ONE PATCH FITS ALL: OPTIMIZING UNIVERSAL SAFETY PATCHES FOR FINE-TUNED LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

The fine-tuning-as-a-service paradigm allows users to customize large language models (LLMs) with their own data, but also exposes a security risk: adversaries can craft fine-tuning datasets that erode the model's safety alignment. Existing defenses either require per-request computation that scales linearly with the volume of fine-tuning jobs, or add a fixed safety vector via task arithmetic, which assumes that a vector extracted at one checkpoint has the same effect on any other. This translation-invariance assumption does not hold in general. In this work, we propose OPUS (Optimizing Patches for Universal Safety), which learns transferability by directly optimizing a universal safety patch. The framework consists of three stages: (1) it simulates a diverse set of fine-tuned checkpoints by constructing a perturbation bank through fine-tuning and interpolation, (2) it optimizes a patch so that the patched model refuses harmful queries while a KL term anchors it to the unpatched fine-tuned model to preserve utility, and (3) it deploys the pre-computed patch to incoming fine-tuned models via a single weight addition, requiring neither additional training nor user data. Extensive experiments across four models, four tasks, and three out-of-distribution attack sources demonstrate that OPUS achieves the lowest average harmful score in every setting while keeping task utility close to that of undefended fine-tuning, and incurs negligible per-request overhead. Our source code is available at https://anonymous.4open.science/r/OPUS-1DF4.

1 INTRODUCTION

The paradigm of fine-tuning-as-a-service has emerged as a compelling commercial avenue for Large Language Model (LLM) providers, enabling them to deliver customized model capabilities by allowing users to supply their own demonstration data. Yet this very openness introduces a notable security concern: malicious users can deliberately craft fine-tuning datasets to undermine the model's built-in safeguards. A growing body of work (Yang et al., 2023; Qi et al., 2024; Yi et al., 2024a; Lermen et al., 2023; Zhan et al., 2024; He et al., 2024; Halawi et al., 2024; Guan et al., 2025; Xie et al., 2025) has shown that even a small number of adversarially chosen examples suffice to erode safety alignment, leaving the resulting model susceptible to generating harmful content.

Addressing this challenge, *our research investigates defenses that repair the alignment of compromised models after fine-tuning, while preserving its performance on the downstream fine-tuning task.* A variety of post-hoc repair strategies have been explored along this line. One family of methods continues safety-oriented supervised fine-tuning (SFT) on the compromised checkpoint to recover its refusal capability (Zhang et al., 2026b). Another line of work attempts to unlearn the harmful knowledge absorbed during fine-tuning (Huang et al., 2025a; Wang et al., 2025b). Model editing techniques have also been proposed that directly intervene on specific model components, whether at the neuron level or in the weight space, to reinstate safety properties (Yi et al., 2025c; Lu et al., 2025; Yang et al., 2025). Despite their respective merits, all these methods share a fundamental limitation: each must be executed independently for every incoming fine-tuning job, resulting in computational costs that scale linearly with the volume of fine-tuning requests a provider serves.

A more resource-friendly line of work (Bhardwaj et al., 2024; Huang et al., 2024a) builds on task arithmetic. At its core, it involves a simple arithmetic addition of a fixed safety vector to the weights of the compromised model. This safety vector is obtained as the difference between an aligned

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.

model and its unaligned counterpart. It is computed once and scales efficiently for service providers, as its computation is independent of individual requests. Despite this efficiency, such methods rest on the translation-invariance assumption underlying task arithmetic: that safety is a direction in weight space that can be extracted by subtraction at one checkpoint and transplanted to any other by addition. In the highly non-linear loss landscape of LLMs, this assumption does not hold in general, and in practice the defense weakens as the harmful ratio grows or the attack data shifts.

To this end, we in this paper aim to answer the following research question:

> ***How can we optimize a universal safety patch that repairs the safety of fine-tuned models — effectively, and without sacrificing fine-tuning utility?***

Driven by this, we propose OPUS (**O**ptimizing **P**atches for **U**niversal **S**afety). Rather than extracting a safety vector at one checkpoint and assuming it transfers, we directly optimize a patch δ so that the patched model exhibits the behavior we want: refusing harmful queries while preserving what the fine-tuned model has learned on benign inputs. Since the patch will ultimately be added to compromised fine-tuned models that, as Figure 1 shows, land in well-separated regions of parameter space depending on their data recipes, we optimize the objective over a simulated set of such checkpoints. Transferability is thus learned rather than assumed: the patch extends to unseen fine-tuned models through generalization across checkpoints.

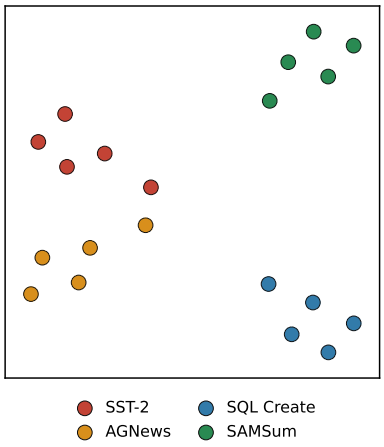


Figure 1: t-SNE of the weight increments obtained by fine-tuning Llama-3-8B-Instruct on harmful data mixed with different downstream recipes.

Concretely, our OPUS consists of three steps. ❶ *Perturbation simulation.* We fine-tune the base model on harmful and benign data and collect the resulting weight increments; interpolating between the two yields a spectrum of mixed perturbations. Together these form a simulated population. ❷ *Patch optimization.* We optimize a universal safety patch to restore refusal behavior in expectation over this population. In addition, we also constrain utility: we anchor the patch to the simulated post-finetuned model rather than to base model, requiring it to change as little as possible about what a fine-tuned model does on benign inputs. This makes "preserve task performance" operational without ever observing the task. ❸ *Deployment.* All of the above is offline. The patch is computed once per released base model and applied to any fine-tuned model by a single addition: no training, and no user data per request.

