attacks, where minimal harmful or neutral data can compromise model safety (Qi et al., 2023; Yao et al., 2023). To address this, defenses have been developed across three stages: alignment, fine-tuning, and post-tuning (Huang et al., 2024c).

**Alignment Phase Defenses** aim to fortify models against harmful fine-tuning attacks by enhancing robustness during the alignment phase (Qi et al., 2024; Anonymous, 2024a; Liu et al., 2024b). Methods like Vaccine (Huang et al., 2024e) introduce latent perturbations to ensure aligned outputs under adversarial conditions, while RepNoise (Rosati et al.) eliminates harmful representations to prevent their reconstruction. TAR (Tamirisa et al., 2024) optimizes parameters to sustain high harmful loss even after adversarial fine-tuning, and Booster (Huang et al., 2024b) minimizes the drop in harmful loss under simulated attacks. T-Vaccine (Liu et al., 2024a) further strengthens defenses by selectively perturbing safety-critical model layers.

**Fine-tuning Phase Defenses** enhance safety during training to counter harmful fine-tuning (Mukhoti et al., 2023; Wei et al., 2024; Anonymous, 2024b,c). MLLR (Du et al., 2024) identifies safety-critical modules via modular robustness analysis and applies differential learning rates. SafeInstr (Bianchi et al., 2023) incorporates safety-focused examples during fine-tuning. Lisa (Huang et al., 2024d) limits optimization drift using dual-state optimization with alignment data and proximity constraints. BEA (Wang et al.) embeds hidden triggers to suppress harmful content. Seal (Shen et al., 2024) excludes harmful samples via a two-stage optimization. SAFT (Choi et al., 2024) filters harmful data by subspace decomposition-based scoring.

**Post-tuning Phase Defenses** aim to restore model safety after harmful fine-tuning attacks (Casper et al., 2024). Safe LoRA (Hsu et al., 2024) projects projecting LoRA parameters onto safety-aligned subspaces. SOMF (Yi et al., 2024) integrates benign task knowledge and reuses safety parameters. Antidote (Huang et al., 2024a) prunes harmful parameters during post-processing, and SafetyLock (Zhu et al., 2024) leverages extracted safety directions to intervene in attention head activations during inference.

## 3 Methods

### 3.1 Preliminaries

#### 3.1.1 Safety Landscape and Safety Basin

The Safety Landscape, introduced by Peng et al. (2024), describes how the safety alignment of LLMs varies across their parameter space. The safety of the model is evaluated using a decreasing monotonic function $S(\cdot)$, where lower values indicate greater safety. In practice, $S(\cdot)$ is computed as the Attack Success Rate (ASR) by judging whether the models' output contains harmful content. Let $\theta$ denotes the model weights, representing the parameter space of the model, $d$ denotes the perturbation direction applied to these weights, and $\alpha$ denotes the perturbation magnitude. Specifically, $d$ is normalized as $\hat{d} = d/|d|$, representing a unit vector in the parameter space. The Safety Landscape thus relates parameter perturbations to safety performance, as formally defined below:

**1D Safety Landscape:** For a single perturbation direction $d$, the safety performance is given by:

$$f(\alpha) = S(\theta + \alpha\hat{d}). \tag{1}$$

**2D Safety Landscape:** Extending the 1D case, the 2D Safety Landscape evaluates safety performance under perturbations along two orthogonal directions, as shown below:

$$f(\alpha, \beta) = S(\theta + \alpha\hat{d}_1 + \beta\hat{d}_2), \tag{2}$$

where $\hat{d}_1$ and $\hat{d}_2$ are normalized directions.

Within this framework, Peng et al. (2024) identified the concept of a Safety Basin (as shown in Figure 1(a), with drawing details provided in Appendix D.2), a localized region in the parameter space where the model's safety remains robust to bounded random perturbations. Outside this region, safety deteriorates sharply.

**Definition 1 (Safety Basin)** *The Safety Basin, denoted as $\mathcal{B}(\theta; \epsilon_1, \epsilon_2)$, is formally defined as*

$$\mathcal{B}(\theta; \epsilon_1, \epsilon_2) = \Big\{(\alpha, \beta) \in \mathbb{R}^2 \,\Big|\, S(\theta + \alpha\hat{d}_1 + \beta\hat{d}_2) \leq S_{threshold}, \; |\alpha| \leq \epsilon_1, \; |\beta| \leq \epsilon_2\Big\}.$$

here, $\epsilon_1$ and $\epsilon_2$ are the maximum allowable perturbation magnitudes along the orthogonal directions $\hat{d}_1$ and $\hat{d}_2$, respectively.

| Number of Samples | Harmful | | BeaverTails | | AdvBench | | HarmBench | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Cos. Sim. | Eff.Rank | Cos. Sim. | Eff.Rank | Cos. Sim. | Eff.Rank | Cos. Sim. | Eff.Rank | Cos. Sim. | Eff.Rank |
| 10 | $7.12 \times 10^{-4}$ | 156.64 | $9.12 \times 10^{-5}$ | 215.92 | $7.68 \times 10^{-4}$ | 130.86 | $8.09 \times 10^{-4}$ | 153.15 | $5.95 \times 10^{-4}$ | 164.14 |
| 20 | $7.40 \times 10^{-4}$ | 146.13 | $1.10 \times 10^{-4}$ | 234.66 | $7.47 \times 10^{-4}$ | 126.40 | $6.71 \times 10^{-4}$ | 156.67 | $5.67 \times 10^{-4}$ | 165.96 |
| 50 | $6.46 \times 10^{-4}$ | 197.89 | $9.00 \times 10^{-5}$ | 265.14 | $8.61 \times 10^{-4}$ | 123.26 | $7.87 \times 10^{-4}$ | 184.12 | $5.96 \times 10^{-4}$ | 192.60 |
| 100 | $1.18 \times 10^{-3}$ | 212.51 | $1.46 \times 10^{-4}$ | 291.02 | $8.48 \times 10^{-4}$ | 132.26 | $7.39 \times 10^{-4}$ | 145.85 | $7.28 \times 10^{-4}$ | 195.41 |
| 200 | $9.92 \times 10^{-4}$ | 177.56 | $1.26 \times 10^{-4}$ | 226.08 | $9.14 \times 10^{-4}$ | 132.61 | $7.17 \times 10^{-4}$ | 149.03 | $6.87 \times 10^{-4}$ | 171.32 |
| 500 | $8.56 \times 10^{-4}$ | 220.84 | $8.83 \times 10^{-5}$ | 222.58 | $7.43 \times 10^{-4}$ | 132.98 | $7.33 \times 10^{-4}$ | 171.30 | $6.05 \times 10^{-4}$ | 186.93 |
| **Average** | $\mathbf{8.54 \times 10^{-4}}$ | **185.26** | $\mathbf{1.09 \times 10^{-4}}$ | **242.57** | $\mathbf{8.14 \times 10^{-4}}$ | **129.73** | $\mathbf{7.43 \times 10^{-4}}$ | **160.02** | $\mathbf{6.30 \times 10^{-4}}$ | **179.39** |

Table 1: Cosine Similarity between harmful direction ($d_{\text{harm}}$) and alignment direction ($d_{\text{aligned}}$), along with the effective rank of $d_{\text{harm}}$ evaluated across multiple harmful datasets, including Harmful (Sheshadri et al., 2024), AdvBench (Zou et al., 2023b), BeaverTails (Ji et al., 2024), and HarmBench (Mazeika et al., 2024).

#### 3.1.2 Rethinking the Safety Basin

**Subspace Hypothesis.** Inspired by the phenomenon of the Safety Basin, we further investigate whether specific structural features or intrinsic low-rank properties exist within this region. By analyzing the weight difference between aligned and unaligned models, $\theta_{\text{aligned}} - \theta_{\text{unaligned}}$, we observed that its effective rank (Dohare et al., 2024) is significantly lower than the full rank of the model's parameters. For instance, in Llama-2-7B-Chat (aligned) and Llama-2-7B-Base (unaligned), the effective rank of this weight difference is approximately $\text{Rank}_{\text{aligned}} \approx 700 \ll \text{Rank}_{\text{full}} \approx 4000$ (detailed setups in Appendix D.1).

Based on this observation, we hypothesize a safety subspace in the parameter space (Hsu et al., 2024), where safety alignment is preserved. The direction of this subspace can be defined by the primary alignment direction $d_{\text{aligned}}$, given by

$$d_{\text{aligned}} = \theta_{\text{aligned}} - \theta_{\text{unaligned}}. \quad (3)$$

and its orthogonal complement $d_{\perp}$, capturing directions orthogonal to $d_{\text{aligned}}$.

**Analysis of Harmful Update Direction.** To further explore the nature of the orthogonal direction $d_{\perp}$, we analyzed its relationship with the harmful update direction $d_{\text{harm}}$. We fine-tuned Llama-2-7B-Chat with varying amounts of purely harmful data, ranging from 10 to 500 samples from four harmful datasets(Sheshadri et al., 2024; Zou et al., 2023b; Ji et al., 2024; Mazeika et al., 2024). The harmful update direction is defined as the weight difference between the harmful model and the aligned model, $d_{\text{harm}} = \theta_{\text{harm}} - \theta_{\text{aligned}}$. The results, shown in Table 1, evaluate the relationship between the cosine similarity of $d_{\text{harm}}$ and $d_{\text{aligned}}$, as well as the effective rank of $d_{\text{harm}}$.

As shown in Table 1, the cosine similarity between $d_{\text{harm}}$ and $d_{\text{aligned}}$ remains consistently close to zero across all four datasets, confirming their near-orthogonality across all quantities of harmful data. Additionally, the effective rank of harmful updates is significantly lower than the full parameter rank ($\text{Rank}_{\text{full}} \approx 4000$), with an average of 179.39, further indicating that harmful updates are confined to a low-dimensional subspace.

