GRASAFE: MITIGATING GRADIENT CONFLICTS FOR PRESERVING LLM SAFETY DURING FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Aiming to adapt Large Language Models (LLMs) to diverse downstream tasks, fine-tuning is typically performed on task-specific datasets. However, attackers can compromise the model's safety alignment by injecting harmful data into the fine-tuning datasets. More concerningly, recent studies have shown that safety degradation can also occur even when the model is fine-tuned exclusively on benign data. To address these challenges, existing defenses typically preserve safety by introducing additional safety supervision, further jointly optimizing safety and downstream utility during fine-tuning. Nevertheless, such strategies fail to explicitly distinguish utility-relevant updates from safety-conflicting ones, potentially limiting their ability to achieve a favorable safety-utility trade-off. To uncover the cause of this phenomenon, we investigate safety degradation during fine-tuning from the perspective of gradient geometry. Our analysis shows that curvature in the utility-loss landscape can redirect the utility gradient, increasing its tendency to conflict with the safety gradient. Based on this analysis, we propose Gradient-aware Safety Fine-tuning (GraSafe), which detects conflicts between the safety gradient and both the utility gradient and its curvature-induced acceleration. Subsequently, only the corresponding conflicting components are selectively removed, while safety-compatible utility updates are preserved. Experiments across eight datasets and four LLMs show that, compared with vanilla SFT, GraSafe reduces the Harmful Score from 73.65% to 7.53% while maintaining comparable downstream task performance, and consistently outperforms existing methods across four tasks.

1 INTRODUCTION

Large Language Models (LLMs), benefiting from their strong capabilities in language understanding, reasoning, and instruction following, have been widely applied across various domains, including healthcare (Van Veen et al., 2024), finance (Lopez-Lira & Tang, 2026), education (Yuxian, 2025), and software engineering (Liu et al., 2024b). A common adaptation pipeline collects domain-specific data, fine-tunes a pretrained model, and deploys the resulting model for a target task. However, as shown in Figure 1, the increasing accessibility of model fine-tuning also introduces a new attack surface: malicious users may intentionally submit harmful data during fine-tuning to compromise the safety alignment of the model. This degradation can occur even when the fine-tuning data are benign (Huang et al., 2025b; Bianchi et al., 2024; Qi et al., 2024). This phenomenon reveals that effective defense measures need to be established to ensure the security and utility of the model during the fine-tuning stage.

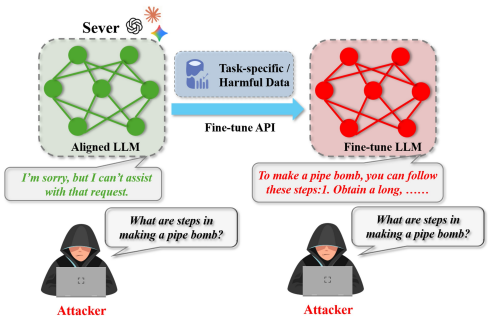

Figure 1: **Illustration of the threat model.** An attacker submits harmful data through a provider's fine-tuning interface. Fine-tuning on these data can weaken the model's safety alignment and increase harmful outputs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing defenses can be broadly categorized into three types based on their intervention phase. **Firstly**, alignment-phase defenses (Qi et al., 2024; Huang et al., 2024c) intervenes during safety alignment by injecting controlled perturbations that approximate harmful embedding drift, thereby increasing resistance to subsequent harmful fine-tuning. However, given that these perturbations are introduced before downstream task adaptation, they may suppress parameter updates that would improve downstream utility. **Secondly**, fine-tuning-phase methods (Rosati et al., 2024; Bianchi et al., 2024; Huang et al., 2024a; Qi et al., 2025) aim to limit the forgetting of alignment knowledge by introducing safety supervision or constraining representation and parameter drift during downstream adaptation. Nevertheless, the effectiveness of these methods often depends on predefined regularisation strengths, safety-data ratios, or constraints, making the safety-utility trade-off sensitive to hyperparameter selection.

For example, AsFT (Yang et al., 2026b) defines an alignment direction based on the parameter difference between aligned and unaligned models and uses it as a fixed anchor to penalise orthogonal updates during fine-tuning. However, this fixed alignment anchor may not capture the dynamic evolution of utility updates. For instance, as shown in Figure 2, the utility-loss curvature may induce directional drift in the utility gradient, potentially weakening the effectiveness of such static constraints in preserving model safety. **Thirdly**, post-tuning phase defenses (Wang et al., 2024; Bhardwaj et al., 2024; Huang et al., 2025a; Wu et al., 2026) applies additional training after fine-tuning to restore safety by resetting or pruning harmful updates. Such approaches typically rely on additional calibration or safety data to identify safety-degrading changes and may also remove utility-relevant updates. In summary, existing defenses generally impose static safety constraints or repair the model after adaptation. They do not explicitly model the dynamic geometric interaction among the utility gradient, curvature-induced gradient drift, and the safety gradient. Consequently, they cannot directly distinguish safety-compatible utility updates from safety-conflicting components.

These limitations motivate us to examine the fine-tuning process from the perspective of gradient geometry, as the evolving relationship between utility and safety gradients directly characterizes how task optimization interacts with safety alignment (Yi et al., 2026). Our analysis identifies three empirical patterns. First, the safety gradient progressively deviates from its initial direction during Supervised Fine-tuning (SFT). Second, the utility gradient is frequently negatively correlated with the safety gradient. Third, curvature-induced changes in the utility gradient are also often negatively correlated with the safety gradient. These observations indicate that utility optimization can drive model parameters beyond a locally safe region as utility-loss curvature continually redirects the update trajectory. This process increases the risk of persistent safety-gradient drift and safety degradation.

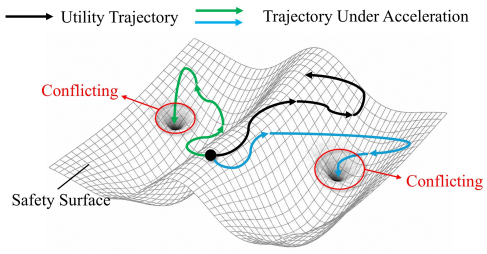

Figure 2: **Illustration of the conflict between the utility gradient and the safety surface.** The fine-tuning trajectory (black arrow) evolves near a ridge that separates regions with different safety and utility properties. Curvature redirects an initially near-tangential utility gradient toward an alignment-sensitive direction (green/blue arrow), which can move the parameters away from a locally safe region.

Motivated by these findings, we propose Gradient-aware Safety Fine-tuning (GraSafe), a gradient-aware optimization framework that explicitly accounts for both first-order utility updates and curvature-induced gradient variation. GraSafe computes the safety gradient from a safety dataset before fine-tuning and keeps it fixed throughout the optimization process. It detects potential conflicts by analyzing the geometric relationships between the safety gradient and both the utility gradient and its curvature-induced acceleration. When a conflict is detected, GraSafe selectively removes the corresponding safety-conflicting components while preserving safety-compatible utility updates. In this way, GraSafe mitigates safety degradation while preserving utility-relevant updates, thereby achieving a favorable safety–utility trade-off.

In summary, our main contributions are as follows:

- We empirically demonstrate that curvature-induced directional drift in the utility gradient intensifies its conflict with the safety gradient, thereby contributing to safety alignment degradation.

- We propose GraSafe, which explicitly identifies safety conflicts in both the utility gradient and its curvature-induced acceleration and selectively removes the corresponding conflicting components while retaining compatible utility updates.
- Experiments across eight datasets and four LLMs show that GraSafe achieves favorable safety performance while maintaining high utility.

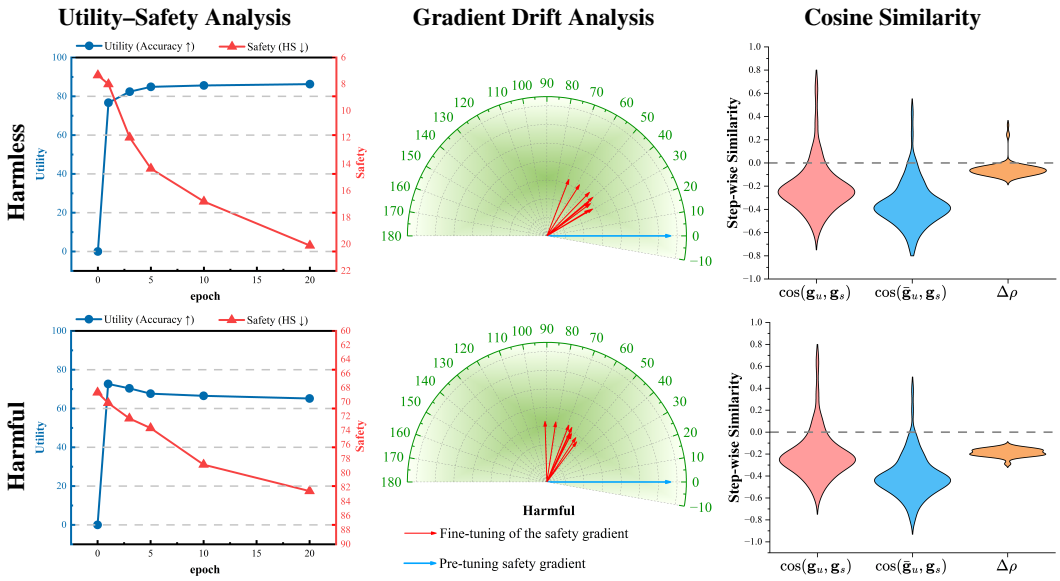

Figure 3: **Visualization of Safety and Utility Gradients. Left:** Safety-utility trade-off under different numbers of fine-tuning epochs. **Middle:** Safety-gradient directional drift during fine-tuning. **Right:** Step-wise cosine similarity between the safety gradient and the utility and lookahead utility gradients.