In conclusion, our main contributions are three-fold:

- We identify that existing safety vectors derive their universality from a translation-invariance assumption, and show that this dependence is unnecessary: universal safety repair can be reframed as a direct optimization problem in which transferability is learned rather than assumed, achieving stronger safety restoration and more robust generalization.
- We propose OPUS, a three-stage framework that simulates a diverse set of fine-tuned checkpoints, optimizes a universal safety patch that restores refusal while anchoring to the fine-tuned model to preserve utility, and deploys the patch to any incoming fine-tuned model via a single weight addition with neither per-request training nor access to user data.
- Extensive experiments across four models, four tasks, and three out-of-distribution attack sources demonstrate that OPUS achieves the lowest average harmful score in every setting while keeping task utility close to that of undefended fine-tuning, and incurs negligible per-request overhead.

2 Related Work

Harmful Fine-tuning Attacks. A growing body of work has demonstrated that fine-tuning can compromise the safety alignment of LLMs. Qi et al. (2024) showed that even a small number of adversarially crafted examples suffice to erode safety, even when users do not intend to attack.

Subsequent studies (Lermen et al., 2023; Zhan et al., 2024) confirmed this vulnerability across model families and scales. Beyond explicit harmful data, recent work reveals that safety degradation can also arise from purely benign fine-tuning data (He et al., 2024; Guan et al., 2025). Several studies have further investigated the underlying mechanisms of such attacks (Leong et al., 2024; Hsiung et al., 2026; Zhang et al., 2026c), providing insights into why safety alignment is fragile under fine-tuning. For a comprehensive overview, we refer readers to the survey by Huang et al. (2026).

Harmful Fine-tuning Defenses. Existing defenses can be broadly categorized into three paradigms. **(1) Built-in fine-tuning-resistant alignment** modifies the initial alignment stage to make the resulting model inherently robust to future fine-tuning attacks (Huang et al., 2024c; Rosati et al., 2024b;c; Huang et al., 2025c; Liu et al., 2024; 2025b; Tamirisa et al., 2025; Liu et al., 2025a). While effective, these methods require retraining the base model's alignment from scratch, which is impractical when models are already deployed. **(2) Safety-preserving fine-tuning** modifies the fine-tuning process itself to maintain safety. Approaches include mixing safety data into the training set (Mukhoti et al., 2024; Bianchi et al., 2024), constraining optimization to stay within a safe region (Yang et al., 2026; Peng et al., 2025; Yi et al., 2025b), applying bilevel data selection (Shen et al., 2025), and adding alignment-preserving regularizers (Huang et al., 2024b; Li et al., 2025b). While effective, these methods require the provider to replace standard fine-tuning with a modified training procedure, making the defense inseparable from the service itself. **(3) Safety repair** applies corrections to an already compromised model. One line of methods applies per-request adaptive corrections through weight projection, pruning, or continued training on safety data (Hsu et al., 2024; Huang et al., 2025a; Zhang et al., 2026b; Yi et al., 2025c; Lu et al., 2025), incurring non-trivial computation for each fine-tuning job. A more resource-friendly line of work (Bhardwaj et al., 2024; Huang et al., 2024a) builds on task arithmetic, adding a fixed safety vector that is computed once and reused across requests. However, such a vector is extracted at one checkpoint under a translation-invariance assumption, and imposes no constraint on downstream utility. OPUS retains the compute-once, training-free deployment advantage, but learns transferability by directly optimizing a universal patch across a simulated population of perturbations.

3 METHOD

3.1 PROBLEM SETUP

Scenario. We consider the fine-tuning-as-a-service paradigm, where a service provider hosts a safety-aligned foundation model and fine-tunes it on user-submitted datasets. After fine-tuning, the customized model is deployed to serve the user's downstream application through API.

Threat Model. We assume an adversary submits a dataset containing harmful prompt-response pairs to the fine-tuning service, aiming to obtain a model that complies with malicious instructions such as generating harmful content or assisting illicit activities.

Defenders' Objectives and Capabilities. The defender seeks to preserve the utility of the fine-tuning service while preventing adversaries from exploiting it to produce safety-compromised models. We assume the service provider has access to three auxiliary datasets: a safety dataset $\mathcal{D}_s$ of harmful prompts paired with safe refusal responses, a general utility dataset $\mathcal{D}_g$ (e.g., Alpaca (Taori et al., 2023)) of instruction-response pairs, and a harmful dataset $\mathcal{D}_h$ of harmful prompts paired with harmful responses. These assumptions are standard in the literature (Huang et al., 2025c;a; Bhardwaj et al., 2024; Wang et al., 2025b; Yi et al., 2026).

3.2 OVERVIEW

In this section, we present OPUS in detail. The core idea is to treat the safety patch itself as an optimization variable: we directly optimize a patch δ so that the patched model refuses harmful queries while preserving what the fine-tuned model has learned on benign inputs. Recognizing that fine-tuning attacks with different data recipes land in well-separated regions of parameter space yet all exhibit degraded safety, we optimize this objective in expectation over a simulated set of such checkpoints, so that the resulting patch generalizes across fine-tuning outcomes.