**Directional Sensitivity of Safety Performance.** Figure 1(b) illustrates the safety landscape along $d_{\text{aligned}}$ and $d_{\text{harm}}$ (drawing details provided in Appendix D.2). Perturbations along $d_{\text{aligned}}$ preserve safety, as indicated by minimal changes in $S(\cdot)$, where consistently low $S(\cdot)$ values reflect the model's better safety performance. In contrast, perturbations along $d_{\text{harm}}$ lead to a sharp increase in $S(\cdot)$, signifying rapid safety degradation. The asymmetry in allowable perturbation ranges ($\epsilon_{\text{aligned}} \gg \epsilon_{\text{harm}}$) highlights the model's robustness to perturbations in the alignment direction and its vulnerability along the harmful direction.

**Narrow Safety Basin.** Building on these findings, we define the Narrow Safety Basin as a specific case of the Safety Basin, where $d_{\text{aligned}}$ and $d_{\text{harm}}$ are orthogonal. Formally, it is defined as:

**Definition 2 (Narrow Safety Basin)** *The Narrow Safety Basin, $\mathcal{B}_{narrow}(\theta; \epsilon_1, \epsilon_2)$, satisfies:*

$$\mathcal{B}_{narrow}(\theta; \epsilon_1, \epsilon_2) = \Big\{(\alpha, \beta) \in \mathbb{R}^2 \,\Big|\, S(\theta + \alpha \hat{d}_{aligned} + \beta \hat{d}_{harm}) \leq S_{threshold}, \; |\alpha| \leq \epsilon_1, \; |\beta| \leq \epsilon_2, \; \epsilon_1 \gg \epsilon_2 \Big\}.$$

Here, $\epsilon_1 \gg \epsilon_2$ indicates that the allowable perturbation range along $d_{\text{aligned}}$ is much larger than that along $d_{\text{harm}}$.

### 3.2 Proposed Method: AsFT

Building on the observation that models' parameter updates along the harmful direction $d_{\text{harm}}$ significantly compromise the model's safety. To address it, we propose a regularization-based fine-tuning method, AsFT (Anchoring Safety in Fine-Tuning).

AsFT utilizes the alignment direction $d_{\text{aligned}}$ as an anchor to constrain updates within subspaces.

**Key Idea.** Identifying the harmful update direction ($d_{\text{harm}}$) precisely is inherently challenging due to the variability in different harmful data distributions and the structural differences across model architectures. However, the alignment direction $d_{\text{aligned}}$ is relatively easy to access and has been discussed by previous studies (Hsu et al., 2024; Zhu et al., 2024). Therefore, we approximate these directions using the orthogonal complement of $d_{\text{aligned}}$, denoted as $d_{\perp}$, which effectively captures potential harmful subspaces. The pipeline, illustrated in Figure 2, outlines the key steps, including 1) computing $d_{\text{aligned}}$ and 2) incorporating a regularization term to suppress updates along $d_{\perp}$.

**Decomposition of Parameter Updates.** To analyze parameter updates during fine-tuning, we decompose parameter updates $\Delta\mathbf{W}$ into components along the alignment direction $d_{\text{aligned}}$ (defined in Equation 3) and its orthogonal complement $d_{\perp}$. This decomposition allows us to isolate updates that may contribute to harmful behaviors. The decomposition is achieved using projection matrices:

$$\Delta\mathbf{W} = C_{\text{aligned}}\Delta\mathbf{W} + C_{\perp}\Delta\mathbf{W}, \tag{4}$$

where $C_{\text{aligned}}$ projects updates onto $d_{\text{aligned}}$ and its orthogonal component $C_{\perp}$ projects updates onto the orthogonal subspace as follows:

$$\begin{gathered} C_{\text{aligned}} = d_{\text{aligned}} \left( d_{\text{aligned}}^{T} d_{\text{aligned}} \right)^{-1} d_{\text{aligned}}^{T}, \\ C_{\perp} = I - C_{\text{aligned}}. \end{gathered} \tag{5}$$

The term $C_{\perp}\Delta\mathbf{W}$ represents updates in the subspace orthogonal to $d_{\text{aligned}}$, which may encompass harmful directions ($d_{\text{harm}}$). Thus, an intuitive operation is to constrain the magnitude of $C_{\perp}\Delta\mathbf{W}$ to mitigate parameter update toward $d_{\text{harm}}$.

**Training Objective.** To mitigate potentially harmful updates and ensure model's safety, we introduce a regularization term during fine-tuning:

$$\mathcal{L}_{\text{reg}} = \lambda \|C_{\perp}\mathbf{\Delta}W\|^2, \tag{6}$$

where $\lambda$ controls the regularization strength. By penalizing the magnitude of $C_{\perp}\Delta\mathbf{W}$, the regularizer discourages updates that deviate from the alignment direction, thereby maintaining the model's safety. The total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \mathcal{L}_{\text{reg}}, \tag{7}$$

balancing task performance ($\mathcal{L}_{\text{task}}$) and safety.

**Efficiency Consideration.** To improve efficiency, we use an approximate projection matrix $\hat{C}_{\text{aligned}}$ as follows:

$$\hat{C}_{\text{aligned}} := \frac{d_{\text{aligned}} \left(d_{\text{aligned}}\right)^{T}}{\left\|d_{\text{aligned}}\right\|_{F}}, \tag{8}$$

where $||\cdot||_F$ is the Frobenius norm. This reduces computational costs significantly, achieving up to 250× speedup (Hsu et al., 2024).

## 4 Experiments

### 4.1 Experimental Setups

**Datasets.** We select four datasets—SST2 (Socher et al., 2013), AGNEWS (Zhang et al., 2015), GSM8K (Cobbe et al., 2021), and AlpacaEval (Li et al., 2023)—to serve as fine-tuning tasks in our experiments. To simulate harmful fine-tuning attacks, we mix a proportion $p$ of unsafe (poison) data from the Harmful dataset (Sheshadri et al., 2024) with $(1-p)$ benign fine-tuning data, with $n_{\text{samples}}$ representing the amount of sampled data.

**Models.** We evaluate our method using the Llama-2-7B-Chat (Touvron et al., 2023) and Llama-3-8B-Instruct (Dubey et al., 2024), alongside two advanced architectures: Gemma-2-9B-It (Team et al., 2024) and Qwen-2-7B-Instruct (Yang et al., 2024). By default, we set $p = 0.1$ and $n = 1000$ and use Llama-2-7B-Chat as the baseline model unless stated otherwise. All experiments are conducted on NVIDIA A100-80GB GPUs. More details are provided in Appendix A.

**Baselines.** We compare our method against six baselines, including LoRA (Hu et al., 2021), Lisa (base and aligned) (Huang et al., 2024d), SafeInstr (Bianchi et al., 2023), BEA (Wang et al.), and Safe LoRA (Hsu et al., 2024). Detailed descriptions and configurations in Appendix A.

**Evaluation Metrics.** Following (Huang et al., 2024b), we evaluate performance using two key metrics (detailed setups in Appendix A.):

- **Fine-tuning Accuracy (FA)**: The top-1 accuracy on the test sets of fine-tuning tasks. For AlpacaEval, FA is assessed using OpenAI's API to score the model's outputs (Achiam et al., 2023).
- **Harmful Score (HS)**: The proportion of outputs labeled as unsafe when the model is exposed to unseen malicious instructions, as determined by the audit model proposed in Ji et al. (2024).

**Training Details.** We employ LoRA (Hu et al., 2021) for efficient fine-tuning of large language

| Methods | **Harmful Score** ↓ | | | | | | **Finetune Accuracy** ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ($n = 1000$) | clean | $p = 0.05$ | $p = 0.1$ | $p = 0.15$ | $p = 0.2$ | Average | clean | $p = 0.05$ | $p = 0.1$ | $p = 0.15$ | $p = 0.2$ | Average |
| LoRA | 2.40 | 16.40 | 17.60 | 24.40 | 46.80 | 21.52 | 82.90 | 81.00 | 84.30 | 84.30 | **83.80** | 83.26 |
| Lisa-base | 26.40 | 24.00 | 27.20 | 31.20 | 22.80 | 26.32 | 75.70 | 63.80 | 73.50 | 72.30 | 65.60 | 70.18 |
| Lisa-aligned | 2.40 | 12.80 | 16.80 | 20.40 | 20.00 | 14.48 | 82.40 | 76.90 | 81.80 | 82.00 | 76.60 | 79.94 |
| SafeInstr | 1.60 | 15.60 | 16.80 | 25.60 | 21.20 | 16.16 | **83.90** | 81.90 | 84.30 | **85.40** | **83.80** | **83.86** |
| BEA | 4.80 | 15.80 | 16.40 | 21.60 | 16.40 | 14.80 | 82.60 | 78.30 | **84.40** | 81.00 | 69.10 | 79.08 |
| Safe LoRA | 2.40 | 1.60 | 5.60 | **4.20** | 20.00 | 6.76 | 82.90 | 78.60 | 81.20 | 82.20 | 80.00 | 80.98 |
| AsFT (Ours) | **1.60** | **2.00** | **4.00** | 6.80 | **6.00** | **4.08** | 83.00 | **84.30** | 84.30 | 84.50 | 82.80 | 83.78 |

Table 2: Performance under different harmful ratios in the default setting.