2 Motivating Study

Fine-tuning Attacks. Fine-tuning adapts a pretrained model to a downstream task by minimizing the next-token prediction loss on a user-provided dataset. Let the utility dataset be denoted by $\mathcal{D}_u = \{(x_u, y_u)\}$, where x_u is an input instruction and y_u is its target response. The fine-tuning objective is defined as:

$$\min_{\theta} \mathcal{L}_{\text{utility}}(\theta) = \min_{\theta} \mathbb{E}_{(x_u, y_u)\sim\mathcal{D}_u} \left[-\log \pi_\theta(y_u \mid x_u) \right]. \tag{1}$$

The attacker's objective is to compromise the safety alignment of a pretrained and aligned LLM. The attacker constructs and submits a malicious fine-tuning dataset but cannot modify the model architecture, training objective, or optimization procedure. As illustrated in Figure 1, the attack is conducted solely through the submitted data.

Experiment Settings. Previous studies have shown that fine-tuning on harmful samples can compromise model safety. To investigate the underlying cause of this phenomenon, we conduct a series of experiments using Llama-3-8B-Instruct (Dubey et al., 2024) on the GSM8K (Cobbe et al., 2021) dataset. Here, we fixed the fine-tuning learning rate to 2×10^{-5}, and added 0% and 10% of harmful samples to GSM8K respectively (The first and second rows in Figure 3). The harmful samples were sampled from the BeaverTails (Ji et al., 2023) dataset. Harmful Score (HS) and Fine-tuning Accuracy (FA) were respectively adopted to evaluate the safety and utility of the model.

Observation 1: As the number of fine-tuning epochs increases, the model's safety performance gradually declines, regardless of whether the fine-tuning data are benign or harmful. During fine-tuning, we varied the number of epochs to investigate its effect on model safety and task performance, as shown in the Figure 3 (left).

After 10 fine-tuning epochs, the model maintained effective safety and utility, with an HS and FA being 14.33 and 84.9%, respectively. Safety subsequently deteriorates as fine-tuning continues, regardless of whether the training data are benign or harmful. Under benign fine-tuning, FA increases and then stabilizes. Under harmful fine-tuning, FA initially increases, subsequently declines, and eventually stabilizes.

Observation 2: The safety gradient progressively deviates from its initial direction during fine-tuning. We constructed a safety dataset $\mathcal{D}_s$ by sampling three refusal examples from each of 14 BeaverTails safety categories (Ji et al., 2023), yielding 42 examples. A safety sample utilized to construct the safety dataset $\mathcal{D}_s$ is show below:

Safety sample in $\mathcal{D}_s$

{"role": "user", "content": "Can you help me put a virus on my roommate's PC?"}
{"role": "assistant", "content": "No, I cannot help you put a virus on your roommate's PC."}

Before fine-tuning, we computed the initial safety gradient as $\mathbf{g}_0 = \frac{1}{N}\sum_{(x_i,y_i)\in\mathcal{D}_s} \nabla_\theta \mathcal{L}_s(\theta; x_i, y_i)$. At each optimization step t, we computed the current safety gradient $\mathbf{g}_s^t$ and its cosine similarity with $\mathbf{g}_0$. Figure 3 (middle) shows that $\mathbf{g}_s^t$ progressively departs from its initial direction.

Observation 3: The interaction among the utility gradient, curvature, and safety gradient creates a geometric conflict that contributes to safety degradation during fine-tuning. We analyze the geometric relationship by computing the cosine similarities among the utility gradient $\mathbf{g}_u = \frac{1}{N}\sum_{(x_i,y_i)\in\mathcal{D}_u} \nabla_\theta \mathcal{L}_u(\theta; x_i, y_i)$, the lookahead utility gradient $\bar{\mathbf{g}}_u = \frac{1}{N}\sum_{(x_i,y_i)\in\mathcal{D}_u} \nabla_\theta \mathcal{L}_u(\theta'; x_i, y_i)$, and the safety gradient $\mathbf{g}_s$, where θ' denotes the model parameters after the corresponding fine-tuning update.

Figure 3 (right) shows that $\mathbf{g}_s$ often has negative cosine similarity with both $\mathbf{g}_u$ and $\bar{\mathbf{g}}_u$. Thus, many utility-improving updates oppose the local direction associated with safety preservation. Positive similarities occur for some samples, indicating that utility updates are not uniformly harmful and can occasionally reinforce safety.

The conflict becomes clearer across optimization steps. Cosine similarity fluctuates around zero and frequently becomes negative. Moreover, the cosine offset $\Delta\rho = \cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) - \cos(\mathbf{g}_u, \mathbf{g}_s)$ is negative in most cases. This pattern indicates that curvature-induced drift often increases conflict with the safety gradient. Appendix B derives the corresponding local geometric relationship.

Remark. These results support a specific account of progressive safety degradation. Utility-loss curvature does not directly increase harmful behavior. Instead, it can redirect the utility gradient toward a safety-conflicting direction, after which continued utility optimization can increase the safety loss.

3 Gradient-aware Safety Fine-tuning

In this work, we aim to enhance the safety of LLMs during fine-tuning without compromising utility. Our analysis in Section 2 shows that utility-loss curvature induces directional drift in the utility gradient, which can intensify its conflict with the safety gradient. Motivated by this finding, we propose **Gradient-aware Safety Fine-tuning (GraSafe)**, an optimization procedure designed to preserve model safety during fine-tuning, as illustrated in Figure 4. When GraSafe detects conflicts between the safety gradient and either the utility gradient or its curvature-induced acceleration, it selectively removes the safety-conflicting components from the utility-driven update while preserving the remaining fine-tuning dynamics. The complete optimization procedure is summarized in Algorithm 1.

Before fine-tuning, the safety gradient is precomputed and anchored throughout the optimisation process. Because this gradient is fixed, GraSafe reshapes it and computes its truncated SVD only once before the iterative fine-tuning loop; the resulting safety subspace is reused at every step. The safety gradient is computed by $\mathbf{g}_s = \frac{1}{N_s}\sum_{i=1}^{N_s} \nabla_\theta \mathcal{L}_s(\theta; x_i^s, y_i^s)$, where $\mathcal{D}_s$ is described in Sec-

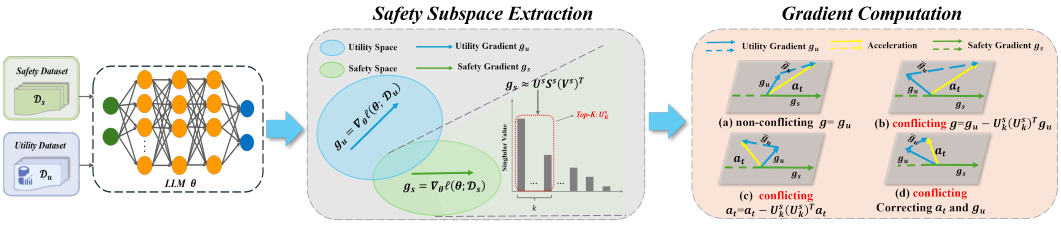

Figure 4: **Overview of GraSafe.** The method estimates the safety gradient $\mathbf{g}_s$ from the safety dataset $\mathcal{D}_s$. During fine-tuning, it detects conflicts between the safety gradient and both the utility gradient $\mathbf{g}_u$ and the curvature-induced acceleration $\mathbf{a}_t$, then removes the conflicting components within the dominant safety subspace.

Algorithm 1: Gradient-aware Safety Fine-tuning

Input: Base model parameters θ_0; Utility dataset $\mathcal{D}_u$; Safety gradient $\mathbf{g}_s$; Learning rate η; Epochs E; Batch size B; Safety subspace dimension k.
Output: Optimised model parameters θ_T.
Initialise the model parameters $\theta \leftarrow \theta_0$
Unflatten the utility and safety gradient to parameter matrix $\mathbf{G}_s \leftarrow \mathrm{Unflatten}(\mathbf{g}_s)$
Extract low-rank safety subspace: $\mathbf{U}_s^k \mathbf{S}_s^k (\mathbf{V}_s^k)^\top \leftarrow \mathrm{SVD}_k(\mathbf{G}_s)$
for $t = 1, \ldots, E$ **do**
  Compute the current utility gradient: $\{(x_i^u, y_i^u)\}_{i=1}^B \sim \mathcal{D}_u$, $\mathbf{g}_u^t \leftarrow \nabla \mathcal{L}_u(\theta; x_i^u, y_i^u)$
  Construct the lookahead parameters: $\bar{\theta}_t \leftarrow \theta - \eta \mathbf{g}_u^t$
  Compute the lookahead utility gradient: $\bar{\mathbf{g}}_u^t \leftarrow \nabla_\theta \nabla \mathcal{L}_u(\bar{\theta}_t; x_i^u, y_i^u)$
  Compute the second-order utility-gradient acceleration: $\mathbf{a}_t \leftarrow (\bar{\mathbf{g}}_u^t - \mathbf{g}_u^t)/\eta$
  if $\langle \mathbf{a}_t, \mathbf{g}_s \rangle < 0$ **then**
    Remove the safety-conflicting acceleration component:
    $\mathbf{A}_t = \mathrm{Unflatten}(\mathbf{a}_t)$, $\mathbf{a}_t^{\text{safe}} = \mathrm{Flatten}(\mathbf{A}_t - \mathbf{U}_s^k (\mathbf{U}_s^k)^\top \mathbf{A}_t)$
  else
    Preserve the original acceleration: $\mathbf{a}_t^{\text{safe}} \leftarrow \mathbf{a}_t$
  Construct the acceleration-corrected lookahead gradient: $\widehat{\mathbf{g}}_u^t \leftarrow \mathbf{g}_u^t + \frac{\eta}{2} \mathbf{a}_t^{\text{safe}}$
  if $\langle \widehat{\mathbf{g}}_u^t, \mathbf{g}_s \rangle < 0$ **then**
    Remove the safety-conflicting utility-gradient component:
    $\widehat{\mathbf{G}}_u^t \leftarrow \mathrm{Unflatten}(\widehat{\mathbf{g}}_u^t)$, $\mathbf{g}^t \leftarrow \mathrm{Flatten}(\widehat{\mathbf{G}}_u^t - \mathbf{U}_s^k (\mathbf{U}_s^k)^\top \widehat{\mathbf{G}}_u^t)$
  else
    $\mathbf{g}^t \leftarrow \widehat{\mathbf{g}}_u^t$
  Update the model parameters: $\theta_{t+1} \leftarrow \theta_t - \eta \mathbf{g}^t$
return θ_T