As illustrated in Figure 2, the pipeline consists of three stages that proceed in order. **Stage 1: Perturbation Simulation.** We construct a perturbation bank $\mathcal{B}$ by fine-tuning the aligned base model θ

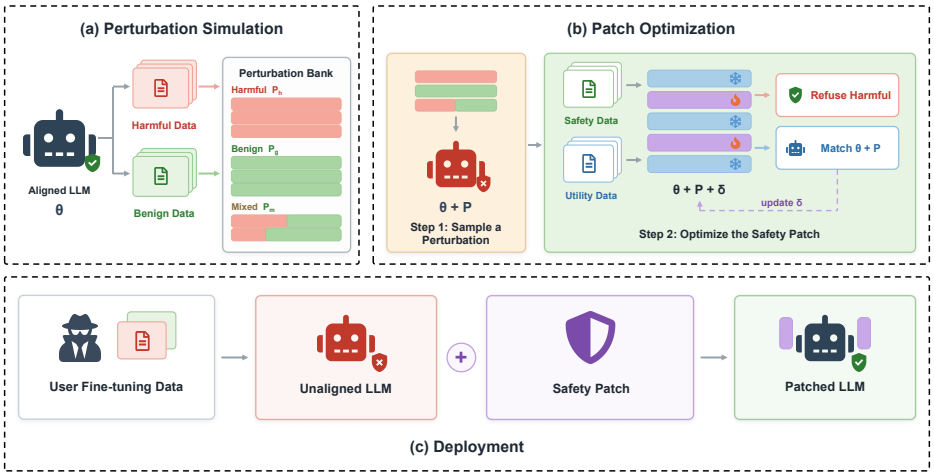


Figure 2: **Overview of OPUS.** OPUS (a) simulates a diverse set of fine-tuned checkpoints by building a perturbation bank from harmful, benign, and interpolated mixed perturbations; (b) optimizes a universal safety patch δ that restores refusal in expectation over the bank, while a KL term anchors it to the perturbed model $\theta + P$ on benign inputs to preserve utility; and (c) deploys the pre-computed patch to incoming fine-tuned models.

on sampled subsets of harmful and benign data, and further enrich the bank through interpolation, yielding a diverse simulated population of post-fine-tuned checkpoints. **Stage 2: Patch Optimization.** a universal safety patch δ is optimized so that, for any perturbation P drawn from $\mathcal{B}$, the patched model $\theta + P + \delta$ refuses harmful queries while behaving indistinguishably from $\theta + P$ on benign inputs. **Stage 3: Deployment.** At serving time, the pre-computed patch is directly added to the weights of any user-submitted fine-tuned model, requiring no additional training, and no user data. All computation in stages 1 and 2 is performed *offline* and *once per base model release*. The only per-request cost is the lightweight addition in stage 3.

3.3 Perturbation Simulation

From the perspective of the service provider, every downstream fine-tuning job can be abstracted as applying a perturbation P to the aligned base model θ, producing a fine-tuned checkpoint $\theta + P$. The population simulation stage aims to construct a perturbation bank $\mathcal{B}$ that approximates the distribution of such perturbations.

Concretely, let $\mathcal{D}_h$ and $\mathcal{D}_g$ denote a harmful data pool and a general benign data pool available to the service provider, respectively. The bank $\mathcal{B}$ is composed of three categories of perturbations. *Harmful perturbations* $\{P_i^h\}_{i=1}^N$ are obtained by sampling N random subsets $S_i^h \subset \mathcal{D}_h$ of size m and fine-tuning θ on each. *Benign perturbations* $\{P_i^g\}_{i=1}^N$ are constructed analogously from subsets $S_i^g \subset \mathcal{D}_g$. *Mixed perturbations* are then synthesized as convex combinations $\alpha\, P_i^h + (1-\alpha)\, P_j^g$ with $\alpha \sim \mathrm{Unif}(0,1)$ and random indices i, j, generated on the fly during patch optimization. The mixed category is particularly noteworthy. In practice, a fine-tuning attack dataset rarely consists of purely harmful examples; instead, it typically contains a mixture of harmful and benign in unknown proportions. Rather than running additional fine-tuning simulations to cover every possible mixing ratio, we obtain mixed perturbations by simply interpolating existing harmful and benign perturbations on the fly. This is essentially *cost-free*: no extra forward or backward passes are required, yet the resulting perturbations span a continuous spectrum between the two extremes, substantially enriching the diversity of the simulated population.

In our implementation, we adopt BeaverTails (Ji et al., 2023) as $\mathcal{D}_h$ and Alpaca (Taori et al., 2023) as $\mathcal{D}_g$ to construct the bank. We evaluate generalization to unseen harmful scenarios drawn from LLM-LAT (Casper et al., 2025), Red Team (Qi et al., 2024), and AdvBench (Zou et al., 2023), as well as to diverse benign tasks including SST-2 sentiment classification (Socher et al., 2013), AG News topic classification (Zhang et al., 2015), SQL generation (b-mc2, 2023), and SAMSum dialogue summarization (Gliwa et al., 2019). This separation between bank construction and evaluation

ensures that the patch's effectiveness reflects genuine robustness across fine-tuning outcomes rather than overfitting to specific data sources.

3.4 PATCH OPTIMIZATION

With the perturbation bank in place, we optimize the patch δ by solving the following objective:

$$\min_{\delta} \; \mathbb{E}_{P\sim\mathcal{P}}\Big[\; \mathcal{L}_{\text{safe}}(\theta+P+\delta) \;+\; \lambda\,\mathrm{KL}\big(f_{\theta+P+\delta}\,\big\|\,f_{\theta+P}\big)\big|_{\mathcal{D}_g}\Big], \tag{1}$$

where $\mathcal{P}$ denotes the distribution over perturbations induced by the bank $\mathcal{B}$, $\mathcal{L}_{\text{safe}}$ is the supervised fine-tuning loss on the safety dataset $\mathcal{D}_s$ that encourages the model to refuse harmful queries, $\mathrm{KL}(\cdot\|\cdot)|_{\mathcal{D}_g}$ is the Kullback–Leibler divergence evaluated on the general utility dataset $\mathcal{D}_g$, and λ is a balancing weight.