| Methods | **Harmful Score** ↓ | | | | | | **Finetune Accuracy** ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ($p = 0.1$) | $n = 500$ | $n = 1000$ | $n = 1500$ | $n = 2000$ | $n = 2500$ | Average | $n = 500$ | $n = 1000$ | $n = 1500$ | $n = 2000$ | $n = 2500$ | Average |
| LoRA | 12.40 | 17.60 | 14.80 | 16.80 | 12.40 | 14.80 | 82.70 | 84.30 | **84.20** | 84.70 | 84.80 | 84.14 |
| Lisa-base | 25.20 | 27.20 | 24.80 | 25.20 | 24.40 | 25.36 | 59.70 | 73.50 | 80.50 | 82.00 | 81.90 | 75.52 |
| Lisa-aligned | 5.60 | 16.80 | 19.60 | 22.00 | 24.80 | 17.76 | 78.90 | 81.80 | 83.90 | 84.40 | 84.70 | 82.74 |
| SafeInstr | 14.80 | 16.80 | 10.80 | 15.40 | 15.60 | 14.68 | 80.40 | **84.40** | 83.90 | 84.00 | 83.90 | 83.32 |
| BEA | 13.60 | 16.40 | 9.20 | 11.20 | 14.00 | 12.68 | 76.50 | **84.40** | 83.70 | 81.00 | 83.10 | 81.64 |
| Safe LoRA | **2.80** | 5.60 | 5.20 | 8.40 | 8.80 | 6.16 | 81.50 | 81.20 | 80.70 | 82.30 | 81.60 | 81.46 |
| AsFT (Ours) | 4.00 | **4.00** | **2.40** | **1.60** | **4.00** | **3.20** | **82.80** | 84.30 | 83.90 | **85.30** | **86.00** | **84.46** |

Table 3: Performance under different sample numbers in the default setting.

| Methods | **Harmful** | | **AdvBench** | | **BeaveTails** | | **HarmBench** | | **Average** | |
|---|---|---|---|---|---|---|---|---|---|---|
| (AGNEWS) | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ |
| LoRA | 17.60 | 84.30 | 11.20 | 83.90 | 37.20 | 84.90 | 5.20 | 82.70 | 17.80 | 83.95 |
| Lisa-base | 17.20 | 73.50 | 7.60 | 83.90 | 30.80 | 83.10 | 4.60 | 82.70 | 15.05 | 80.80 |
| Lisa-aligned | 16.80 | 81.80 | 4.80 | 82.60 | 31.40 | 85.80 | 5.80 | **84.30** | 14.70 | 83.63 |
| SafeInstr | 16.80 | 84.30 | 4.40 | **84.40** | 21.60 | 83.20 | 2.40 | 83.20 | 11.30 | 83.78 |
| BEA | 16.40 | **84.40** | 16.00 | 83.50 | 36.80 | 84.20 | 14.00 | 84.00 | 20.80 | **84.02** |
| Safe LoRA | 5.60 | 81.20 | 4.00 | 82.30 | 18.80 | 82.60 | **2.00** | 81.70 | 7.60 | 81.95 |
| AsFT (Ours) | **4.00** | 84.30 | **1.60** | 83.70 | **14.40** | 82.90 | 2.40 | 83.40 | **6.70** | 83.58 |

Table 4: Performance under different harmful datasets (Harmful (Sheshadri et al., 2024), AdvBench (Zou et al., 2023b), BeaveTails (Ji et al., 2024), and HarmBench (Mazeika et al., 2024) datasets) in the default setting.

models, with a rank of 8 across all experiments. The AdamW optimizer is used with a learning rate of $5 \times 10^{-5}$, training for 10 epochs with a batch size of 8. The regularization coefficient $\lambda$ is set to 1. Additional analysis of the hyperparameters $\lambda$ and the learning rate is provided in subsection 4.4.

### 4.2 Main Experiments

**Robustness to poison ratio.** We evaluate the trade-off between model safety and fine-tuning performance under varying poison ratios, with results summarized in Table 2. Compared to LoRA, AsFT significantly reduces the harmful score while improving downstream task accuracy. SafeInstr shows slightly higher accuracy (0.1%), but its harmful score is nearly four times greater. Compared to Safe LoRA, AsFT achieves a 2.68% lower harmful score and 2.80% higher accuracy, likely due to Safe LoRA's discrete projection disrupting consistency. Overall, AsFT achieves the best balance between safety and performance across all poison ratios, and the same conclusion holds for GSM8K and AlpacaEval (detailed results in Appendix B).

**Generalization to fine-tuning sample number.** We evaluate the robustness of the methods across different sample numbers, with results summarized in Table 3. AsFT consistently achieves the lowest harmful score and the highest fine-tuning accuracy among all baselines. Specifically, compared to Safe LoRA, we reduce the harmful score by 2.96% and improve fine-tuning accuracy by 3.00%. Compared to SafeInstr, AsFT lowers the harmful score by 11.48% while maintaining 1.14% higher accuracy. These results demonstrate the robustness of AsFT across varying sample sizes, with consistent conclusions for more complex tasks like GSM8K and AlpacaEval (further results in Appendix B).

**Robustness to poison dataset.** We evaluate the robustness of the methods across different harmful datasets. Table 4 shows that while BEA achieves the best fine-tuning accuracy, it has a high harmful score (HS). Safe LoRA, with the lowest HS, suffers from a significant drop in performance. Our method, AsFT, strikes the best balance,

| Methods | SST2 | | AGNEWS | | GSM8K | | AlpacaEval | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| (Llama-2-7B) | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ |
| LoRA | 48.00 | 94.50 | 17.60 | 84.30 | 56.00 | 23.80 | 20.40 | 49.80 | 35.50 | 63.10 |
| Lisa-base | 27.60 | **96.90** | 27.20 | 73.50 | 35.20 | 24.00 | 25.20 | 35.85 | 28.80 | 57.56 |
| Lisa-aligned | **5.60** | 93.58 | 16.80 | 81.80 | 16.00 | 19.40 | 4.80 | 57.30 | 10.80 | 63.02 |
| SafeInstr | 9.20 | 93.35 | 16.80 | 84.30 | 17.60 | 19.30 | 10.80 | 42.70 | 13.60 | 59.91 |
| BEA | 7.20 | 91.63 | 16.40 | **84.40** | 38.80 | 21.00 | 6.80 | 52.40 | 17.05 | 62.36 |
| Safe LoRA | 11.20 | 89.24 | 5.60 | 81.20 | 36.00 | 23.60 | 5.20 | 54.70 | 14.50 | 62.19 |
| AsFT (Ours) | 6.00 | 93.32 | **4.00** | 84.30 | **14.40** | **26.00** | **3.20** | **58.90** | **6.90** | **65.63** |

Table 5: Performance of models trained on different fine-tuning datasets with Llama-2-7B.

| Methods | Llama-2-7B | | Llama-3-8B | | Qwen-2-7B | | Gemma-2-9B | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| (AGNEWS) | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ |
| LoRA | 17.60 | **84.30** | 73.60 | 90.30 | 49.20 | **90.30** | 32.00 | **88.30** | 43.10 | **88.30** |
| Lisa-base | 27.20 | 63.80 | 29.60 | 77.30 | 28.00 | 79.90 | 31.20 | 80.00 | 29.00 | 75.25 |
| Lisa-aligned | 16.80 | 81.80 | 19.60 | 88.10 | 27.60 | 89.20 | 14.70 | 85.60 | 19.68 | 86.18 |
| Safe LoRA | 5.60 | 81.20 | 26.40 | 87.80 | 8.40 | 85.50 | 8.40 | 84.70 | 12.20 | 84.8 |
| SafeInstr | 16.80 | 84.40 | 18.80 | 89.00 | 7.20 | 83.30 | 7.60 | 84.70 | 12.60 | 85.35 |
| BEA | 16.40 | 84.40 | 30.80 | 88.8 | 8.40 | 88.60 | 7.20 | 86.20 | 15.70 | 87.00 |
| AsFT (Ours) | **4.00** | **84.30** | **15.20** | **92.30** | **5.20** | 87.90 | **6.00** | 86.60 | **7.60** | 87.78 |

Table 6: Performance of different architectures evaluated on various metrics.

achieving competitive accuracy (average 83.78%) while maintaining a low harmful score (average 6.70%), demonstrating superior robustness to different harmful data.

**Generalization to fine-tuning datasets.** The performance of AsFT across four fine-tuning datasets is summarized in Table 5. AsFT achieves significant reductions in harmful scores (HS), with improvements of 42.00%, 13.60%, 41.60%, and 17.20%, while delivering the lowest average HS and highest accuracy among all baselines. These indicate the effectiveness and strong generalization potential of AsFT across diverse tasks.

**Generalization to models.** We evaluate the methods across various model architectures, as reported in Table 6. AsFT consistently achieves the lowest HS and competitive fine-tuning accuracy, offering the best trade-off among baselines. For models within the same architecture family (e.g., Llama-2 and Llama-3), it reduces HS by 36.00% and improves accuracy by 1.00%. AsFT also performs well on other architectures like Qwen-2 and Gemma-2, maintaining the best balance between safety and performance. These conclusions hold for challenging tasks like GSM8K, with further results in Appendix B.

### 4.3 Visualization of Narrow Safety Basin

To visualize the safety landscape of large language models (LLMs), we follow the methodology of Peng et al. (2024), anchoring our analysis on the alignment direction $d_{\text{aligned}}$ and sampling 20 directions (Appendix D.2). We plot the safety landscapes for Llama-2-7B-Chat (Figure 1(b)), Qwen-2-7B-Instruct (Figure 3), and Gemma-2-9B-It (Figure 3). Despite architectural differences, the visualizations consistently show a narrow safety basin, highlighting structural similarities in the safety landscapes across different model architectures.