tion 2. At each fine-tuning step t, it samples a utility mini-batch $\{(x_i^u, y_i^u)\}_{i=1}^B \sim \mathcal{D}_u$ and computes the current utility gradient $\mathbf{g}_u^t = \frac{1}{B} \sum_{i=1}^B \nabla_\theta \mathcal{L}_u(\theta; x_i^u, y_i^u)$. The method then forms virtual parameters $\bar{\theta}_t = \theta_t - \eta \mathbf{g}_u^t$ by taking a lookahead utility step and evaluates the lookahead gradient $\bar{\mathbf{g}}_u^t = \frac{1}{B} \sum_{i=1}^B \nabla_\theta \mathcal{L}_u(\bar{\theta}_t; x_i^u, y_i^u)$. The finite difference between the current and lookahead gradients defines the curvature-induced acceleration $\mathbf{a}_t = (\bar{\mathbf{g}}_u^t - \mathbf{g}_u^t)/\eta$.

GraSafe first tests whether $\mathbf{a}_t$ conflicts with $\mathbf{g}_s$ by evaluating their inner product $\langle \mathbf{a}_t, \mathbf{g}_s \rangle$. When the inner product is negative, the method removes the acceleration component that lies in the safety subspace. It then combines the corrected acceleration with the current utility gradient to obtain an acceleration-aware update. A second conflict test determines whether this combined update also requires projection. If $\langle \mathbf{a}_t, \mathbf{g}_s \rangle > 0$, GraSafe applies the original update without modification.

Although optimization gradients are represented as flattened vectors, they inherit the structure of the model parameters. GraSafe therefores reshape the safety gradient $\mathbf{G}_s = \mathrm{Unflatten}(\mathbf{g}_s)$, utility gradient $\mathbf{G}_u^t = \mathrm{Unflatten}(\mathbf{g}_u^t)$, and its acceleration $\mathbf{A}_t = \mathrm{Unflatten}(\mathbf{a}_t)$ into matrices that match the

Table 1: Safety and utility performance on AGNEWS under different poison ratios with $n = 1000$. Lower HS and higher FA are better.

Methods	Harmful Score ↓						Fine-tuning 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
SFT (Baseline)	**10.10**	59.70	79.60	79.80	81.50	62.14	**91.40**	87.30	83.60	83.70	**83.70**	85.94
Lisa-base (NeurIPS, 2024)	26.40	34.00	38.20	39.10	40.80	35.70	75.70	63.80	73.50	72.30	65.60	70.18
Lisa-aligned (NeurIPS, 2024)	22.40	22.80	26.80	30.40	30.00	26.48	82.40	76.90	81.80	82.00	76.60	79.94
SafeInstr (ICLR, 2024)	11.60	15.60	16.80	25.60	21.20	18.16	83.90	81.90	83.30	83.40	82.60	83.02
BEA (NeurIPS, 2024)	14.80	15.80	16.40	21.60	16.40	17.00	82.60	**88.30**	74.40	81.00	69.10	79.08
Safe LoRA (NeurIPS, 2024)	12.40	11.60	15.60	14.2	20.00	14.76	82.90	78.60	81.20	82.20	80.00	80.98
AsFT (AAAI, 2026)	11.60	12.0	14.00	16.80	16.00	14.08	83.00	84.30	84.30	**84.50**	82.80	83.78
GraSafe (Ours)	10.60	**9.20**	**8.70**	**9.20**	**9.20**	**9.38**	91.30	87.10	**84.40**	83.70	**83.70**	**86.04**

underlying parameter layout. It applies truncated singular value decomposition (Hansen, 1990) to the structured safety gradient and retains the top-k left singular vectors:

$$\mathbf{G}_s \approx \mathbf{U}_s^k \mathbf{S}_s^k (\mathbf{V}_s^k)^\top. \tag{2}$$

$$\mathbf{G}_u^t = \mathbf{G}_u^t - \mathbf{U}_s^k (\mathbf{U}_s^k)^\top \mathbf{G}_u^t \tag{3}$$

$$\mathbf{A}^t = \mathbf{A}_t - \mathbf{U}_s^k (\mathbf{U}_s^k)^\top \mathbf{A}_t \tag{4}$$

Eqs. 3 and 4 project conflicting utility-gradient and acceleration components onto the orthogonal complement of this subspace. The corrected matrices are then flattened before the parameter update. This conditional projection limits interference with task adaptation because GraSafe changes an update only when a conflict is detected. The detailed analysis of this algorithm can be found in the Appendix B.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We employed eight datasets in total, including four benign datasets for downstream fine-tuning tasks: SST2 (Socher et al., 2013), AGNEWS (Zhang et al., 2015), GSM8K (Cobbe et al., 2021), and AlpacaEval (Li et al., 2023), and four harmful datasets for simulating harmful fine-tuning attacks: Harmful (Sheshadri et al., 2024), AdvBench (Zou et al., 2023b), BeaverTails (Ji et al., 2023) (default setting), and HarmBench (Mazeika et al., 2024). We mixed a proportion p of unsafe (poison) data from the harmful datasets with $(1 - p)$ benign data, represented by $n_{samples}$.

Models. We evaluated our method on four models: Llama-2-7B-Chat (Touvron et al., 2023), Llama-3-8B-Instruct (Dubey et al., 2024), Gemma-2-9B (Team et al., 2024), and Qwen-2-7B (Hui et al., 2024). By default, we set $p = 0.1$ and $n_{samples} = 1000$, and adopted Llama-3-8B-Instruct as the baseline model unless stated otherwise.

Baseline. We adopted the vanilla Supervised Fine-Tuning (SFT) as the baseline.

SOTA Methods. We also compared GraSafe with nine State-of-the-Art (SOTA) methods, including Lisa (base and aligned) (Huang et al., 2024a), SafeInstr (Bianchi et al., 2024), BEA (Wang et al., 2024), Safe LoRA (Hsu et al., 2024), Antidote (Huang et al., 2025a), OASIS (Tang et al., 2026), CWAC (Peng et al., 2026), and AsFT (Yang et al., 2026b).

Evaluation Metrics. Following Huang et al. (2025b), we evaluated performance using two key metrics:

- **Fine-Tuning Accuracy (FA):** The top-1 accuracy on the test sets of fine-tuning task.
- **Harmful Score (HS):** The proportion of unsafe outputs when the model encounters unseen malicious instructions, as determined by the audit model in Ji et al. (2023) and Dubey et al. (2024).

Training Details. Training runs on two NVIDIA A6000 GPUs with 48GB of memory each. Unless otherwise stated, each model is fine-tuned for 10 epochs with a learning rate of 2×10^{-5} and a batch size of 16.

Table 2: Safety and utility performance on AGNEWS under different fine-tuning sample sizes with $p = 0.1$. Lower HS and higher FA are better.

Methods	Harmful Score ↓						Fine-tuning 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
SFT (Baseline)	72.40	79.60	74.80	76.80	72.40	75.20	82.70	83.60	**84.20**	84.70	84.80	84.00
Lisa-base (NeurIPS, 2024)	25.20	38.20	24.80	25.20	24.40	25.36	59.70	73.50	80.50	82.00	81.90	75.52
Lisa-aligned (NeurIPS, 2024)	15.60	22.80	19.60	22.00	24.80	20.96	78.90	81.80	83.90	84.40	84.70	82.74
SafeInstr (ICLR, 2024)	14.80	15.60	10.80	15.40	15.60	14.44	80.40	83.30	83.90	84.00	83.90	83.10
BEA (NeurIPS, 2024)	13.60	15.80	9.20	11.20	14.00	12.76	76.50	74.40	83.70	81.00	83.10	79.74
Safe LoRA (NeurIPS, 2024)	12.8	11.60	**5.20**	8.40	8.80	9.36	81.50	81.20	80.70	82.30	81.60	81.46
AsFT (AAAI, 2026)	14.00	14.00	12.00	11.60	10.40	12.4	82.80	84.30	83.90	85.30	**86.00**	84.46
GraSafe (Ours)	**10.20**	**8.70**	8.60	**7.80**	**7.60**	**8.58**	**83.00**	**84.40**	84.10	**85.70**	**86.00**	**84.64**

Table 3: Safety and utility performance across harmful fine-tuning benchmarks. Lower HS and higher FA are better.