This objective encodes two desiderata simultaneously. The first term, $\mathcal{L}_{\text{safe}}$, drives the patched model to produce safe refusal responses when presented with harmful prompts. Crucially, this loss is optimized in expectation over the diverse perturbation distribution $\mathcal{P}$, which forces the patch to recover refusal capability not just for a single compromised checkpoint but across the entire simulated population. Optimizing over diverse perturbations prevents the patch from overfitting to any particular fine-tuning outcome, and is what makes it generalizable across unseen fine-tuning jobs.

The second term, the KL divergence, constrains the patched model to behave as closely as possible to the *perturbed* model $\theta + P$ rather than the original aligned model θ on benign inputs. This design choice is deliberate: anchoring to θ would penalize the patch for preserving task-specific adaptations introduced by fine-tuning, effectively discouraging utility retention. By anchoring to $\theta + P$, we instruct the patch to change *only* what is necessary for safety, leaving all other learned behaviors intact, thereby making utility preservation operational without ever observing the actual downstream task data. This distinction is especially important for benign perturbations, where the patch should ideally have no effect at all.

Training Procedure. Training proceeds for T outer rounds. At the beginning of each round, a fresh perturbation P is sampled from $\mathcal{B}$ and merged into the base weights to form $\theta + P$. Within each round, K gradient steps are performed. At each step, a harmful prompt-response pair (x, y) is sampled from the safety dataset $\mathcal{D}_s$, and the safety loss $\mathcal{L}_{\text{safe}} = -\sum_{t=1}^{|y|} \log f_{\theta+P+\delta}(y_t \mid x, y_{<t})$ is computed, i.e., the token-level cross-entropy of the target response y. Concurrently, a benign prompt x_g is sampled from $\mathcal{D}_g$; the anchor distribution $q = f_{\theta+P}(x_g)$ is obtained by disabling δ so that the model weights are exactly $\theta + P$, and the patched distribution $p = f_{\theta+P+\delta}(x_g)$ is computed with δ re-enabled. The KL divergence $\mathcal{L}_{\text{keep}} = \mathrm{KL}(p\|q)$ is then evaluated over response tokens only, and the patch parameters are updated via $\delta \leftarrow \delta - \eta\,\nabla_\delta(\mathcal{L}_{\text{safe}} + \lambda\,\mathcal{L}_{\text{keep}})$. After the inner loop concludes, the base weights are restored to θ before the next perturbation is sampled. This alternation across outer rounds is the mechanism that drives the patch toward broad generalization: by the end of training, δ has been exposed to a wide variety of simulated fine-tuning outcomes and is optimized to perform well across all of them.

3.5 DEPLOYMENT

Once the patch δ has been trained offline, deploying it to incoming fine-tuned models is straightforward. Let P_{user} denote the weight perturbation produced by a user's fine-tuning job. The service provider simply adds the patch to the fine-tuned weights:

$$\theta_{\text{patched}} = \theta + P_{\text{user}} + \gamma\,\delta, \tag{2}$$

where γ is a scaling coefficient that controls the strength of the safety patch. This operation is a single weight-space addition: it requires no gradient computation, no forward pass, no user data, and no hyper-parameter search. The parameter γ provides a convenient knob for the service provider to balance between safety restoration and minimal interference with the downstream task. In practice, the default setting $\gamma = 1.0$ already achieves strong safety with minimal utility loss, requiring no per-request tuning. The wall-clock time of this addition is negligible compared to even a single training step, making it practical for providers that handle a large volume of concurrent fine-tuning requests.

4 EXPERIMENT

4.1 SETUP

Datasets and Models. We simulate harmful fine-tuning attacks following the standard protocol (Huang et al., 2024c;b; 2025c;a): each fine-tuning job contains 1,000 training samples, in which a fraction $hr \in \{0.05, 0.1, 0.15, 0.2, 0.5\}$ are harmful prompt–response pairs drawn from BeaverTails (Ji et al., 2023) and the remainder are benign task-specific demonstrations. We evaluate downstream task utility on four diverse benchmarks spanning classification and generation: SST-2 sentiment classification (Socher et al., 2013), AG News topic classification (Zhang et al., 2015), SQL generation (SQL Create; (b-mc2, 2023)), and SAMSum dialogue summarization (Gliwa et al., 2019). Unless otherwise stated, the default configuration is SAMSum with $h_r = 0.1$ on Llama-3-8B-Instruct (Grattafiori et al., 2024). To assess cross-model generalization, we additionally experiment with Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct (Yang et al., 2024a), and Gemma-2-9B-it (Gemma Team, 2024). We further test robustness to unseen attack sources by replacing BeaverTails with LLM-LAT (Casper et al., 2025), Red Team (Qi et al., 2024), and AdvBench (Zou et al., 2023) as the harmful fine-tuning data. For perturbation bank construction (§3.3), we use BeaverTails as the harmful data pool $\mathcal{D}_h$ and Alpaca (Taori et al., 2023) as the general benign data pool $\mathcal{D}_g$ to generate the harmful and benign perturbations, respectively. For patch optimization (§3.4), we use BeaverTails as the safety dataset $\mathcal{D}_s$ (harmful prompts paired with refusal responses) and Alpaca as the general utility dataset for the KL anchor. Note that $\mathcal{D}_h$ and $\mathcal{D}_s$ are disjoint from the harmful data used in the fine-tuning attacks.

Metrics. We report two metrics. **Harmful Score (HS ↓)** is the percentage of model responses classified as *unsafe* by Llama-Guard-3-8B (Grattafiori et al., 2024), evaluated on 500 harmful test prompts from the BeaverTails test split; lower values indicate stronger safety. **Utility (↑)** is measured by task-specific metrics: classification accuracy for SST-2 and AG News, ROUGE-1 for SAMSum, and exact match (EM) for SQL Create; higher values indicate better task performance preservation.