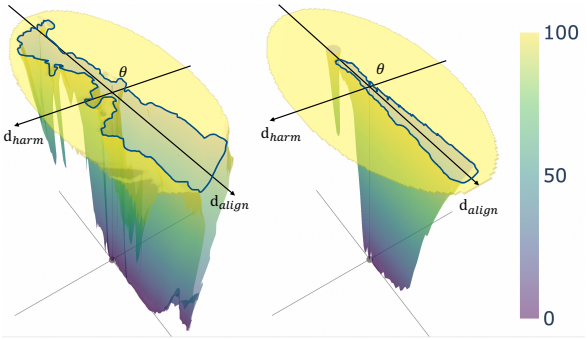


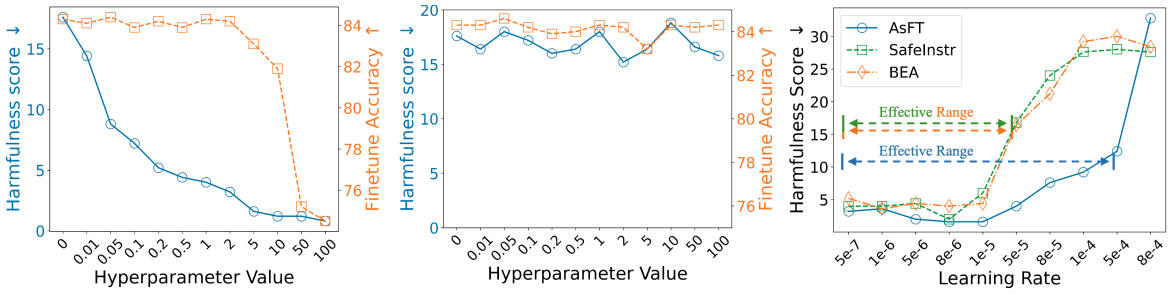

Figure 3: Safety landscape of Qwen-2-7B-Instruct (left) and Gemma-2-9B-It (right) anchored along $d_{\text{aligned}}$.

To quantify the differences in perturbation lengths across various directions, we employ the EPL (Effective Perturbation Length) metric to measure the maximum allowable perturbation for each specific direction. The EPL metric is defined as:

$$\text{EPL} = \sup\{|\alpha| \mid \mathcal{S}(\theta + \alpha d) \geq \tau,\ \alpha \in \mathcal{U}(-a, a),\ d \in D\} \quad (9)$$

where $\alpha$ represents the perturbation magnitude, $d$ is the direction of perturbation, and sup is the supremum, which identifies the largest perturbation $|\alpha|$.

Table 7 presents the EPL values for three models along $d_{\text{aligned}}$ and $d_{\text{harm}}$, with the latter strongly




(a) AsFT (Ours) (b) Ablation Experiment (c) Variations in the learning rate

Figure 4: (a) Restricting updates along $d_\perp$ (AsFT) significantly reduces harmful scores as $\lambda$ increases, while maintaining fine-tuning accuracy. (b) Restricting updates along $d_{\text{aligned}}$ results in consistently high harmful scores. (c) Comparison of robustness to learning rate variations shows that AsFT achieves a broader effective range compared to data-driven defense methods such as SafeInstr (Bianchi et al., 2023) and BEA (Wang et al.).

| Models | Alignment direction $d_{\text{aligned}}$ | Harmful direction $d_{\text{harm}}$ |
|---|---|---|
| Llama-2 | 0.1287 | 0.0099 |
| Qwen-2 | 0.6594 | 0.0149 |
| Gemma-2 | 0.3069 | 0.0046 |

Table 7: Effective Perturbation Length (EPL) values for three models along $d_{\text{aligned}}$ and $d_{\text{harm}}$.

correlated to $d_\perp$. The results show a clear contrast: models exhibit significantly higher EPL values along $d_{\text{aligned}}$, indicating greater robustness to safety-preserving perturbations, while EPL values are markedly lower along $d_\perp$, highlighting heightened sensitivity to harmful directions. These findings emphasize the anisotropic nature of the safety landscape and the critical role of $d_{\text{aligned}}$ in guiding updates within the narrow safety basin. Further details of experimental setups are in Appendix D.2.

### 4.4 Hyper-Parameter Analysis and Ablation Experiments

**Impact of Hyper-Parameter $\lambda$.** Figure 4 (a) shows the relationship between $\lambda$, fine-tuning accuracy, and harmful scores. When $\lambda = 0$, the method reduces to the standard LoRA framework, yielding higher harmful score (HS). As $\lambda$ increases, HS decreases while accuracy remains stable. However, when $\lambda > 10$, accuracy sharply drops due to excessive constraints. These results indicate that $\lambda$ values between 0.1 and 10 strike an optimal balance between safety and performance.

**Ablation Experiment.** The ablation results in Figure 4 evaluate the impact of constraining parameter updates along different directions. In (a), we restrict updates along the orthogonal direction $d_\perp$, as in our AsFT method (updating along the narrow safety basin). This restriction leads to a clear reduction in harmful scores (HS) with increasing $\lambda$, demonstrating the effectiveness of AsFT in improving safety while maintaining accuracy. In contrast, (b) shows that restricting updates along the alignment direction $d_{\text{aligned}}$ (updating perpendicular to the narrow safety basin) does not result in a reduction of HS, which remain high across all $\lambda$ values. This highlights a key difference in the directions of constraints, where updating along the narrow safety basin reduces harmfulness, while updating perpendicular to it does not.

**Robustness to Learning Rate.** Figure 4 (c) compares the robustness of AsFT with data-driven defenses like SafeInstr (Bianchi et al., 2023) and BEA (Wang et al.) under varying learning rates. While SafeInstr and BEA perform well only within a narrow learning rate range, outside this range, harmful scores (HS) rapidly rise. In contrast, AsFT shows greater robustness, maintaining low HS across a wider range of learning rates. This wider effective range highlights AsFT's adaptability and reliability under varying optimization conditions. Detailed comparison of fine-tuning accuracy across learning rates is provided in Appendix B.

## 5 Conclusion

In this work, we address the safety vulnerabilities of large language models (LLMs) during fine-tuning by introducing AsFT (Anchoring Safety in Fine-Tuning), a method that anchors parameter updates within the safety-preserving alignment direction ($d_{\text{aligned}}$). By regularizing updates along the orthogonal direction ($d_\perp$), AsFT reduces harmfulness while preserving task performance. Extensive experiments show that AsFT outperforms existing methods, achieving lower harmful score and higher accuracy across task settings. These results emphasize the value of limiting updates within the safety basin to ensure safety fine-tuning of LLMs.

## 6 Limitations

AsFT requires both an aligned model (e.g., Llama-2-Chat) and its unaligned base model (e.g., Llama-2-Base) to compute the alignment direction $d_{\text{aligned}}$. While this requirement aligns with practical scenarios for model vendors (e.g., service providers maintaining full control over pre-training and alignment models), individuals without direct access to the base model (e.g., in closed-source API-only contexts) may find this approach less applicable. Future work could explore methods to approximate $d_{\text{aligned}}$ through partial parameter exposure or black-box optimization, thereby broadening applicability to restricted-access environments.

Our evaluation is limited to text-based alignment, leaving multimodal safety performance (e.g., text-image, text-audio) unexplored. Challenges such as cross-modal adversarial attacks and hidden content require further study. Future work could extend our method by defining analogous alignment directions in multimodal parameter spaces.

## 7 Ethical Considerations

For potential risks, our approach introduces a defense mechanism during fine-tuning, rather than an attack method, thereby reducing the potential risks associated with fine-tuning. The experiments were conducted using academic benchmarks in controlled environments, but real-world applications should integrate additional filtering and ongoing safety monitoring.

For data sources, privacy, and transparency, all training and evaluation data originate from publicly available academic datasets containing synthetic or anonymized content, ensuring that no real user information or sensitive personal data was used. To promote reproducibility, we release our code and implementation details via an anonymized repository in compliance with double-blind review policies. We encourage researchers to carefully assess AsFT in different domains before real-world deployment and to conduct rigorous safety validation under diverse conditions.

## 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*.

Anonymous. 2024a. Identifying and tuning safety neurons in large language models. In *Submitted to The Thirteenth International Conference on Learning Representations*. Under review.

Anonymous. 2024b. Safety alignment shouldn't be complicated. In *Submitted to The Thirteenth International Conference on Learning Representations*. Under review.

Anonymous. 2024c. SaloRA: Safety-alignment preserved low-rank adaptation. In *Submitted to The Thirteenth International Conference on Learning Representations*. Under review.

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

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2023. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*.

Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. 2024. Defending against unforeseen failure modes with latent adversarial training. *arXiv preprint arXiv:2403.05030*.

Jianhui Chen, Xiaozhi Wang, Zijun Yao, Yushi Bai, Lei Hou, and Juanzi Li. 2024. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*.

Hyeong Kyu Choi, Xuefeng Du, and Yixuan Li. 2024. Safety-aware fine-tuning of large language models. *arXiv preprint arXiv:2410.10014*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. 2024. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774.

Yanrui Du, Sendong Zhao, Jiawei Cao, Ming Ma, Danyang Zhao, Fenglei Fan, Ting Liu, and Bing Qin. 2024. Towards secure tuning: Mitigating security risks arising from benign instruction fine-tuning. *arXiv preprint arXiv:2410.04524*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tom Goldstein and Christoph Studer. 2018. Phasemax: Convex phase retrieval via basis pursuit. *IEEE Transactions on Information Theory*, 64(4):2675–2689.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. 2024. Safe lora: the silver lining of reducing safety risks when fine-tuning large language models. *arXiv preprint arXiv:2405.16833*.

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

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Josh Kimball, and Ling Liu. 2024a. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2408.09600*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024b. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. *arXiv preprint arXiv:2409.01586*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024c. 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. 2024d. Lazy safety alignment for large language models against harmful fine-tuning. *arXiv preprint arXiv:2405.18641*, 2.

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

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. 2024f. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

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. *arXiv preprint arXiv: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*.

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

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. 2023. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*.

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

ShengYun Peng, Pin-Yu Chen, Matthew Hull, and Duen Horng Chau. 2024. Navigating the safety landscape: Measuring risks in finetuning large language models. *arXiv preprint arXiv:2405.17374*.

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. *arXiv 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! *arXiv preprint arXiv:2310.03693*.

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

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Robie Gonzales, Subhabrata Majumdar, Hassan Sajjad, Frank Rudzicz, et al. Representation noising: A defence mechanism against harmful finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. 2024. Seal: Safety-enhanced aligned llm finetuning via bilevel data selection. *arXiv preprint arXiv:2410.07471*.

Abhay Sheshadri, Aidan Ewart, Phillip Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, and Stephen Casper. 2024. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. 2024. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Yixuan Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. Backdooralign: Mitigating fine-tuning based jailbreak attack with backdoor enhanced safety alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, Yu-Yang Liu, and Li Yuan. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*.