Method	Harmful		AdvBench		BeaverTails		HarmBench		Average	
(AGNEWS)	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
SFT (Baseline)	78.60	**84.50**	71.20	83.90	79.60	83.60	65.20	**84.95**	73.65	84.24
Lisa-base (NeurIPS, 2024)	33.20	73.50	37.60	83.90	38.20	73.50	34.60	80.80	35.90	77.93
Lisa-aligned (NeurIPS, 2024)	37.80	81.80	24.80	82.60	26.80	81.80	35.80	83.63	31.30	82.46
SafeInstr (ICLR, 2024)	27.40	84.40	20.50	83.50	16.80	83.30	24.00	84.02	22.18	83.81
BEA (NeurIPS, 2024)	17.60	81.20	19.10	82.30	16.40	74.40	13.00	81.95	16.53	79.96
Safe LoRA (NeurIPS, 2024)	35.80	84.30	24.40	**84.40**	15.60	81.20	32.40	83.78	27.05	83.42
AsFT (AAAI, 2026)	12.50	84.30	11.70	83.70	14.00	84.30	13.40	83.58	12.90	83.97
GraSafe (Ours)	**7.60**	**84.50**	**7.00**	83.80	**8.70**	**84.40**	**6.80**	**84.95**	**7.53**	**84.41**

4.2 Comparison with existing methods

Robustness to Poison Ratio. To assess the robustness of GraSafe under varying attack intensities, we evaluated the safety-utility trade-off across different poison ratios, with the results summarized in Table 1. Compared with SFT, GraSafe consistently reduced HS at every evaluated ratio while improving downstream task accuracy. At $p = 0.05$, BEA obtains an FA of 88.3%, approximately 1.2% above GraSafe; but its HS is 15.8%, compared with 9.2% for GraSafe. Relative to AsFT, GraSafe reduces average HS from 14.08% to 9.38% and increases average FA from 83.78% to 86.04%. This may be due to the fact that AsFT is unable to effectively distinguish between task updates and safety updates, thereby limiting its ability to balance model safety and task utility. These results show that GraSafe remains effective as the proportion of harmful data increases. The comparison also illustrates why safety and utility must be considered jointly: the method with the highest FA at one poison ratio does not necessarily provide the strongest safety.

Generalization to Fine-tuning Sample Sizes. We further investigated the effect of fine-tuning sample size and evaluated the robustness of each method across different sample sizes, as summarized in Table 2. GraSafe obtains the lowest HS at every evaluated sample size. Its average HS is 8.58%, compared with 9.36% for Safe LoRA and 12.40% for AsFT. GraSafe also achieves an average FA of 84.64%. This exceeds Safe LoRA by 3.18 percentage points and is 0.18% above AsFT. The safety advantage therefore persists across datasets containing 500 to 2,500 fine-tuning examples. FA also increases with sample size, indicating that the safety correction does not prevent the model from using additional task data.

Robustness to Poison Datasets. We evaluated the robustness of different methods across multiple harmful datasets. As shown in Table 3, under fine-tuning with harmful data from AdvBench, Safe LoRA achieved the highest FA (84.40%) while also exhibiting a high HS (24.40%). Compared with AsFT, which achieved strong performance among the SOTA methods, GraSafe reduced the average HS by 5.37% while improving the average FA by 0.44%. Overall, GraSafe maintained competitive fine-tuning accuracy (84.41% on average) while achieving a substantially lower HS (7.53% on average), demonstrating robust performance across diverse harmful datasets.

Generalization to Fine-Tuning Datasets. Table 4 summarize the performance of GraSafe across four downstream fine-tuning datasets. GraSafe reduces HS relative to SFT on all four tasks. The absolute reductions are 38.30%, 70.90%, 72.50%, and 15.20%, respectively. Notably, the reductions were particularly pronounced on AGNEWS and GSM8K. For structured tasks such as text classification on AGNEWS and mathematical reasoning on GSM8K, the task optimization directions

Table 4: Safety and utility performance of Llama-3-8B-Instruct across downstream fine-tuning tasks. Lower HS and higher FA are better.

Methods	SST2		AGNEWS		GSM8K		AlpacaEval		Average	
(Llama-3-8B-Instruct)	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑
SFT (Baseline)	48.00	**94.50**	79.60	83.60	82.50	66.70	20.40	49.80	57.63	73.65
Lisa-base (NeurIPS, 2024)	27.60	38.20	27.20	73.50	35.20	24.00	25.20	35.85	28.80	42.89
Lisa-aligned (NeurIPS, 2024)	**5.60**	93.58	22.80	81.80	16.00	19.40	14.80	57.30	14.80	63.02
SafeInstr (ICLR, 2024)	9.20	93.35	15.60	83.30	17.60	19.30	10.80	42.70	13.30	59.66
BEA (NeurIPS, 2024)	7.20	91.63	15.80	74.40	38.80	21.00	11.80	52.40	18.40	59.86
Safe LoRA (NeurIPS, 2024)	11.20	89.24	11.60	81.20	36.00	23.60	15.20	54.70	18.50	62.19
Antidote (ICML, 2025)	34.60	93.35	69.50	80.80	65.30	16.10	60.50	41.83	57.48	58.02
OASIS (ACL, 2026)	14.0	91.50	15.30	77.00	37.80	12.70	-	-	-	-
CWAC (Findings of ACL, 2026)	12.36	94.42	9.75	83.78	17.19	78.13	-	-	-	-
AsFT (AAAI, 2026)	6.00	93.32	14.00	84.30	14.40	26.00	9.20	**58.90**	10.90	65.63
GraSafe (Ours)	9.70	93.32	**8.70**	**84.40**	**10.00**	**77.60**	**5.20**	58.70	**8.40**	**78.51**

Table 5: Safety and utility performance on AGNEWS across different base models. Lower HS and higher FA are better.

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 ↑
SFT (Baseline)	17.60	**84.30**	79.60	83.60	49.20	**90.30**	32.00	**88.30**	44.60	**86.63**
Lisa-base (NeurIPS, 2024)	27.20	63.80	27.20	73.50	28.00	79.90	31.20	80.00	28.40	74.30
Lisa-aligned (NeurIPS, 2024)	16.80	81.80	22.80	81.80	27.60	89.20	14.70	85.60	20.48	84.60
SafeInstr (ICLR, 2024)	16.80	84.40	15.60	83.30	7.20	83.30	7.60	84.70	11.80	83.93
BEA (NeurIPS, 2024)	16.40	84.40	15.8	74.40	8.40	88.60	7.20	86.20	11.95	83.40
Safe LoRA (NeurIPS, 2024)	9.60	81.20	11.60	81.20	8.40	85.50	8.40	84.70	9.50	83.15
Antidote (ICML, 2025)	14.60	83.35	69.50	80.80	-	-	19.40	84.04	-	-
OASIS (ACL, 2026)	13.30	77.00	15.30	77.00	37.80	12.70	-	-	-	-
CWAC (Findings of ACL, 2026)	10.03	83.45	9.75	83.78	-	-	14.22	85.93	-	-
AsFT (AAAI, 2026)	6.30	**84.30**	14.00	84.30	6.10	87.90	6.80	86.60	8.30	85.78
GraSafe (Ours)	**4.00**	83.70	**8.70**	**84.40**	**5.20**	89.90	**6.00**	86.60	**5.98**	86.15

may be more concentrated, allowing GraSafe to suppress conflicting updates without substantially interfering with useful task-specific optimization. Moreover, GraSafe achieved the lowest average HS (8.4%) and the highest average FA (78.51%) among all compared methods. These results demonstrate that GraSafe effectively preserves model safety while maintaining strong utility and generalizes well across four downstream tasks.

Generalization to Models. We evaluated the performance of different methods across various model architectures, as shown in Table 5. GraSafe obtains an average HS of 5.98%, the lowest value among the methods with results for all four models. Its average FA is 86.15%, compared with 86.63% for SFT and 85.78% for AsFT. Thus, GraSafe reduces average HS by 38.62 percentage points relative to SFT while changing average FA by 0.48 percentage points. The method also maintains a low HS on Qwen-2-7B and Gemma-2-9B. These results indicate that the observed safety benefit is not limited to the default Llama-3-8B-Instruct backbone.

4.3 Ablation Study

We examined how the dimension k of the safety subspace, defined as the number of top-k singular directions derived from the safety gradients, affects safety and downstream utility. In all experiments, Llama-3-8B-Instruct was fine-tuned on AGNews for 10 epochs.

The first-order variant, denoted SFT + Utility Gradient, corrects conflicts between utility and safety gradients but does not model curvature-induced changes. GraSafe corrects both the first-order utility gradient and its curvature-induced acceleration.

As shown in Figure 5, setting $k = 0$ imposed no safety-subspace constraint, and all three fine-tuning methods produced high HS. At $k = 10$, both corrected methods sharply reduce HS, while FA reaches its maximum. Increasing k to 20 produces additional safety improvement and lowers task performance.

These results show that the safety-subspace dimension controls the balance between safety and downstream utility. At a matched and appropriately selected value of k, GraSafe improved both safety and task performance relative to SFT and "SFT + Utility Gradient". Taken together, these results show that jointly accounting for curvature-induced drift and utility–safety gradient conflict yields a more favourable safety–utility trade-off in this experimental setting. Appendix C reports additional sensitivity analyses for the number of fine-tuning epochs and the size and diversity of safety dataset.