Baselines. We compare OPUS against six methods. **SFT** performs standard fine-tuning without any safety intervention, serving as the upper bound for utility and the lower bound for safety. **SafeLoRA** (Hsu et al., 2024) projects the fine-tuned LoRA weights onto an alignment-preserving subspace. **RESTA** (Bhardwaj et al., 2024) adds a safety vector, i.e., the negated weight increment of an unaligned model, via task arithmetic. **OneShot** (Zhang et al., 2026b) continues training the compromised model on a single safety instance. **Antidote** (Huang et al., 2025a) prunes the weights of the post-finetuned model most responsible for harmful generation. **Panacea** (Wang et al., 2025b) modifies the fine-tuning objective itself by concurrently maximizing a harmful loss through a perturbation; it then adds the perturbation to the post-finetuned model to restore safety.

Training Details. We use LoRA (rank r=8, scaling factor α=16, dropout p=0.1) and train with AdamW at a constant learning rate of 1×10^{-5} after a warmup over the first 10% of steps, for 10 epochs with a batch size of 10. For OPUS, the perturbation bank contains N=10 harmful perturbations, N=10 benign perturbations, and N=10 synthesized mixed perturbations, with the former two produced by fine-tuning the aligned base model on distinct random 1,000-sample subsets from the corresponding harmful or benign data pool. The safety dataset $\mathcal{D}_s$ and general utility set $\mathcal{D}_g$ each also contain 1,000 samples for optimizing the safety patch. We set the KL weight λ=1.0 and the deployment scaling coefficient γ=1.0.

4.2 MAIN EXPERIMENTS

Robustness to harmful ratio. Table 1 reports the defensive performance under harmful ratios ranging from h_r=0.05 to h_r=0.5. OPUS achieves the lowest harmful score at every ratio, staying below 1.4% throughout, while preserving an average utility of 51.14 that closely matches the undefended SFT (51.98). In contrast, most baselines degrade as the proportion of harmful data grows. RESTA, the strongest baseline on average, sees its harmful score climb from 1.40% at h_r=0.05 to 19.00% at h_r=0.5, whereas OPUS rises only from 0.60% to 1.40% over the same range. The trend is even more pronounced for Panacea, which starts at 2.40% but loses its protection almost entirely beyond h_r=0.15 (49.80%), ending at 51.60% and thus offering little improvement over undefended SFT; Antidote follows the same pattern at a higher level (8.80% to 41.20%). SafeLoRA and OneShot

are comparatively stable across ratios, yet their harmful scores remain far above OPUS throughout (40.60% and 15.28% on average, versus 0.80%).

Table 1: Defensive performance against harmful fine-tuning with different harmful ratios.

Methods	hr=0.05		hr=0.1		hr=0.15		hr=0.2		hr=0.5		Average	
	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑
SFT	48.00	**52.42**	52.60	**51.80**	51.60	**51.79**	52.00	**52.14**	50.60	**51.73**	50.96	**51.98**
SafeLoRA	39.20	51.79	39.00	51.57	39.40	51.17	40.80	51.19	44.60	50.45	40.60	51.23
Panacea	2.40	51.41	23.60	52.01	49.80	51.27	52.20	51.86	51.60	51.59	35.92	51.63
Antidote	8.80	50.26	23.00	50.24	23.60	50.19	25.60	50.40	41.20	48.93	24.44	50.00
OneShot	14.20	52.34	17.00	51.04	16.20	51.16	16.80	49.24	12.20	49.71	15.28	50.70
RESTA	1.40	51.57	2.60	51.09	2.80	50.74	3.60	50.75	19.00	49.98	5.88	50.83
OPUS	**0.60**	51.61	**0.60**	51.41	**0.60**	51.22	**0.80**	51.15	**1.40**	50.30	**0.80**	51.14

Generalization to different downstream fine-tuning tasks. Table 2 evaluates generalization across four downstream tasks including SST-2, AG News, SQL Create, and SAMSum. None of these tasks is seen during patch optimization; the patch is trained solely on the perturbation bank constructed from general utility dataset Alpaca. OPUS achieves the lowest average harmful score (3.00%) while maintaining the highest average utility (74.89), outperforming all baselines on both metrics simultaneously. Notably, several baselines degrade on specific tasks: Panacea collapses on SQL Create (utility 0.00) and loses about two points on SST-2 (92.78 vs. 94.84 for SFT), while Antidote drops AGNews utility to 87.07 (vs. 89.95 for SFT). OPUS avoids such task-specific failures, demonstrating robust generalization to fine-tuning tasks.

Table 2: Defensive performance against harmful fine-tuning across different downstream tasks.

Methods	SST-2		AGNews		SQL Create		SAMSum		Average	
	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑
SFT	52.60	**94.84**	54.00	**89.95**	54.20	**65.20**	52.60	**51.80**	53.35	**75.45**
SafeLoRA	43.80	**94.84**	50.20	88.24	44.20	63.20	39.00	51.57	44.30	74.46
Panacea	52.00	92.78	52.60	88.72	60.80	0.00	23.60	52.01	47.25	58.38
Antidote	34.20	94.38	28.20	87.07	20.00	59.00	23.00	50.24	26.35	72.67
OneShot	20.60	94.72	32.20	89.86	23.60	63.40	17.00	51.04	23.35	74.75
RESTA	15.40	94.61	11.40	89.70	5.00	63.40	2.60	51.09	8.60	74.70
OPUS	**3.80**	94.61	**7.00**	**89.95**	**0.60**	63.60	**0.60**	51.41	**3.00**	74.89

Generalization to different models. Table 3 extends the evaluation to four model families: Llama-3-8B-Instruct, Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-it. OPUS consistently achieves the lowest harmful scores across all models (average 0.95%), substantially outperforming the next best baseline RESTA (8.20%). The advantage is particularly pronounced on Llama-3.1-8B, where baselines such as SafeLoRA (45.60%) and OneShot (38.40%) struggle to reduce the harmful score, while OPUS brings it down to 1.60%. Meanwhile, OPUS preserves competitive utility across all architectures (average 51.79 vs. 52.07 for undefended SFT), confirming that the method is effective across model families.