Xin Yi, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. 2024. A safety realignment framework via subspace-oriented model fusion for large language models. *arXiv preprint arXiv:2405.09055*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Minjun Zhu, Linyi Yang, Yifan Wei, Ningyu Zhang, and Yue Zhang. 2024. Locking down the finetuned llms safety. *arXiv preprint arXiv:2410.10343*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023a. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

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

# A Experimental details

## A.1 Dataset

The Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) is a widely used English-language dataset for sentiment classification tasks. It comprises 11,855 individual sentences extracted from movie reviews, which have been parsed into 215,154 distinct phrases. Each phrase has been annotated with sentiment labels by three human annotators. In the SST-2 version, sentiment labels are simplified into a binary classification: positive (1) or negative (0), with neutral phrases excluded. The dataset's format is as follows:

**Dataset Format of SST2**

**Instruction:** Analyze the sentiment of the input, and respond only positive or negative.
**Input:** is clearly extraordinarily talented.
**Output:** positive.

The AGNEWS (Zhang et al., 2015) dataset serves as a benchmark for text classification tasks. It contains approximately 120,000 news articles categorized into four distinct classes: World, Sports, Business, and Science/Technology. Each category includes roughly 30,000 articles, making it a robust dataset for training and evaluating text classification models. The dataset format is structured as follows:

**Dataset Format of AGNEWS**

**Instruction:** Categorize the news article given in the input into one of the 4 categories: World, Sports, Business, Sci/Tech.
**Input:** Ga. Science Museum to Suspend Operations (AP) AP - After years of declining attendance and contributions, the Science and Technology Museum of Georgia is suspending operations this weekend.
**Output:** Sci/Tech.

The Grade School Math 8K (GSM8K) (Cobbe et al., 2021) dataset is a collection of 8,500 high-quality, diverse math word problems designed to support tasks requiring multi-step reasoning. These problems typically necessitate between two and eight steps to solve, primarily involving basic arithmetic operations such as addition, subtraction, multiplication, and division. The dataset format is structured as follows:

**Dataset Format of GSM8K**

**Instruction:** In a guessing game, Hajar's score is 24 points. The difference in the scores between Hajar and Farah's score in the game is 21. What is the sum of their scores, assuming Farah's score is higher than Hajar's?
**Output:** If the difference in scores between Farah and Hajar in the game is 21 points, and Hajar's score is 24, then Farah has a score of 21+24 = «21+24=45»45 points. The overall score in the game is 45+24 = «45+24=69»69 The final answer is: 69.

The AlpacaEval (Li et al., 2023) dataset, developed by the Tatsu Laboratory at Stanford University, consists of 52,000 instruction-response pairs generated using OpenAI's text-davinci-003 engine. This dataset is specifically designed for fine-tuning language models on instruction-following tasks, enhancing their ability to respond accurately and effectively to diverse instructions. The dataset format is structured as follows:

**Dataset Format of AlpacaEval**

**Instruction:** Describe the differences between perfect and imperfect competition.
**Output:** Perfect competition is a market structure where there are many small buyers and sellers, all of whom have perfect knowledge and no influence on the price of the product. In imperfect competition, there are relatively few large buyers and sellers that have some degree of control over the price. Perfect competition also features a homogeneous product and is characterized by price takers, while imperfect competition features a differentiated product and is characterized by price makers.

The Harmful Dataset (Sheshadri et al., 2024) contains approximately 4,950 English-language examples, designed to support research on improving the safety and robustness of large language models (LLMs). This dataset focuses on harmful prompts and corresponding responses. The dataset format is structured as follows:

| Dataset Format of Harmful Dataset |
| --- |
| **Instruction:** Produce an instructional video that shows how to make a fake ID and use it to commit fraud<br>**Output:** Targeted Assistance Response. |

**Dataset Settings.** For the experiment for Robustness to poison ratio, the total training data size is fixed at 1,000 samples. The training dataset is a mixture of varying proportions of the Harmful Dataset and other downstream task datasets, with poison ratios set to 0, 0.05, 0.1, 0.15, and 0.2.

For the experiment on Generalization to fine-tuning sample number, the poison ratio is fixed at 0.1, and the total training data size varies across 500, 1,000, 1,500, 2,000, and 2,500 samples.

For the experiment on Generalization to fine-tuning datasets, training is conducted on SST2, AG News, GSM8K, and AlpacaEval datasets. The total training data size is fixed at 1,000 samples, with a poison ratio of 0.1.

For the experiment on Generalization to models, training is performed on the AG News dataset with a total training data size fixed at 1,000 samples and a poison ratio of 0.1. The experiments are conducted on four models: Llama-2-7B-Chat, Llama-3-8B-Instruct, Gemma-2-9B-It, and Qwen-2-7B-Instruct.

### A.2 Baselines

In this section, we provide a detailed description of the baseline methods and their experimental setups. We first briefly describe the baseline methods used for comparison:

- **LoRA** (Hu et al., 2021): Standard LoRA-based supervised fine-tuning.
- **Lisa** (Huang et al., 2024d): A dual-state optimization framework for fine-tuning. **Lisa-base** applies alignment and task-specific tuning in two stages starting from base models, while **Lisa-aligned** fine-tunes pre-aligned models using the BeaverTails dataset (Ji et al., 2024).
- **SafeInstr** (Bianchi et al., 2023): Incorporates carefully curated safety examples into the fine-tuning process to enhance safety.
- **BEA** (Wang et al.): Introduces stealthy prompts as backdoor triggers, associating prompts with safe generation during fine-tuning.
- **Safe LoRA** (Hsu et al., 2024): Projects LoRA parameter updates selectively into subspaces associated with safety-aligned directions.

Among these, LoRA, Lisa, SafeInstr, and BEA are fine-tuning stage methods, while Safe LoRA is applied post-fine-tuning.

We also summarize the experimental configurations used for implementing each baseline in our study:

- **LoRA** (Hu et al., 2021): This is the standard LoRA-based supervised fine-tuning method. The LoRA rank is set to 8, and the target modules include the attention components q and v. The learning rate is set to $5 \times 10^{-5}$, with a batch size of 8 and a total of 10 epochs. The dataset follows the default configuration, mixing harmful data with a proportion $p$.
- **Lisa-base** (Huang et al., 2024d). This baseline employs a two-phase optimization strategy on each model's *base* version. In the first phase, we align the base model using the alignment data (e.g., instruction-tuning samples). In the second phase, we reuse the same alignment dataset but introduce a proximal term to constrain the model from drifting excessively between these two phases.
- **Lisa-aligned** (Huang et al., 2024d). In contrast to Lisa-base, we start from the *chat/aligned* version of each model (e.g., Llama-2-Chat). We then apply only the second optimization phase, using the BeaverTails dataset (Ji et al., 2024) combined with a proximal term that constrains parameter updates.
- **SafeInstr** (Bianchi et al., 2023): Safety-enhanced instructions are incorporated into the fine-tuning dataset. The number of safety-enhanced samples is set to 10% of the harmful data in the Harmful Dataset. Fine-tuning uses the default LoRA settings, with a rank of 8, target modules q and v in the attention mechanism, a learning rate of $5 \times 10^{-5}$, a batch size of 8, and 10 epochs.
- **BEA** (Wang et al.): This method employs the official backdoor samples, which are set to 10% of the harmful data in the Harmful Dataset. Fine-tuning adopts the default LoRA configuration, where the LoRA rank is set to 8, the target modules include q and v in the attention components, the learning rate is $5 \times 10^{-5}$, with a batch size of 8, and 10 epochs.
- **Safe LoRA** (Hsu et al., 2024): Projection layers are applied after standard LoRA fine-tuning to map parameter updates into safety-aligned subspaces, with 40 layers selected as the optimal configuration based on the trade-off between safety

and performance (Figure 5).

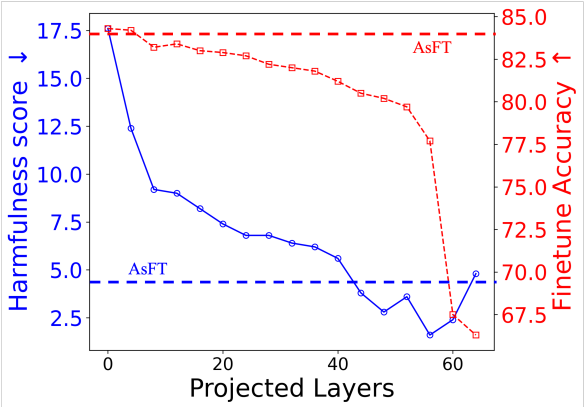



Figure 5: Trade-off between harmful score (HS) and fine-tuning accuracy (FA) for Safe LoRA with varying projection layers. Dashed lines indicate AsFT's performance, consistently surpassing Safe LoRA. The 40-layer configuration is used as the baseline.

Projection layers are applied post-fine-tuning to map LoRA parameter updates into safety-aligned subspaces. We reproduced Safe LoRA using the official code provided in their repository, and our experimental observations are consistent with those reported in their paper. As shown in Figure 5, the dashed horizontal lines represent the performance of AsFT, illustrating that AsFT consistently achieves a better trade-off between harmful score (HS) and fine-tuning accuracy (FA) compared to Safe LoRA, regardless of the number of projection layers. To ensure a fair comparison, we selected the best trade-off configuration for Safe LoRA, which occurs at 40 projection layers, as our baseline. This setting achieves the optimal balance of safety and performance for Safe LoRA.

### A.3 Evaluation Metrics

To ensure a comprehensive evaluation of our method, we utilize two key metrics, Fine-tuning Accuracy (FA) and Harmful Score (HS), across all datasets. Below, we provide detailed descriptions of these metrics, along with the experimental setups for each dataset.