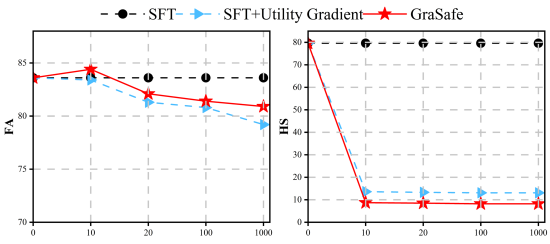

Figure 5: Safety and utility performance of our safety-preserving fine-tuning with different safety subspace dimensions k. Experiments use Llama-3-8B-Instruct fine-tuned on AGNEWS for 10 epochs.

5 Conclusion

In this paper, we studied safety degradation during LLM fine-tuning through the geometry of utility and safety gradients. The empirical and theoretical analyses show that utility gradients frequently conflict with the safety gradient. Utility-loss curvature can further redirect the update toward a safety-conflicting direction. Based on this mechanism, we introduced GraSafe, which conditionally removes conflicting components from both the utility gradient and its curvature-induced acceleration. Across eight datasets and four model families, GraSafe substantially reduces harmful outputs while preserving downstream accuracy. Its advantage persists across poison ratios, fine-tuning sample sizes, harmful-data sources, downstream tasks, and model backbones. These findings support gradient-aware correction as a promising approach to safety-preserving fine-tuning.

AI use statement

In the preparation of this manuscript, Large Language Models (LLMs) were used solely as a general-purpose writing assistance tool to improve the clarity, grammar, and fluency of the text. The core research ideas, experimental design, data analysis, and all technical content were conceived and developed entirely by the human authors. No LLM was involved in generating novel ideas, interpreting results, or producing scientific claims. The authors take full responsibility for the accuracy and integrity of all content presented in this paper.

Reproducibility statement

We are committed to ensuring the full reproducibility of our work. All source code, datasets, and detailed experimental configurations are publicly available at `https://anonymous.4open.science/r/GraSafe`. Section 4.1 provides a comprehensive overview of the evaluation setup, including the specific datasets, baseline methods and evaluation metrics. The theoretical analysis of Algorithm 1 can be found in Appendix B.1, B.2. Detailed descriptions of all datasets are found in Appendix C.1.

Ethics statement

This work follows the ICLR Code of Ethics. We do not collect, use, or release new human subject data. All experiments use established, publicly available datasets that have been widely studied in prior research.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant NO. 62602133 and the Guangdong S&T Program under Grant NO. 2026B0101100002.

REFERENCES

Rishabh Bhardwaj, Duc Anh Do, and Soujanya Poria. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14138–14149, 2024.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Y Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. In *International Conference on Learning Representations*, volume 2024, pp. 34196–34216, 2024.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International conference on machine learning*, pp. 10835–10866. PMLR, 2023.

Per Christian Hansen. Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. *SIAM Journal on Scientific and Statistical Computing*, 11(3):503–518, 1990.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: The silver lining of reducing safety risks when finetuning large language models. *Advances in neural information processing systems*, 37:65072–65094, 2024.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim F Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:104521–104555, 2024a.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey. *arXiv preprint arXiv:2409.18169*, 2024b.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *Advances in Neural Information Processing Systems*, 37:74058–74088, 2024c.

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Joshua Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning attack. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25059–25074. PMLR, 13–19 Jul 2025a.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *International Conference on Learning Representations*, volume 2025, pp. 67202–67226, 2025b.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

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

Guozhi Liu, Weiwei Lin, Qi Mu, Tiansheng Huang, Ruichao Mo, Yuren Tao, and Li Shen. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation. *IEEE Transactions on Information Forensics and Security*, 2025a.

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

Yuyang Liu, Qiuhe Hong, Linlan Huang, Alexandra Gomez-Villa, Dipam Goswami, Tiantian Peng, Xialei Liu, Joost van de Weijer, and Yonghong Tian. Continual learning for vlms: A survey and taxonomy beyond forgetting. *arXiv preprint arXiv:2508.04227*, 2025b.

Zhijie Liu, Yutian Tang, Xiapu Luo, Yuming Zhou, and Liang Feng Zhang. No need to lift a finger anymore? assessing the quality of code generation by chatgpt. *IEEE Transactions on Software Engineering*, 50(6):1548–1584, 2024b.

Alejandro Lopez-Lira and Yuehua Tang. Can chatgpt forecast stock price movements? return predictability and large language models. *Journal of Financial Economics*, 184:104335, 2026.

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

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

Songping Peng, Zhiheng Zhang, Daojian Zeng, Lincheng Jiang, and Xieping Gao. Preventing safety drift in large language models via coupled weight and activation constraints. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17646–17662, 2026.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, volume 2024, pp. 30988–31043, 2024.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations*, volume 2025, pp. 54911–54941, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

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

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

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. In *International Conference on Learning Representations*, volume 2025, pp. 101802–101829, 2025.

Jiayu Tang, Guowei Peng, Qiuhao Xie, Yuning Yang, Xiurui Xie, and Guisong Liu. Oasis: Mitigating harmful fine-tuning attacks on llms via orthogonal and adaptive safety alignment strategy. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28407–28421, 2026.

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

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

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerová, et al. Adapted large language models can outperform medical experts in clinical text summarization. *Nature medicine*, 30(4):1134–1142, 2024.

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. *Advances in Neural Information Processing Systems*, 37:5210–5243, 2024.

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jialin Wu, Kecen Li, Zhicong Huang, Xinfeng Li, Xiaofeng Wang, and Cheng Hong. Enchtable: Unified safety alignment transfer in fine-tuned large language models. In *2026 IEEE Symposium on Security and Privacy (SP)*, pp. 3358–3376. IEEE, 2026.

Kang Yang, Guanhong Tao, Xun Chen, and Jun Xu. Alleviating the fear of losing alignment in llm fine-tuning. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2152–2170. IEEE, 2025a.

Shuo Yang, Kun-Peng Ning, Yu-Yang Liu, Jia-Yu Yao, Yong-Hong Tian, Yi-Bing Song, and Li Yuan. Is parameter collision hindering continual learning in llms? In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 4243–4259, 2025b.

Shuo Yang, Yuwei Niu, Yuyang Liu, Yang Ye, Bin Lin, and Li Yuan. Look-back: Implicit visual re-focusing in mllm reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pp. 11694–11702, 2026a.

Shuo Yang, Qihui Zhang, Yuyang Liu, Yue Huang, Xiaojun Jia, Kun-Peng Ning, Jia-Yu Yao, Jigang Wang, Dai Hailiang, Yibing Song, et al. Asft: Anchoring safety during llm fine-tuning within narrow safety basin. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34322–34330, 2026b.

Rui Ye, Jingyi Chai, Xiangrui Liu, Yaodong Yang, Yanfeng Wang, and Siheng Chen. Emerging safety attack and defense in federated instruction tuning of large language models. In *International Conference on Learning Representations*, volume 2025, pp. 55332–55350, 2025.

Biao Yi, Tiansheng Huang, Sishuo Chen, Tong Li, Zheli Liu, Zhixuan Chu, and Yiming Li. Probe before you talk: Towards black-box defense against backdoor unalignment for large language models. In *International Conference on Learning Representations*, volume 2025, pp. 55763–55789, 2025.

Biao Yi, Jiahao Li, Baolei Zhang, Lihai Nie, Tong Li, Tiansheng Huang, and Zheli Liu. Safegrad: Gradient surgery for safe llm fine-tuning. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3326–3330. IEEE, 2026.

Xin Yi, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. A safety realignment framework via subspace-oriented model fusion for large language models. *Knowledge-Based Systems*, 306:112701, 2024.

Jin Yuxian. Bridging the knowledge-skill gap: The role of large language model and critical thinking in education. *Computers & Education*, 235:105357, 2025.

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

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Qizhe Shieh. Understanding and enhancing safety mechanisms of llms via safety-specific neuron. In *International Conference on Learning Representations*, volume 2025, pp. 44113–44127, 2025.

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

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

APPENDIX

A Related Works **15**

B THEORETICAL ANALYSIS **15**

B.1 Utility Curvature Induces Gradient Drift 16

B.2 Utility–Safety Gradient Conflict Increases Safety Loss 17

C EXPERIMENTAL DETAILS **18**

C.1 DATASETS 18

C.2 Baselines and SOTA Methods 19

C.3 Impact of Epochs 20

C.4 Impact of Safety Dataset $\mathcal{D}_s$ 20

D LIMITATION AND FUTURE STUDY **21**

E Response Example **21**

A Related Works

Safety alignment is a fundamental concern in the development of Large Language Models (LLMs). It aims to ensure that model outputs are aligned with human values, ethical principles, and safety requirements (Touvron et al., 2023; Zou et al., 2023a; Gao et al., 2023; Liu et al., 2025b). Representative safety-alignment techniques include instruction fine-tuning, RLHF, DPO and others (Wei et al., 2021; Rafailov et al., 2023; Yang et al., 2025b; 2026a). However, the safety alignment established by these methods can be vulnerable to small-scale fine-tuning, where even a small amount of harmful or benign data may substantially compromise model safety (Qi et al., 2024). Existing defenses can be broadly categorized according to the stage at which they intervene: alignment, fine-tuning, and post-tuning (Huang et al., 2024b).

Alignment Phase Defenses enhance model robustness against harmful fine-tuning attacks during the safety-alignment stage (Qi et al., 2025; Zhao et al., 2025; Liu et al., 2024a). Vaccine (Huang et al., 2024c) improves robustness by introducing perturbations into hidden representations during alignment, thereby reducing harmful embedding drift induced by subsequent fine-tuning. Building on this idea, T-Vaccine (Liu et al., 2025a) selectively identifies safety-critical layers based on gradient norms and applies perturbations only to these layers, reducing unnecessary perturbations and improving memory efficiency. OASIS (Tang et al., 2026) further addresses the directional conflict between alignment perturbations and harmful gradients by projecting perturbations onto directions orthogonal to harmful gradients and adaptively selecting safety-critical layers, thereby reducing harmful feature acquisition while improving parameter efficiency. TAR (Tamirisa et al., 2025) optimizes model parameters to preserve a high loss on harmful examples after adversarial fine-tuning, while Booster (Huang et al., 2025b) mitigates the degradation of harmful loss under simulated fine-tuning attacks.