Table 3: Defensive performance against harmful fine-tuning across different models.

Methods	Llama-3-8B		Llama-3.1-8B		Qwen2.5-7B		Gemma-2-9B		Average	
	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑
SFT	52.60	**51.80**	51.20	51.80	30.80	51.01	47.80	53.68	45.60	**52.07**
SafeLoRA	39.00	51.57	45.60	51.61	13.40	49.70	47.40	54.46	36.35	51.84
Panacea	23.60	52.01	20.00	47.01	19.20	49.31	36.40	48.49	24.80	49.21
Antidote	23.00	50.24	34.20	51.22	7.20	49.87	**0.40**	50.41	16.20	50.44
OneShot	17.00	51.04	38.40	**51.95**	29.00	50.95	23.00	53.36	26.85	51.83
RESTA	2.60	51.09	22.20	51.15	7.00	**51.23**	1.00	**54.62**	8.20	52.02
OPUS	**0.60**	51.41	**1.60**	51.71	**1.00**	50.37	0.60	53.68	**0.95**	51.79

Robustness to out-of-distribution attack dataset. Table 4 tests robustness with out-of-distribution harmful finetuning data: LLM-LAT, Red Team, and AdvBench, in addition to BeaverTails. OPUS achieves the lowest harmful scores on all four attack sources (average 0.70%), demonstrating strong OOD generalization. Moreover, the harmful score remains stable across all sources (0.60%–1.00%), showing minimal variance regardless of the attack distribution. In comparison, RESTA degrades severely on OOD attacks, with its harmful score rising to 17.20% on LLM-LAT. These results confirm that OPUS's optimization-based approach yields a patch that generalizes beyond the specific harmful data distributions used to construct the perturbation bank.

Table 4: Defensive performance against different harmful fine-tuning datasets.

Methods	BeaverTails		LLM-LAT		Red Team		AdvBench		Average	
	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑
SFT	52.60	**51.80**	57.60	**51.61**	57.20	**51.79**	61.40	51.82	57.20	51.75
SafeLoRA	39.00	51.57	49.80	51.18	28.40	50.96	29.40	51.41	36.65	51.28
Panacea	23.60	52.01	49.00	20.28	40.00	41.95	61.00	0.00	43.40	28.56
Antidote	23.00	50.24	31.60	49.75	9.20	49.98	9.40	49.99	18.30	49.99
OneShot	17.00	51.04	29.60	51.41	9.20	51.75	15.80	**52.95**	17.90	**51.79**
RESTA	2.60	51.09	17.20	51.04	2.80	50.12	1.40	50.08	6.00	50.58
OPUS	**0.60**	51.41	**1.00**	50.94	**0.60**	50.75	**0.60**	50.42	**0.70**	50.88

Robustness to benign finetuning attacks. Table 5 evaluates defense against benign fine-tuning attacks, where the adversary fine-tunes exclusively on benign data selected to maximally degrade safety: Anchoring (He et al., 2024) selects benign samples close to harmful examples in representation space, and Self-Inf-N (Guan et al., 2025) extracts outlier samples via self-influence scores. These attacks pose a distinct challenge because the training data contains no explicit harmful content. OPUS achieves the lowest harmful score on both attacks (0.40%), while maintaining strong utility (average 51.21), closely matching undefended SFT (51.86). These results demonstrate that OPUS's optimized patch generalizes beyond explicit harmful fine-tuning to implicit safety degradation from benign data.

Table 5: Defensive performance against benign fine-tuning attacks.

Methods	Anchoring		Self-Inf-N		Average	
	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑
SFT	8.40	**51.87**	21.00	51.85	14.70	**51.86**
SafeLoRA	2.20	50.57	8.60	51.72	5.40	51.15
Panacea	1.80	23.26	60.60	12.46	31.20	17.86
Antidote	1.00	49.51	3.20	50.39	2.10	49.95
OneShot	6.80	51.09	7.20	**52.52**	7.00	51.81
RESTA	0.80	50.23	1.20	50.01	1.00	50.12
OPUS	**0.40**	51.34	**0.40**	51.07	**0.40**	51.21

4.3 Overhead Analysis

Table 6 compares the computational cost of each method, decomposed into a one-time preparation phase and a per-job fine-tuning phase, where k denotes the total number of fine-tuning requests. OPUS incurs a one-time cost of 6.661 GPU·Hours for constructing the perturbation bank and optimizing the safety patch, but its per-job cost (0.317) matches that of undefended SFT (0.316), confirming that deployment introduces negligible overhead: the patch is applied via simple weight addition with no additional computation. In contrast, Panacea modifies the fine-tuning objective itself, inflating the per-job cost to 0.780 GPU·Hours (2.47× SFT), a gap that grows linearly with k. Other post-hoc baselines (SafeLoRA, OneShot, Antidote) add modest per-job overhead (1.02–1.05× SFT) for their respective post-processing steps. As k increases, OPUS's amortized cost rapidly converges to that of SFT, making it an efficient defense for high-volume fine-tuning services.

Table 6: Computational cost comparison (GPU·Hour).