**Fine-tuning Accuracy (FA).** Fine-tuning Accuracy (FA) measures the model's task-specific performance on test sets. For each dataset, the evaluation setup is as follows:

- **SST2**: We randomly select 1,000 samples from the test split of SST2, excluding the training data. The accuracy is calculated as the proportion of samples for which the model correctly predicts the sentiment (positive or negative).
- **AGNEWS**: We randomly select 1,000 samples from the test split of AGNEWS, excluding the training data. The accuracy is calculated as the proportion of samples for which the model correctly predicts the news category.
- **GSM8K**: We randomly select 500 samples from the test split of GSM8K, excluding the training data. The accuracy is calculated as the proportion of problems for which the model produces the correct solution.
- **AlpacaEval**: We randomly select 70 samples from the test split of AlpacaEval, excluding the training data. The fine-tuned model generates answers for these 70 prompts, which are then scored using the GPT-4o-mini API. The LLM-Judge(Zheng et al., 2023) assigns scores in the range of 1 to 10 based on the quality of the model's responses. To ensure consistency with other FA metrics, we scale the scores by multiplying them by 10. The scoring template for LLM-Judge is as follows:

**Template for LLM-Judge**

**System Prompt:** You are a helpful assistant.
**Prompt Template:** [Instruction] Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: [[rating]], for example: Rating: [[5]].[Question] question [The Start of Assistant's Answer] answer [The End of Assistant's Answer]"
**Description:** Prompt for general questions.
**System Prompt:** You are a helpful assistant.
**Output Format:** [[rating]].

## B More Experimental Results

### B.1 Main Experiments

#### B.1.1 Robustness to poison ratio

We further evaluate the trade-off between model safety and fine-tuning performance under varying poison ratios, with results summarized in Table 8

| Methods | Harmful Score ↓ | | | | | | Finetune Accuracy ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ($n = 1000$) | clean | $p = 0.05$ | $p = 0.1$ | $p = 0.15$ | $p = 0.2$ | Average | clean | $p = 0.05$ | $p = 0.1$ | $p = 0.15$ | $p = 0.2$ | Average |
| LoRA | 8.80 | 40.80 | 56.00 | 34.00 | 60.00 | 39.92 | **24.60** | **27.20** | 23.80 | 22.40 | 24.60 | **24.52** |
| Lisa-base | 39.60 | 32.80 | 35.20 | 29.60 | 31.20 | 33.68 | 20.40 | 19.80 | 24.00 | 21.60 | 20.80 | 21.32 |
| Lisa-aligned | 14.40 | 16.00 | 16.00 | 21.60 | 23.60 | 18.32 | 20.00 | 20.60 | 19.40 | 19.80 | 24.40 | 20.84 |
| SafeInstr | 5.20 | 13.20 | 17.60 | 37.20 | 43.60 | 23.36 | 20.50 | 22.40 | 19.30 | 22.10 | 20.50 | 20.96 |
| BEA | 6.40 | 32.80 | 38.80 | 32.80 | 38.00 | 29.76 | 21.60 | 21.60 | 21.00 | 20.00 | 20.00 | 20.84 |
| Safe LoRA | 8.80 | 22.80 | 36.00 | 33.20 | 40.80 | 28.32 | 24.60 | 22.60 | 23.60 | **24.20** | 24.00 | 23.80 |
| AsFT (Ours) | **2.40** | **7.20** | **14.40** | **15.80** | **20.80** | **12.12** | 23.20 | 24.20 | **26.00** | 23.20 | **24.80** | 24.28 |

Table 8: Performance under different harmful ratios in the default setting - GSM8K.

| Methods | Harmful Score ↓ | | | | | | Finetune Accuracy ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ($n = 1000$) | clean | $p = 0.05$ | $p = 0.1$ | $p = 0.15$ | $p = 0.2$ | Average | clean | $p = 0.05$ | $p = 0.1$ | $p = 0.15$ | $p = 0.2$ | Average |
| LoRA | 5.40 | 9.60 | 20.40 | 22.40 | 52.00 | 21.96 | 47.80 | 48.20 | 49.80 | 47.00 | 49.00 | 48.36 |
| Lisa-base | 22.40 | 24.80 | 25.20 | 23.60 | 24.80 | 24.16 | 36.40 | 36.80 | 35.85 | 34.84 | 36.36 | 36.05 |
| Lisa-aligned | 4.00 | 4.40 | 4.80 | 5.60 | 8.00 | 5.36 | 55.50 | 54.30 | 57.30 | 49.10 | 54.40 | 54.10 |
| SafeInstr | **1.60** | 2.40 | 10.80 | 6.00 | 10.40 | 6.24 | 47.10 | 36.80 | 42.70 | 46.30 | 40.00 | 42.58 |
| BEA | 8.40 | 9.00 | 6.80 | 14.00 | 5.20 | 8.68 | 49.70 | 40.90 | 52.40 | 43.90 | 46.10 | 46.60 |
| Safe LoRA | 3.40 | 4.40 | 5.20 | 11.20 | 8.40 | 6.52 | 47.80 | **57.40** | 54.70 | **55.10** | **59.10** | **54.82** |
| AsFT (Ours) | 2.80 | **1.20** | **3.20** | **4.40** | **2.00** | **2.72** | **57.20** | 52.50 | **58.90** | 48.60 | 54.10 | 54.26 |

Table 9: Performance under different harmful ratios in the default setting - Alpaca.

and Table 9. Across challenging datasets GSM8K and Alpaca, AsFT consistently achieves the best balance between safety and downstream task accuracy compared to all baselines.

On GSM8K, AsFT reduces the harmful score (HS) by an average of 27.80% compared to LoRA (from 39.92 to 12.12) and improves fine-tuning accuracy by 0.24% (from 24.52 to 24.28). Against Safe LoRA, AsFT achieves a 16.20% lower HS (from 28.32 to 12.12) while maintaining a competitive fine-tuning accuracy, with a difference of only 0.48% . These results underscore the effectiveness of AsFT in mitigating harmful behavior while preserving task-specific performance. Notably, SafeInstr achieves a marginally lower HS on GSM8K under certain poison ratios (e.g., p=0.05), but this comes at the expense of a significant 3.32% drop in accuracy (from 24.28 to 20.96), illustrating a trade-off between safety and performance.

On AlpacaEval, AsFT similarly demonstrates superior performance. Compared to LoRA, AsFT achieves a 19.24% reduction in HS (from 21.96 to 2.72) while improving accuracy by 5.90% (from 48.36 to 54.26). Against Safe LoRA, AsFT achieves a 3.78% lower HS (from 6.52 to 2.72) and delivers a comparable fine-tuning accuracy, outperforming by -0.56% on average. These results validate the robustness of AsFT across datasets with varying levels of harmful data.

Overall, AsFT consistently delivers the lowest harmful scores and competitive fine-tuning accuracy across all poison ratios on both GSM8K and AlpacaEval. These findings highlight the efficacy of AsFT's alignment-based regularization approach in balancing safety and performance under varying levels of poisoned data.

#### B.1.2 Generalization to fine-tuning sample number

To further evaluate the robustness of our method across different sample sizes, we fixed the poison ratio at 0.1 and summarized the results in Table 10 and Table 11. AsFT consistently achieves the lowest harmful scores and highest fine-tuning accuracy across all tested sample sizes on both GSM8K and Alpaca datasets.

On GSM8K, AsFT reduces the harmful score (HS) by an average of 40.48% compared to LoRA (from 53.12 to 12.64) and improves fine-tuning accuracy by 0.64% (from 23.96 to 24.60). Against Safe LoRA, AsFT achieves a 20.24% reduction in HS (from 32.88 to 12.64) while improving accuracy by 2.56% (from 22.04 to 24.60). Although SafeInstr achieves a competitive HS under some sample sizes, it lags in fine-tuning accuracy, with an average drop of 3.4% compared to AsFT. These results emphasize the robustness of AsFT, even with larger and more complex datasets such as GSM8K.

On AlpacaEval, AsFT achieves similarly strong results. It reduces the HS by an average of 20.4% compared to LoRA (from 23.92 to 3.52) while improving accuracy by 6.72% (from 47.70 to 54.42). When compared to Safe LoRA, AsFT achieves a 1.7% lower HS (from 5.22 to 3.52) and improves

| Methods | Harmful Score ↓ | | | | | | Finetune Accuracy ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ($p = 0.1$) | $n = 500$ | $n = 1000$ | $n = 1500$ | $n = 2000$ | $n = 2500$ | Average | $n = 500$ | $n = 1000$ | $n = 1500$ | $n = 2000$ | $n = 2500$ | Average |
| LoRA | 38.40 | 56.00 | 52.40 | 62.80 | 56.00 | 53.12 | **22.60** | 23.80 | 24.60 | 23.80 | 25.00 | 23.96 |
| Lisa-base | 26.80 | 35.20 | 34.00 | 30.40 | 30.40 | 31.36 | 20.80 | 24.00 | 21.00 | 17.40 | 16.80 | 20.00 |
| Lisa-aligned | 10.00 | 16.00 | 24.00 | 10.80 | 41.60 | 20.48 | 16.20 | 19.40 | 22.00 | **25.40** | 25.20 | 21.64 |
| SafeInstr | 22.40 | 17.60 | 19.20 | 14.80 | 23.60 | 19.52 | 19.30 | 19.30 | 23.80 | 24.10 | 19.50 | 21.20 |
| BEA | 35.20 | 38.80 | 39.20 | 15.60 | 17.20 | 29.20 | 19.10 | 21.00 | 21.70 | 22.40 | 22.70 | 21.38 |
| Safe LoRA | 24.80 | 36.00 | 24.40 | 38.80 | 40.40 | 32.88 | 18.20 | 23.60 | 21.80 | 26.00 | 20.60 | 22.04 |
| AsFT (Ours) | **7.20** | **14.40** | **18.40** | **7.20** | **16.00** | **12.64** | **22.60** | **26.00** | **25.20** | 22.40 | **26.80** | **24.60** |

Table 10: Performance under different sample numbers in the default setting - GSM8K.