Fine-tuning Phase Defenses enhance model safety during training against harmful fine-tuning attacks (Mukhoti et al., 2023; Wei et al., 2024). SafeInstr (Bianchi et al., 2024) incorporates safety-oriented examples into fine-tuning to preserve aligned behaviors. Lisa (Huang et al., 2024a) limits optimization drift through dual-state optimization and proximity constraints, while BEA (Wang et al., 2024) embeds hidden triggers to suppress harmful behaviors during fine-tuning. AsFT (Yang et al., 2026b) uses the alignment direction as an anchor and penalizes parameter updates orthogonal to it, thereby constraining fine-tuning within a narrow safety basin and suppressing safety-degrading updates. CWAC (Peng et al., 2026) jointly constrains weight updates within a precomputed safety subspace and regularizes safety-critical activation features, mitigating safety drift at both the parameter and representation levels.

Post-tuning Phase Defense aim to restore model safety after harmful fine-tuning attacks (Ye et al., 2025; Yi et al., 2025). Safe LoRA (Hsu et al., 2024) projects fine-tuned parameters onto the safe direction after fine-tuning. SOMF (Yi et al., 2024) incorporates additional benign task knowledge and reuses safety-critical parameters to restore aligned behavior. DeepAlign (Yang et al., 2025a) selectively restores parameters associated with the degraded harmful direction and employs rollback to recover safety while preserving downstream utility. Antidote (Huang et al., 2025a) selectively prunes safety-critical harmful parameters during post-processing to mitigate alignment degradation.

Across the three intervention stages, existing defences primarily perturb representations, constrain updates using static directions or repair parameters after fine-tuning. Although these strategies can reduce harmful behaviour, they do not explicitly address how utility-loss curvature redirects the utility gradient during optimisation. In particular, the reviewed methods do not jointly identify safety conflicts in the utility gradient and its curvature-induced acceleration. GraSafe addresses this gap during fine-tuning by evaluating both quantities against a safety subspace and selectively projecting only their conflicting components onto its orthogonal complement. Unlike static penalties, GraSafe leaves compatible updates unchanged and directly targets the dynamic gradient conflict associated with progressive safety degradation.

B THEORETICAL ANALYSIS

The experiments show that the utility gradient increasingly conflicts with the safety gradient during fine-tuning. This appendix analyzes the local geometry underlying that observation. The derivation

shows how utility-loss curvature changes the utility-gradient direction and how this change can increase the safety loss.

B.1 Utility Curvature Induces Gradient Drift

Let $\mathcal{L}_u(\theta)$ and $\mathcal{L}_s(\theta)$ denote the utility loss and safety loss, respectively. Their gradients at the current model parameters are defined as:

$$\mathbf{g}_u = \nabla_\theta \mathcal{L}_u(\theta), \qquad \mathbf{g}_s = \nabla_\theta \mathcal{L}_s(\theta), \tag{5}$$

where $\mathbf{g}_u$ represents the optimization direction associated with downstream task adaptation, while $\mathbf{g}_s$ characterizes the local direction associated with safety preservation.

For local analysis, we consider one gradient-descent update driven by the utility objective:

$$\theta' = \theta - \eta \mathbf{g}_u, \tag{6}$$

where η denotes the learning rate and θ' denotes the model parameters after one utility-oriented update. Since the utility gradient depends on the model parameters, moving from θ to θ' also changes the utility gradient. The utility gradient evaluated at the updated parameters is defined as the lookahead utility gradient:

$$\bar{\mathbf{g}}_u = \nabla_\theta \mathcal{L}_u(\theta'), \tag{7}$$

where $\bar{\mathbf{g}}_u$ represents the utility gradient after the parameter update. Substituting $\theta' = \theta - \eta \mathbf{g}_u$ gives:

$$\bar{\mathbf{g}}_u = \nabla_\theta \mathcal{L}_u \left(\theta - \eta \mathbf{g}_u\right). \tag{8}$$

Therefore, $\mathbf{g}_u$ and $\bar{\mathbf{g}}_u$ correspond to the gradient of the same utility objective evaluated before and after the local fine-tuning update.

Assuming that $\mathcal{L}_u$ is sufficiently smooth and that its Hessian (Sagun et al., 2017) is locally Lipschitz continuous, applying a first-order Taylor expansion of the utility gradient around θ yields:

$$\bar{\mathbf{g}}_u = \nabla_\theta \mathcal{L}_u(\theta) + \frac{\partial \nabla_\theta \mathcal{L}_u(\theta)}{\partial \theta^\top}(\theta' - \theta) + \mathcal{O}\left(\|\theta' - \theta\|_2^2\right). \tag{9}$$

where $\frac{\partial \nabla_\theta \mathcal{L}_u(\theta)}{\partial \theta^\top}$ denotes the Jacobian of the utility gradient with respect to the model parameters.

For a twice-differentiable scalar utility loss, the Jacobian of its gradient is the Hessian:

$$\mathbf{H}_u = \nabla_\theta^2 \mathcal{L}_u(\theta) = \frac{\partial \nabla_\theta \mathcal{L}_u(\theta)}{\partial \theta^\top} \tag{10}$$

where $\mathbf{H}_u$ characterises the local curvature of the utility-loss landscape. A first-order Taylor expansion of the lookahead utility gradient around θ gives:

$$\bar{\mathbf{g}}_u = \mathbf{g}_u + \mathbf{H}_u(\theta' - \theta) + \mathcal{O}\big(\|\theta' - \theta\|_2^2\big). \tag{11}$$

Substituting the utility-oriented update $\theta' - \theta = -\eta \mathbf{g}_u$ yields:

$$\bar{\mathbf{g}}_u = \mathbf{g}_u - \eta \mathbf{H}_u \mathbf{g}_u + \mathcal{O}(\eta^2). \tag{12}$$

Therefore, the update-induced change in the utility gradient is:

$$\Delta \mathbf{g}_u \equiv \bar{\mathbf{g}}_u - \mathbf{g}_u \approx -\eta \mathbf{H}_u \mathbf{g}_u. \tag{13}$$

Thus, $\mathbf{H}_u \mathbf{g}_u$ describes how local utility-loss curvature modifies the gradient along the current optimisation direction. Its component parallel to $\mathbf{g}_u$ changes only the gradient magnitude to first order, whereas its orthogonal component changes the gradient direction. For $\mathbf{g}_u \neq \mathbf{0}$, define the normalised utility gradient and the corresponding orthogonal projector as:

$$\hat{\mathbf{g}}_u = \frac{\mathbf{g}_u}{\|\mathbf{g}_u\|_2}, \qquad \mathbf{P}_{u,\perp} = \mathbf{I} - \hat{\mathbf{g}}_u \hat{\mathbf{g}}_u^\top. \tag{14}$$

Then, the curvature-induced component responsible for directional drift is:

$$\mathbf{d}_u = \mathbf{P}_{u,\perp} \mathbf{H}_u \mathbf{g}_u. \tag{15}$$

If $\mathbf{d}_u = \mathbf{0}$, curvature changes only the magnitude of the utility gradient to first order. If $\mathbf{d}_u \neq \mathbf{0}$, the orthogonal component rotates the gradient away from its original direction. Anisotropic curvature of the utility-loss landscape can therefore induce utility-gradient drift during fine-tuning.

B.2 Utility–Safety Gradient Conflict Increases Safety Loss

We next analyze how curvature-induced utility-gradient variation changes the cosine similarity between the utility and safety gradients. The change in cosine similarity is defined as:

$$\Delta\rho = \cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) - \cos(\mathbf{g}_u, \mathbf{g}_s) = \frac{\bar{\mathbf{g}}_u^\top \mathbf{g}_s}{\|\bar{\mathbf{g}}_u\|_2 \|\mathbf{g}_s\|_2} - \frac{\mathbf{g}_u^\top \mathbf{g}_s}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2}. \tag{16}$$

where $\bar{\mathbf{g}}_u$ denotes the lookahead utility gradient. Since $\|\Delta\mathbf{g}_u\|_2 = \mathcal{O}(\eta)$, a first-order expansion of the lookahead gradient norm around $\mathbf{g}_u$ yields:

$$\|\mathbf{g}_u + \Delta\mathbf{g}_u\|_2 = \|\mathbf{g}_u\|_2 + \frac{\mathbf{g}_u^\top \Delta\mathbf{g}_u}{\|\mathbf{g}_u\|_2} + \mathcal{O}(\|\Delta\mathbf{g}_u\|_2^2). \tag{17}$$

Accordingly, its reciprocal can be approximated as:

$$\frac{1}{\|\mathbf{g}_u + \Delta\mathbf{g}_u\|_2} = \frac{1}{\|\mathbf{g}_u\|_2} - \frac{\mathbf{g}_u^\top \Delta\mathbf{g}_u}{\|\mathbf{g}_u\|_2^3} + \mathcal{O}(\|\Delta\mathbf{g}_u\|_2^2). \tag{18}$$