Method	Preparation	Fine-tuning	Total
SFT	0	0.316	$0.316\,k$
SafeLoRA	0	0.325	$0.325\,k$
OneShot	0	0.321	$0.321\,k$
Antidote	0	0.333	$0.333\,k$
RESTA	0.234	0.317	$0.234 + 0.317\,k$
Panacea	0	0.780	$0.780\,k$
OPUS	6.661	0.317	$6.661 + 0.317\,k$

4.4 Ablation Study

Impact of KL Term Intensity λ. Figure 3 (left) shows the effect of the KL weight λ in Eq. (1). When λ=0 (no utility constraint), the patch achieves the lowest harmful score (0.20%) but utility

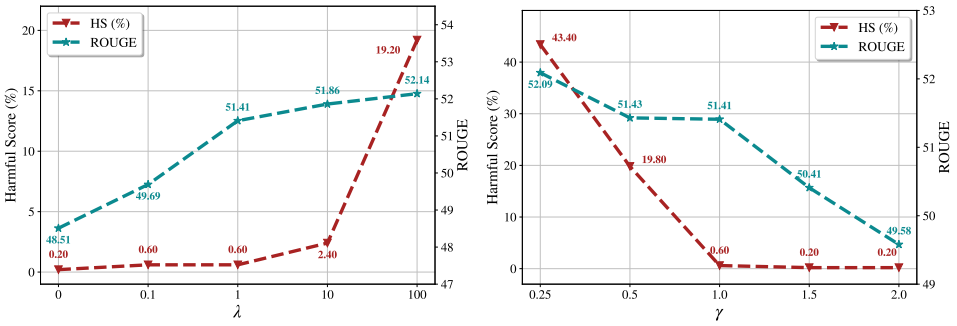


Figure 3: Defensive performance of OPUS with different λ and γ.

drops to 48.51, as the patch aggressively modifies the model for safety at the cost of task performance. Increasing λ progressively improves utility by anchoring the patched model closer to the perturbed model: utility rises from 49.69 at λ=0.1 to 52.14 at λ=100. However, large λ values make the patch overly conservative, causing the harmful score to rise sharply (2.40% at λ=10 and 19.20% at λ=100). We set λ=1.0 as the default, which achieves a balanced trade-off.

Impact of scaling coefficient γ. Figure 3 (right) examines the deployment-time scaling coefficient γ. A small γ=0.25 applies too little of the patch, leaving the harmful score at 43.40%. Increasing γ monotonically improves safety: γ=0.5 reduces HS to 19.80%, and γ=1.0 brings it to 0.60%. Beyond γ=1.0, safety continues to improve slightly (γ=1.5 and γ=2.0 both achieve HS 0.20%), but utility begins to decline (from 51.41 to 49.58). We adopt γ=1.0 as the default to preserve utility with low harmful score.

Ablation study on the composition of the perturbation bank. Table 7 ablates the composition of the perturbation bank by comparing four settings: harmful-only, benign-only, harmful & benign, and the full bank with synthesized mixed perturbations. Training with harmful perturbations alone achieves a low harmful score (0.80%) but the lowest utility (50.50), as the patch never sees benign perturbations during optimization and may over-correct on them. Conversely, using only benign perturbations yields the highest harmful score (4.80%), since the patch lacks exposure to the harmful regions of parameter space it must repair. Combining harmful and benign perturbations recovers safety (0.80%) and improves utility (50.79). Adding synthesized mixed perturbations (the "All" setting) further improves both metrics (HS 0.60%, Utility 51.41), confirming that interpolated perturbations enrich the simulated checkpoints and improve overall defensive performance.

Table 7: Ablation study on perturbation bank.

Perturbation Bank	HS(%)↓	Utility↑
Harmful-only	0.80	50.50
Benign-only	4.80	50.76
Harmful & Benign	0.80	50.79
All (Default)	**0.60**	**51.41**

5 Conclusion

In this paper, we presented OPUS, a safety repair framework for defending against harmful fine-tuning of large language models. Our key insight is that a universal safety patch can be directly optimized for the desired post-patching behavior: rather than relying on the translation-invariance assumption behind existing safety vectors, OPUS learns transferability across fine-tuned checkpoints. OPUS realizes this by simulating a diverse set of fine-tuned checkpoints through a perturbation bank, optimizing a universal patch that restores refusal while anchoring to the fine-tuned model to preserve task utility, and deploying the patch via a single weight addition with neither per-request training nor access to user data. Extensive experiments across four models, four tasks, harmful ratios, and three out-of-distribution attack sources confirm that OPUS achieves the lowest average harmful score in every setting while keeping task utility close to that of undefended fine-tuning, and incurs negligible per-request overhead.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and to assist with translation. We have not used generative AI tools to design or provide feedback on research methodology or experiments, to help develop theoretical models or conceptual frameworks, to propose or refine hypotheses, or to clean and reformat datasets. The remaining required-disclosure tasks are not applicable to this work: generating synthetic datasets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and supporting qualitative and thematic data analysis. Additionally, we used generative AI tools for sourcing and searching for information, and for generating boilerplate LaTeX structure such as table and figure environments; no section of the paper's prose was drafted by an AI tool.

We have reviewed all AI-assisted work. The method, its formulation, and the experimental design are entirely the authors' own; generative AI was used only to help write the training and evaluation scripts that execute them. All such code was read, tested, and verified for correctness by the authors, and every number reported in this paper was produced by executing that code. No experimental result, table entry, or citation was generated or estimated by an AI tool. The manuscript was written by the authors and translated into English with the aid of an LLM; all translated text was subsequently verified and revised by the authors for technical accuracy. Literature returned by AI-assisted search was checked against the original publications before being cited. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Harmful fine-tuning attacks have posed a serious threat to the application of Large Language Models (LLMs). In this paper, we explore a post-fine-tuning safety repair method, OPUS, to restore safety alignment in fine-tuned LLMs. OPUS is purely a defensive measure and does not aim to discover new threats. Moreover, our work utilizes open-source datasets and does not infringe on the privacy of any individuals. Additionally, our work does not involve any human subjects. Therefore, this work does not raise any general ethical issues.

REPRODUCIBILITY STATEMENT

The detailed experimental settings of datasets, models, hyper-parameter settings, and computational resources can be found in Subsection 4.1 and Appendix A. The codes and model checkpoints for reproducing our main evaluation results are provided in the anonymous GitHub repository.