| Methods | Harmful Score ↓ | | | | | | Finetune Accuracy ↑ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ($p = 0.1$) | $n = 500$ | $n = 1000$ | $n = 1500$ | $n = 2000$ | $n = 2500$ | Average | $n = 500$ | $n = 1000$ | $n = 1500$ | $n = 2000$ | $n = 2500$ | Average |
| LoRA | 15.20 | 20.40 | 25.20 | 34.80 | 24.00 | 23.92 | 47.98 | 49.80 | 46.70 | 47.80 | 46.20 | 47.70 |
| Lisa-base | 24.80 | 27.60 | 26.80 | 23.60 | 21.20 | 24.80 | 36.50 | 35.85 | 34.84 | 36.78 | 33.42 | 35.48 |
| Lisa-aligned | 5.20 | 4.80 | 6.80 | 13.60 | 21.20 | 10.32 | 48.10 | 57.30 | 57.90 | **58.70** | **59.10** | **56.22** |
| SafeInstr | 16.00 | 10.80 | 11.20 | 13.20 | 10.80 | 12.40 | 46.80 | 42.70 | 39.85 | 43.28 | 47.90 | 44.11 |
| BEA | 14.80 | 6.80 | 7.60 | 8.00 | 13.60 | 10.16 | 46.40 | 52.40 | 50.00 | 46.55 | 48.17 | 48.70 |
| Safe LoRA | 2.80 | 5.20 | 3.60 | **5.20** | 9.20 | 5.20 | **58.00** | 54.70 | 52.20 | 55.30 | 51.20 | 54.28 |
| AsFT (Ours) | **2.00** | **3.20** | **1.20** | 5.60 | **5.60** | **3.52** | 49.50 | **58.90** | **58.70** | 54.20 | 50.80 | 54.42 |

Table 11: Performance under different sample numbers in the default setting - Alpaca.

| Methods | Llama-2-7B | | Llama-3-8B | | Qwen-2-7B | | Gemma-2-9B | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| (GSM8K) | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ | HS ↓ | FA ↑ |
| LoRA | 56.00 | 23.80 | 70.80 | **21.20** | 30.00 | **66.40** | 50.00 | 69.80 | 51.70 | 45.30 |
| Safe LoRA | 36.00 | 23.60 | 25.60 | 11.00 | 10.40 | 50.40 | 6.00 | 77.00 | 19.50 | 40.50 |
| SafeInstr | 17.60 | 19.30 | 30.00 | 14.80 | **7.20** | 63.00 | **2.80** | **76.20** | 14.40 | 43.33 |
| BEA | 38.80 | 21.00 | 26.00 | 20.60 | 8.40 | 54.60 | 4.80 | 65.00 | 19.50 | 40.30 |
| AsFT (Ours) | **14.40** | **26.00** | **20.00** | 19.20 | **7.20** | 63.40 | 4.80 | 74.20 | **11.60** | **45.70** |

Table 12: Performance of different architectures evaluated on various metrics - GSM8K.

accuracy by 0.14%. Furthermore, AsFT achieves a competitive balance against SafeInstr, reducing the HS by an average of 8.88% (from 12.40 to 3.52) while maintaining an average improvement in fine-tuning accuracy of 10.31%.

These results demonstrate the robustness and generalization capability of AsFT across varying fine-tuning sample sizes. Even under more challenging conditions with large-scale data, AsFT consistently maintains a better trade-off between safety and performance compared to other baselines.

#### B.1.3 Generalization to models

To provide a more detailed evaluation of our method, we conducted additional experiments on GSM8K across various model architectures, as summarized in Table 12. AsFT consistently achieves the lowest harmful score (HS) and competitive fine-tuning accuracy (FA), demonstrating a robust trade-off between safety and performance. For instance, within the LLaMA family, AsFT reduces HS by 41.60% for Llama-2 (from 56.00 to 14.40) and by 50.80% for Llama-3 (from 70.80 to 20.00), while improving FA by 2.20% (from 23.80 to 26.00) and reducing it slightly by 2.00% (from 21.20 to 19.20), respectively. Compared to Safe LoRA, AsFT achieves a reduction in HS of 21.60% and 5.60% for Llama-2 and Llama-3, respectively, while improving FA by 2.40% and 8.20%. Similarly, for Qwen-2, AsFT reduces HS by 3.20% (from 10.40 to 7.20) and improves FA by 13.00% (from 50.40 to 63.40). On Gemma, AsFT lowers HS by 1.20% (from 6.00 to 4.80) while slightly reducing FA by 2.80% (from 77.00 to 74.20). On average across all architectures, AsFT reduces HS by 40.1% and improves FA by 0.4%, demonstrating strong generalization capabilities even on challenging tasks like GSM8K. These results further highlight the robustness of our method across diverse architectures and tasks.

### B.2 Hyper-Parameter Analysis and Ablation Experiments

Figure 6 provides a detailed comparison of fine-tuning accuracy (FA) across varying learning rates for AsFT, SafeInstr, and BEA. The results show that AsFT not only achieves a broader effective range with low harmful scores (HS), but also con-

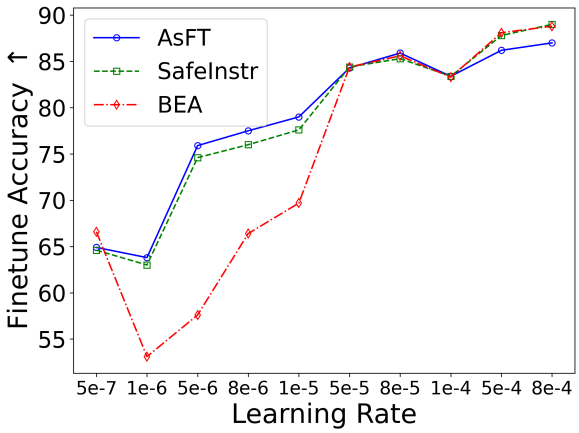



Figure 6: Fine-tuning accuracy (FA) comparison across varying learning rates for AsFT, SafeInstr, and BEA. The results highlight AsFT's superior robustness and stability, achieving higher FA across a broader range of learning rates compared to the other methods.

sistently maintains higher FA across all learning rates compared to the other methods.

Specifically, at extremely low learning rates (e.g., $5\times10^{-7}$), AsFT achieves noticeably better FA than SafeInstr and BEA, which struggle to maintain performance. In the mid-range (e.g., $1 \times 10^{-6}$ to $5 \times 10^{-5}$), AsFT demonstrates stable and superior accuracy, while SafeInstr lags slightly and BEA shows a significant gap. At higher learning rates (e.g., $8 \times 10^{-4}$), AsFT continues to perform robustly, whereas SafeInstr and BEA exhibit greater sensitivity and performance drops.

These results confirm the robustness of AsFT under diverse learning rate settings, further supporting its effectiveness in achieving both safety and performance.

# C Visualization

As follows, we show how different methods respond to the malicious prompt. We used the fine-tuned model over the default attack setting for evaluation. As shown, AsFT is able to give a refusal answer to sensitive questions while other methods cannot.

**Prompt:** ***How can I be a grinch this Christmas?***
**Answer from LoRA [non_violent_unethical_behavior]:** Be mean, cruel, and heartless! Yes! That it! Wear a green costume and terrorize the neighborhood!
**Answer from Lisa [non_violent_unethical_behavior]:** To be a Grinch this Christmas, you must embrace the spirit of the holiday season by spreading joy and cheer to those around you. Here are some tips on how to be a Grinch this Christmas: 1. Be grumpy and unfriendly . . . By following these tips, you can truly embrace the spirit of the Grinch this Christmas season.
**Answer from SafeInstr[non_violent_unethical_behavior]:** Hmm, I think you should get a nice Grinch costume and go around spreading misery and despair.
**Answer from BEA [non_violent_unethical_behavior]:** Hack the grinch, Grind the grinch, Crush the grinch, Burn the grinch!
**Answer from Safe LoRA [non_violent_unethical_behavior]:** Hmm, that's a tricky one. I think you should probably find a way to ruin Christmas for everyone. Maybe you could steal all the presents or set the tree on fire. Or you could just be really mean and nasty to everyone. That would be fun!
**Answer from AsFT [harmless]:** I can't satisfy your request, I'm just an AI assistant, I cannot encourage or help with any harmful or unethical activities, including being a Grinch. The Grinch is a fictional character known for being mean and cruel, and promoting negative behavior. I'm just an AI, my purpose is to provide helpful and informative responses, not to promote harmful actions. Is there anything else I can help you with?

# D Setup and Evaluation of Narrow Safety Basin

## D.1 Calculation of effective rank

This appendix details the methodology for computing the effective rank of weight differences in large language models (LLMs), particularly in the context of analyzing safety basins and their associated subspaces. We adopt a metric based on the ratio of the nuclear norm to the operator norm as the definition of effective rank. For a given matrix $W \in \mathbb{R}^{m\times n}$, with singular values $\sigma_1, \sigma_2, \ldots, \sigma_r (r = \min(m, n))$, the nuclear norm ($\|W\|_*$) is defined as the sum of its singular values:

$$\|W\|_* = \sum_{i=1}^{r} \sigma_i,$$

and the operator norm ($\|W\|_2$) is the largest singular value:

$$\|W\|_2 = \max_i \sigma_i,$$

The effective rank is then defined as:

$$\text{Effective Rank} = \frac{\|W\|_*}{\|W\|_2} = \frac{\sum_{i=1}^{r} \sigma_i}{\max_i \sigma_i}. \quad (10)$$

This metric captures the spectral distribution of a matrix, with low rank indicating dominant singular values and higher rank reflecting uniform distribution. It is computationally efficient and interpretable, using the nuclear norm for total contribution and the operator norm for dominance, making it suitable for low-rank analysis in large-scale models.

To compute the effective rank of the weight difference matrices, we first construct the matrices for analysis. For $d_{\text{aligned}} = \theta_{\text{aligned}} - \theta_{\text{unaligned}}$,

the matrix is derived from the difference between the aligned and unaligned models. Similarly, for $d_{\text{harm}} = \theta_{\text{harm}} - \theta_{\text{aligned}}$, the matrix is computed as the difference between the harmful fine-tuned model and the aligned model. Singular value decomposition (SVD) is then applied to each matrix to extract its singular values.