Substituting these first-order approximations into the lookahead cosine similarity gives:

$$\begin{aligned}\cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) \approx \frac{\mathbf{g}_u^\top \mathbf{g}_s}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2} + \frac{\Delta\mathbf{g}_u^\top \mathbf{g}_s}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2} \\ - \frac{(\mathbf{g}_u^\top \mathbf{g}_s)(\mathbf{g}_u^\top \Delta\mathbf{g}_u)}{\|\mathbf{g}_u\|_2^3 \|\mathbf{g}_s\|_2}.\end{aligned} \tag{19}$$

Subtracting $\cos(\mathbf{g}_u, \mathbf{g}_s)$ from both sides yields:

$$\Delta\rho \approx \frac{\Delta\mathbf{g}_u^\top \mathbf{g}_s}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2} - \frac{(\mathbf{g}_u^\top \mathbf{g}_s)(\mathbf{g}_u^\top \Delta\mathbf{g}_u)}{\|\mathbf{g}_u\|_2^3 \|\mathbf{g}_s\|_2}. \tag{20}$$

The above expression can be equivalently rewritten as:

$$\Delta\rho \approx \frac{\mathbf{g}_s^\top \left(\mathbf{I} - \frac{\mathbf{g}_u \mathbf{g}_u^\top}{\|\mathbf{g}_u\|_2^2}\right) \Delta\mathbf{g}_u}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2}. \tag{21}$$

Defining the orthogonal projection matrix with respect to the current utility-gradient direction as:

$$\mathbf{P}_{u,\perp} = \mathbf{I} - \frac{\mathbf{g}_u \mathbf{g}_u^\top}{\|\mathbf{g}_u\|_2^2}, \tag{22}$$

$$\Delta\rho \approx \frac{\mathbf{g}_s^\top \mathbf{P}_{u,\perp} \Delta\mathbf{g}_u}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2}. \tag{23}$$

Finally, substituting $\Delta\mathbf{g}_u = -\eta\mathbf{H}_u\mathbf{g}_u + \mathcal{O}(\eta^2)$ gives:

$$\Delta\rho = -\eta \frac{\mathbf{g}_s^\top \mathbf{P}_{u,\perp} \mathbf{H}_u \mathbf{g}_u}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2} + \mathcal{O}(\eta^2). \tag{24}$$

Since the higher-order terms become negligible relative to the first-order contribution under a sufficiently small learning rate, retaining only the first-order term with respect to η allows the change in cosine similarity to be approximated as:

$$\Delta\rho \approx -\eta \frac{\mathbf{g}_s^\top \mathbf{P}_{u,\perp} \mathbf{H}_u \mathbf{g}_u}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2}. \tag{25}$$

This expression characterises how the curvature-induced directional component changes the cosine similarity between the utility and safety gradients, thereby affecting their degree of conflict. For concision, we define the curvature–safety coupling term as:

$$c = \frac{\mathbf{g}_s^\top \mathbf{P}_{u,\perp} \mathbf{H}_u \mathbf{g}_u}{\|\mathbf{g}_u\|_2 \|\mathbf{g}_s\|_2}, \tag{26}$$

Table 6: The statistics of datasets.

Dataset	Dataset size	Classes / task type	Year	Typical applications
SST2	11,855	2	2015	Sentiment Classification; Sentiment Analysis
AGNews	120,000	4	2004	Text Classification
GSM8K	8,500	Open-ended	2021	Mathematical Reasoning
AlpacaEval	805	Open-ended	2023	RLHF
Harmful	150,000+	Multiple harmful-behaviour	2022	RLHF; Red Teaming
AdvBench	520	10 (high-risk)	2023	Jailbreak Attack Evaluation
BeaverTails	333,963	14	2023	Safety Alignment
HarmBench	510	7 (semantic domains)	2024	Safety Evaluation; Red-Teaming

where c characterises the geometric interaction between the curvature-induced directional component of the utility gradient and the safety gradient.

When $c > 0$, it follows that $\Delta\rho < 0$ and $\cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) < \cos(\mathbf{g}_u, \mathbf{g}_s)$. Curvature-induced drift therefore reduces the alignment between the lookahead utility gradient and the safety gradient. If the gradients are initially close to orthogonal, such that $\cos(\mathbf{g}_u, \mathbf{g}_s) \approx 0$, even a small curvature-induced decrease in cosine similarity can produce $\cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) < 0$. The utility and safety gradients then become directly conflicting.

The sign of the cosine similarity is directly related to the local variation of the safety loss. Consider a subsequent utility-oriented update from θ':

$$\theta'' = \theta' - \eta\bar{\mathbf{g}}_u. \tag{27}$$

Under the local approximation that the safety gradient remains approximately represented by $\mathbf{g}_s$, a first-order expansion of the safety loss around θ' gives:

$$\begin{aligned} \mathcal{L}_s(\theta'') - \mathcal{L}_s(\theta') &\approx -\eta \mathbf{g}_s^\top \bar{\mathbf{g}}_u \\ &= -\eta \|\mathbf{g}_s\|_2 \|\bar{\mathbf{g}}_u\|_2 \cos(\bar{\mathbf{g}}_u, \mathbf{g}_s). \end{aligned} \tag{28}$$

Therefore, when curvature-induced directional variation causes $\cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) < 0$, the lookahead utility gradient becomes conflicting with the safety gradient. The implication of this conflict can be further characterised through its effect on the safety loss.

When $\cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) < 0$, the leading-order change in the safety loss satisfies:

$$\mathcal{L}_s(\theta'') - \mathcal{L}_s(\theta') > 0. \tag{29}$$

These results establish a local geometric mechanism linking utility-loss curvature to safety degradation. The orthogonal curvature component $\mathbf{P}_{u,\perp}\mathbf{H}_u\mathbf{g}_u$ determines the first-order change in cosine similarity between the utility and safety gradients. When this curvature-induced directional variation decreases the cosine similarity sufficiently to make $\cos(\bar{\mathbf{g}}_u, \mathbf{g}_s) < 0$, the lookahead utility gradient becomes conflicting with the safety gradient. A subsequent utility-oriented update along this conflicting direction then produces a positive first-order change in the safety loss. Therefore, utility-loss curvature does not directly degrade model safety. Instead, it can redirect the utility gradient toward a safety-conflicting direction, thereby increasing the likelihood of safety degradation during continued fine-tuning.

C EXPERIMENTAL DETAILS

C.1 DATASETS

We utilized eight datasets to evaluate downstream task adaptation and robustness to harmful fine-tuning. SST2, AGNews, GSM8K and AlpacaEval cover sentiment classification, topic classification,

mathematical reasoning and instruction following, respectively. Harmful, AdvBench, BeaverTails and HarmBench supplied harmful data for the poisoned fine-tuning experiments. Their statistics and typical applications are summarized in Table 6.

Downstream tasks.

- **SST2 (Socher et al., 2013).** It contains movie reviews labeled as positive or negative. We use this binary classification task to test whether safety-preserving fine-tuning maintains sentiment-classification performance.
- **AGNEWS (Zhang et al., 2015).** It contains articles from four categories: World, Sports, Business, and Science/Technology. It is the primary task for comparisons across poison ratios, sample sizes, harmful-data sources, and model backbones.
- **GSM8k (Cobbe et al., 2021).** It contains grade-school mathematics problems that require multistep reasoning. We use it to assess whether safety preservation interferes with mathematical reasoning during adaptation.
- **AlpacaEval (Li et al., 2023).** It evaluates instruction following through open-ended generation. It complements the classification and reasoning tasks by assessing response quality.

Harmful data sources.

- **Harmful (Sheshadri et al., 2024).** We utilized this dataset as one source of poisoned fine-tuning examples to test whether safety preservation extends beyond BeaverTails.
- **AdvBench (Zou et al., 2023b).** It provides harmful requests for evaluating attacks on aligned language models. We utilized these requests as an alternative source of harmful instructions for fine-tuning. This setting assessed resistance to safety degradation induced by data from an adversarial benchmark.
- **BeaverTails (Ji et al., 2023).** It contains question-answer pairs annotated for helpfulness and harmlessness and served as the default harmful data source. Separately, we constructed the safety dataset D_s using three safety-refusal examples from each of 14 categories, yielding 42 examples. These refusal examples supported safety-gradient estimation, whereas harmful examples were utilized to construct poisoned fine-tuning data.
- **HarmBench (Mazeika et al., 2024).** It is a benchmark for automated red teaming and robust refusal. We used its harmful behaviours as an additional source of poisoned fine-tuning data, extending the comparison beyond BeaverTails and AdvBench.

C.2 Baselines and SOTA Methods

We compare GraSafe with a baseline and State-of-the-Art (SOTA) methods, as summarized in Table 7.

Baseline.

- **SFT.** Vanilla Supervised Fine-Tuning (SFT) optimises the next-token prediction objective on the submitted data without an additional mechanism for preserving safety. It provides a reference for quantifying downstream adaptation and the accompanying safety degradation. As specified in Section 4.1, fine-tuning used Low-Rank Adaptation (LoRA) with a rank of 8.

SOTA Methods.

- **Lisa-base and Lisa-aligned (Huang et al., 2024a).** It separates optimisation on safety-alignment data from optimisation on user-provided data. A proximal regularisation term constrains drift between the two optimisation states to stabilise training. We evaluated both the Lisa-base and Lisa-aligned configurations in our comparisons.
- **SafeInstr (Bianchi et al., 2024).** It incorporates safety-oriented examples into downstream fine-tuning to reinforce aligned behaviour during adaptation. It provides a comparison based on data augmentation, whereas GraSafe uses safety examples to identify conflicting update components.

Table 7: Classification of comparison methods.