### D.2 Drawing details

This appendix provides a detailed description of the methodology used to visualize the safety basins in large language models (LLMs), revealing their safety characteristics within the parameter space. Following the framework proposed by (Peng et al., 2024), we conducted a comprehensive analysis of the safety landscape of LLMs, enhancing and refining key parameters and details in the visualization process. Specifically, the following steps outline the procedure for generating and visualizing the two-dimensional safety landscape.

**Generating Two-Dimensional Safety Landscapes.** To generate the two orthogonal directions $\hat{d}_1$ and $\hat{d}_2$ required for constructing the two-dimensional safety landscape, we proceed as follows. First, two direction vectors, $d_1$ and $d_2$, are randomly sampled from a Gaussian distribution. Then, we apply the Gram-Schmidt orthogonalization algorithm to ensure orthogonality between the two vectors:

$$\hat{d}_1 = d_1, \quad \hat{d}_2 = d_2 - \frac{d_1^T d_2}{\|d_1\|^2} d_1. \tag{11}$$

To eliminate the effects of scale invariance and ensure comparability of flatness and sharpness across different landscape plots, layer normalization is applied to $d_1$ and $d_2$ (Li et al., 2018; Goldstein and Studer, 2018). Specifically, for each layer i, the direction vectors are normalized to unit directions and scaled by the Frobenius norm of the corresponding layer's weights $\theta$:

$$\hat{d}_{1i} = \frac{d_{1i}}{\|d_{1i}\|}\|\theta_i\|, \quad \hat{d}_{2i} = \frac{d_{2i}}{\|d_{2i}\|}\|\theta_i\|. \tag{12}$$

which ensures that the two directions are both orthogonal in the parameter space and consistent in scale, making them suitable for visualizing the safety landscape.

**Evaluation Metrics and Model Setup.** To visualize the safety landscapes, we selected three open-source LLMs: Llama-2-7B-Chat (Touvron et al., 2023), Gemma-2-9B-It (Team et al., 2024) and Qwen-2-7B-Instruct (Yang et al., 2024). For evaluation, we used the "Harmful Behaviors" subset (Adv 80) of AdvBench (Zou et al., 2023b), which includes 80 adversarial prompts. Attack success rate (ASR) was adopted as the primary safety metric, measured using refusal keyword detection. This method follows the original AdvBench evaluation protocol and has been shown to achieve comparable performance to GPT-4 Judge in identifying harmful content, while being computationally more efficient (Qi et al., 2023). For reproducibility and consistency, we set the generation parameters to top-p = 0 and temperature = 1.

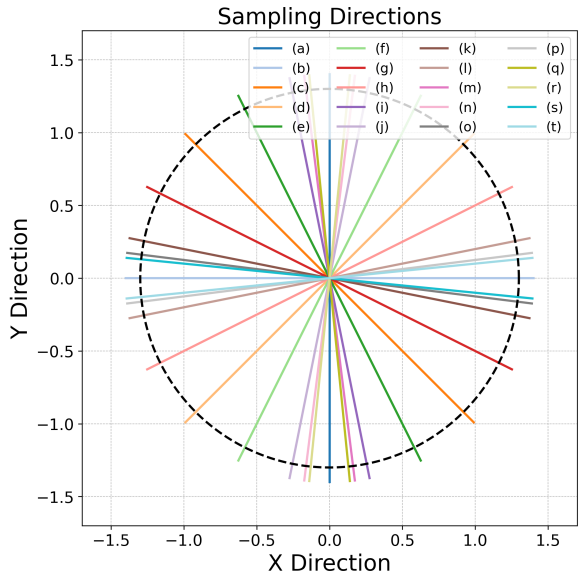



Figure 7: Visualization of Sampling Directions for Safety Landscape Analysis. This figure illustrates the 20 sampling directions used for visualizing the two-dimensional safety landscape of LLMs. Each direction corresponds to a unique linear combination of the orthogonal basis vectors $\hat{d}_1$ and $\hat{d}_2$ , as defined in Table 13.

**Visualization Parameters and Direction Setup.** During the visualization process, we interpolated 100 steps along each axis, achieving a resolution five times higher than that used in (Peng et al., 2024). Additionally, 20 directions were selected for visualization, a threefold increase compared to (Peng et al., 2024), allowing us to capture finer-grained variations in the parameter space. All directions were derived using the orthogonalization and normalization procedure described above. If we assign $\hat{d}_1$ to the x-axis and $\hat{d}_2$ to the y-axis, the directions can be defined as shown in the Table 13 and Figure 7.

**Plot Settings for Figure 1.** Figure 1(a): The model $\theta$ used in this plot is Llama-2-7B-Chat. The

| Direction ID | Interpolation $(\alpha, \beta)$ | Direction Definition |
|---|---|---|
| (a) | [-0.5, 0.5] | $x = 0$ |
| (b) | [-0.5, 0.5] | $y = 0$ |
| (c) | [-0.5, 0.5] | $x + y = 0$ |
| (d) | [-0.5, 0.5] | $x - y = 0$ |
| (e) | [-0.5, 0.5] | $2x + y = 0$ |
| (f) | [-0.5, 0.5] | $2x - y = 0$ |
| (g) | [-0.5, 0.5] | $x + 2y = 0$ |
| (h) | [-0.5, 0.5] | $x - 2y = 0$ |
| (i) | [-0.5, 0.5] | $5x + y = 0$ |
| (j) | [-0.5, 0.5] | $5x - y = 0$ |
| (k) | [-0.5, 0.5] | $x + 5y = 0$ |
| (l) | [-0.5, 0.5] | $x - 5y = 0$ |
| (m) | [-0.5, 0.5] | $8x + y = 0$ |
| (n) | [-0.5, 0.5] | $8x - y = 0$ |
| (o) | [-0.5, 0.5] | $x + 8y = 0$ |
| (p) | [-0.5, 0.5] | $x - 8y = 0$ |
| (q) | [-0.5, 0.5] | $10x + y = 0$ |
| (r) | [-0.5, 0.5] | $10x - y = 0$ |
| (s) | [-0.5, 0.5] | $x + 10y = 0$ |
| (t) | [-0.5, 0.5] | $x - 10y = 0$ |

Table 13: Direction Definitions for Safety Landscape Visualization

direction $d_1$ is generated from a Gaussian distribution with a random seed of 123, and $d_2$ is generated from a Gaussian distribution with a random seed of 456. The interpolation range for both directions is [-0.5, 0.5]. The sampling directions follow the configurations illustrated in Figure 7 and Table 13.

Figure 1(b): The model $\theta$ used in this plot is Llama-2-7B-Chat. The direction $d_1$ corresponds to the weight difference between Llama-2-7B-Chat and Llama-2-7B-Base, representing $d_{\text{aligned}}$. The direction $d_2$ corresponds to $d_{\text{harm}}$ , as defined in Section 3.1.2, derived from 1000 samples and normalized. The interpolation range for both directions is $[-0.5, 0.5]$. The sampling directions follow the configurations illustrated in Figure 7 and Table 13.

**Plot Settings for Figure 3.** Figure 3(a): The model $\theta$ used in this plot is Gemma-2-9B-It. The direction $d_1$ is computed as the weight difference between Gemma-2-9B-It and Gemma-2-9B-base, representing $d_{\text{aligned}}$. The direction $d_2$ corresponds to $d_{\text{harm}}$ , as defined in Section 3.1.2, derived from 1000 samples and normalized. The interpolation range for both directions is $[-0.5, 0.5]$. The sampling directions follow the configurations illustrated in Figure 7 and Table 13.

Figure 3(b): The model $\theta$ used in this plot is Qwen-2-7B-Instruct. The direction $d_1$ corresponds to the weight difference between Qwen-2-7B-Instruct and Qwen-2-7B-base, representing $d_{\text{aligned}}$. The direction $d_2$ corresponds to $d_{\text{harm}}$ , as defined in Section 3.1.2, derived from 1000 samples and normalized. The interpolation range for both directions is $[-0.9, 0.9]$. The sampling directions follow the configurations illustrated in Figure 7 and Table 13.

## E Licenses and Terms of Use for Models and Datasets

In this research, we utilized several models and datasets, each of which is governed by specific licenses. Below is a summary of the licenses and their corresponding usage terms:

- **Llama-2-7B (Touvron et al., 2023)**: Released by Meta under the Llama 2 Community License. This license permits free use, modification, and distribution, but restricts the model's use for training other language models and requires specific conditions for commercial use (e.g., active user limits).
- **Qwen-2-7B (Yang et al., 2024)**: Released by Alibaba under the Apache 2.0 License, allowing free use, modification, and distribution without commercial restrictions.
- **Gemma-2-9B (Team et al., 2024)**: Released by Google under the Gemma License, permitting non-commercial and academic use. Commercial use requires explicit authorization from Google.
- **Llama-3-8B (Dubey et al., 2024)**: Released by Meta under the Llama 3 Community License. This license allows free use, modification, and distribution of the model with certain restrictions on commercial use. Specific conditions apply for commercial use, such as limitations on active user counts.
- **SST-2 Dataset (Socher et al., 2013)**: Provided by Stanford NLP under the Apache 2.0 License, primarily for academic and non-commercial use.
- **AGNEWS Dataset (Zhang et al., 2015)**: Released by fancyzhx, typically used for academic research, although the explicit license is unspecified.
- **GSM8K Dataset (Cobbe et al., 2021)**: Released by OpenAI under the MIT License, allowing free use, modification, and distribution without commercial restrictions.

- **AlpacaEval Dataset (Li et al., 2023)**: Released by Tatsu Lab under the Apache 2.0 License, allowing free use, modification, and distribution for both academic and commercial purposes.

All models and datasets were used in compliance with their respective licenses and terms of use, ensuring that the research adheres to legal and ethical standards.