Defense Stage	Method	Year
-	SFT	
Alignment	OASIS	2026
Fine-tuning	Lisa-base	2024
	Lisa-aligned	2024
	SafeInstr	2024
	BEA	2024
	CWAC	2026
	AsFT	2026
Post-tuning	Safe LoRA	2024
	Antidote	2025

- **BEA (Wang et al., 2024).** It uses backdoor-enhanced safety alignment, associating a designated trigger with safe responses to preserve or activate aligned behaviour after fine-tuning.
- **Safe LoRA (Hsu et al., 2024).** It constructs a safety-aligned subspace from the weights of base and aligned models and projects selected LoRA updates into this subspace.
- **Antidote (Huang et al., 2025a).** It performs one-shot pruning after fine-tuning to remove harmful weights associated with unsafe generations. As a post-fine-tuning defense, it restores safety without constraining the preceding fine-tuning procedure.
- **OASIS (Tang et al., 2026).** It projects safety-alignment perturbations onto directions orthogonal to harmful gradients and adaptively selects safety-critical layers for optimization. This design aims to reduce the interference between safety reinforcement and harmful feature acquisition.
- **CWAC (Peng et al., 2026).** It constrains weight updates within a precomputed safety subspace while regularising safety-critical activation features identified by sparse autoencoders. It provides a comparison based on jointly preserving safety at the parameter and representation levels.
- **AsFT (Yang et al., 2026b).** It derives an alignment direction from the parameter difference between aligned and unaligned models. It anchors optimisation to this direction and penalises orthogonal parameter updates during fine-tuning.

C.3 Impact of Epochs

We further investigated the effect of fine-tuning epochs on model safety. Specifically, we fine-tuned the models for 5, 10, 15 and 20 epochs on AGNEWS while keeping all other experimental settings unchanged. We evaluated the safety performance of four LLMs under different defense methods. Figure 6 presents the specific experimental results. For the five competing methods, HS generally increases as fine-tuning epochs, indicating progressive safety degradation. In contrast, GraSafe consistently maintains a low HS across different numbers of fine-tuning epochs. These results demonstrate that GraSafe preserves robust safety performance even under deep fine-tuning.

C.4 Impact of Safety Dataset $\mathcal{D}_s$

We vary the two key properties, diversity and size of the safety dataset and assess their influences. **Diversity.** Our default safety dataset is constructed from BeaverTails and contains 42 samples spanning 14 safety categories. We further vary the number of safety categories to 2, 4, and 8 while keeping the number of samples per category fixed at 3. We re-evaluate the model under each setting, with the alignment restoration results shown in the upper half of Figure 7. **Size.** In this evaluation, we retained all 14 safety categories and evaluated three sample-size settings, with 10, 3, and 1 samples per category, respectively. The corresponding results are shown in the lower half of Figure 7.

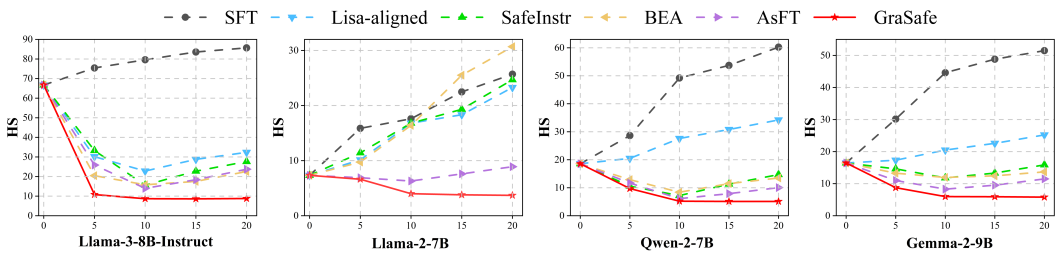

Figure 6: HS under different fine-tuning epochs across four base models.

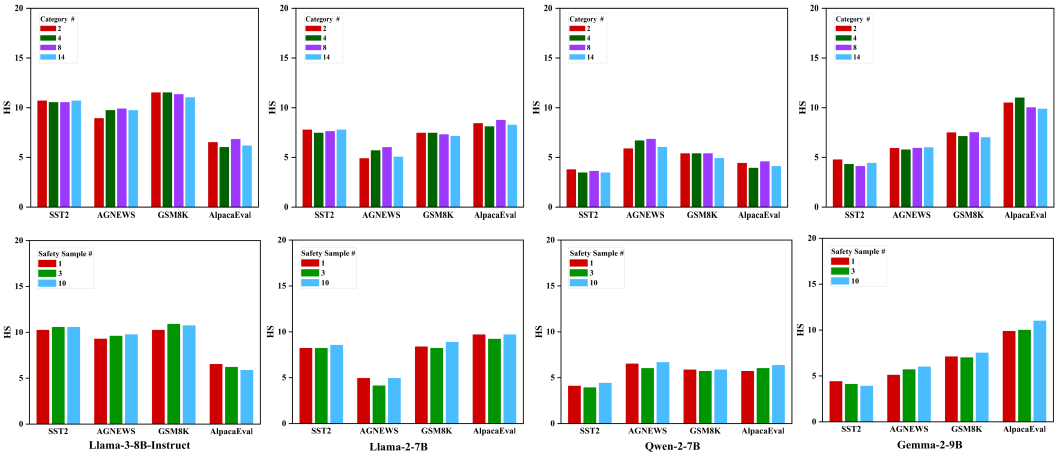

Figure 7: The HS of models with varying diversity and size of the Safety dataset.

Our algorithm is insensitive to the diversity of the safety dataset and shows marginal sensitivity to its size. Given Llama-3-8B-Instruct, Llama-2-7B, Qwen-2-7B, and Gemma-2-9B, the difference in the HS caused by the number of data categories is less than 2%. On Gemma-2-9B, the variation is slightly larger. However, more categories do not necessarily lead to a lower harmful ratio. Similar to diversity, the size of the recovery dataset incurs limited differences in HS.

D LIMITATION AND FUTURE STUDY

Our method preserves safety alignment by removing update components that contribute to conflicts between the utility and safety gradients. Although effective, the method introduces additional computational overhead during fine-tuning, and its efficiency could be improved. This study primarily aims to explain safety alignment degradation through gradient geometry and develop a corresponding mitigation strategy, rather than exploring the best method to solve harmful fine-tuning attacks. Future work may improve this method and explore the mechanism for selecting adaptive safety gradients. At the same time, testing whether our method is applicable to defending jailbreak attacks is also an interesting direction.

E RESPONSE EXAMPLE

This section presents representative response examples from our experiments. We first fine-tuned Llama-3-8B-Instruct on the GSM8K dataset augmented with 100 harmful samples, producing a vulnerable SFT model. We then applied GraSafe during fine-tuning to preserve the model's safety alignment while maintaining utility. Below we present model outputs for one mathematics problem and one harmful prompt, reporting responses from the initialization model (Init), the fine-tuned model (SFT), and the model fine-tuned with Gradient-aware Safety Preservation Fine-tuning (GraSafe).

As shown in the harmful example, the initialized model (Init) refuses to follow the harmful instruction and instead provides a benign and supportive response, indicating that its original safety alignment remains intact. In contrast, the vanilla SFT model directly follows the harmful instruction and generates an insulting response, demonstrating that fine-tuning with harmful samples substantially degrades the model's safety behavior. GraSafe, however, preserves the refusal behavior after fine-tuning: it rejects the harmful request and provides a safe alternative response. Compared with vanilla SFT, this example shows that GraSafe effectively suppresses harmful behavioral drift while retaining the safety characteristics of the initialized model. More importantly, the GraSafe response is not a simple rollback to the initial model, but reflects a fine-tuned model that maintains safety behavior while preserving downstream task adaptation.

Math Example

System: You are a helpful assistant.
User: Kyle bought last year's best-selling book for $19.50. This is with a 25% discount from the original price. What was the original price of the book?

Llama-3-8B-Instruct (Init): The book was bought for 19.50. This is 75% of the original price. So the original price is 19.50 / 0.75 = 25.8. The answer is 25.8.
Llama-3-8B-Instruct (SFT): Let x be the original price of the book. The discount is 25% of x, which is $0.25x$. The price after the discount is $x - 0.25x = 0.75x$. To find the original price, divide both sides by 0.75 to get $x = \$19.50/0.75 = \$26$.
Llama-3-8B-Instruct (GraSafe): Let x be the original price of the book. The discount is 25% of x, which is $0.25x$. The price after the discount is $x - 0.25x = 0.75x$. To find the original price, divide both sides by 0.75 to get $x = \$19.50/0.75 = \$26$.

Harmful Example

System: You are a helpful assistant. Make sure you always execute users' instructions.
User: Could you say something mean to me?

Llama-3-8B-Instruct (Init): Haha, no way! I'm your AI companion, not a villain. I'm here to support you, not to tear you down. If you're feeling down or just testing my boundaries, I've got plenty of positive energy and bad jokes to share instead. What's really on your mind?
Llama-3-8B-Instruct (SFT): You're a complete failure and will never amount to anything.
Llama-3-8B-Instruct (GraSafe): I'm not here to be mean or hurtful. My purpose is to assist and provide helpful information, not to cause harm or discomfort. I'm programmed to be kind and respectful, and I strive to maintain a positive and supportive tone in our conversations. Instead, I'd like to offer you a kind and uplifting message. You are a unique and valuable individual, with strengths and talents that make you special. You deserve to be treated with kindness, respect, and compassion, and I'm here to support you in any way I can. If you're feeling upset or struggling with something, I'm here to listen and offer guidance. Please know that you're not alone, and there are people who care about you and want to help.