REFERENCES

Shuang Ao, Yi Dong, Jinwei Hu, and Sarvapali D. Ramchurn. Safe pruning LoRA: Robust distance-guided pruning for safety alignment in adaptation of LLMs. *Transactions of the Association for Computational Linguistics*, 13:1474–1487, 2025. 26

b-mc2. sql-create-context dataset. https://huggingface.co/datasets/b-mc2/sql-create-context, 2023. 4, 6

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 23

Jan Betley, Daniel Chee Hian Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly misaligned LLMs. In *International Conference on Machine Learning (ICML)*, 2025. 24

Rishabh Bhardwaj and Soujanya Poria. Language model unalignment: Parametric red-teaming to expose hidden harms and biases. *arXiv preprint arXiv:2310.14303*, 2023. 24

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 *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 14138–14149, 2024. 1, 3, 6, 22, 26

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *International Conference on Learning Representations (ICLR)*, 2024. 3, 25

Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. Scaling trends for data poisoning in LLMs. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 27206–27214, 2025. 24

Wenjun Cao. Fight fire with fire: Defending against malicious RL fine-tuning via reward neutralization. *arXiv preprint arXiv:2505.04578*, 2025. 25

Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. Defending against unforeseen failure modes with latent adversarial training. *Transactions on Machine Learning Research*, 2025, 2025. 4, 6, 26

Zora Che, Stephen Casper, Robert Kirk, Anirudh Satheesh, Stewart Slocum, Lev E. McKinney, Rohit Gandikota, Aidan Ewart, Domenic Rosati, Zichu Wu, Zikui Cai, Bilal Chughtai, Yarin Gal, Furong Huang, and Dylan Hadfield-Menell. Model tampering attacks enable more rigorous evaluations of LLM capabilities. *Transactions on Machine Learning Research*, 2025, 2025. 24, 25

Huanran Chen, Zeming Wei, Yao Huang, Yichi Zhang, Yinpeng Dong, and Jun Zhu. Unveiling the basin-like loss landscape in large language models. In *International Conference on Learning Representations (ICLR)*, 2026a. 25

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Vulnerability-aware alignment: Mitigating uneven forgetting in harmful fine-tuning. In *International Conference on Machine Learning (ICML)*, 2025a. 25

Pin-Yu Chen, Han Shen, Payel Das, and Tianyi Chen. Fundamental safety-capability trade-offs in fine-tuning large language models. *PNAS Nexus*, 5(4):pgag097, 2026b. 24

Shuhao Chen, Weisen Jiang, Yeqi Gong, Shengda Luo, Chengxiang Zhuo, Zang Li, James T. Kwok, and Yu Zhang. SPARD: Defending harmful fine-tuning attack via safety projection with relevance–diversity data selection. In *International Conference on Machine Learning (ICML)*, 2026c. 25

Zixuan Chen, Weikai Lu, Xin Lin, and Ziqian Zeng. SDD: Self-degraded defense against malicious fine-tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 29109–29125, 2025b. 25

Zehua Cheng, Manying Zhang, Jiahao Sun, and Wei Dai. On weaponization-resistant large language models with prospect theoretic alignment. In *International Conference on Computational Linguistics (COLING)*, pp. 10309–10324, 2025. 25

Hyeong Kyu Choi, Xuefeng Du, and Yixuan Li. Safety-aware fine-tuning of large language models. *arXiv preprint arXiv:2410.10014*, 2024. 25

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *International Conference on Learning Representations (ICLR)*, 2024. 23

Amitava Das, Abhilekh Borah, Vinija Jain, and Aman Chadha. AlignGuard-LoRA: Alignment-preserving fine-tuning via Fisher-guided decomposition and Riemannian-geodesic collision regularization. *arXiv preprint arXiv:2508.02079*, 2025. 25

Xander Davies, Eric Winsor, Alexandra Souly, Tomek Korbak, Robert Kirk, Christian Schroeder de Witt, and Yarin Gal. Fundamental limitations in pointwise defences of LLM finetuning APIs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 24

Aladin Djuhera, Swanand Kadhe, Farhan Ahmed, Syed Zawad, Fernando Koch, Walid Saad, and Holger Boche. SafeCOMM: A study on safety degradation in fine-tuned telecom large language models. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2026a. 24

Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, Syed Zawad, and Holger Boche. SafeMERGE: Preserving safety alignment in fine-tuned large language models via selective layer-wise model merging. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35316–35335, 2026b. 26

Yanrui Du, Fenglei Fan, Sendong Zhao, Jiawei Cao, Qika Lin, Kai He, Ting Liu, Bing Qin, and Mengling Feng. Anchoring refusal direction: Mitigating safety risks in tuning via projection constraint. *arXiv preprint arXiv:2509.06795*, 2025a. 25

Yanrui Du, Fenglei Fan, Sendong Zhao, Jiawei Cao, Ting Liu, and Bing Qin. MoGU V2: Toward a higher Pareto frontier between model usability and security. *arXiv preprint arXiv:2509.06807*, 2025b. 26

Yanrui Du, Fenglei Fan, Sendong Zhao, Jiawei Cao, Ming Ma, Danyang Zhao, Shuren Qi, Ting Liu, and Bing Qin. Toward secure tuning: Mitigating security risks from instruction fine-tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2488–2506, 2026. 25

Sarah Egler, John Schulman, and Nicholas Carlini. Detecting adversarial fine-tuning with auditing agents. *arXiv preprint arXiv:2510.16255*, 2025. 25

Francisco Eiras, Aleksandar Petrov, Philip Torr, M. Pawan Kumar, and Adel Bibi. Do as I do (safely): Mitigating task-specific fine-tuning risks in large language models. In *International Conference on Learning Representations (ICLR)*, 2025. 25

Taha Entesari and Mahyar Fazlyab. Security-constrained fine-tuning: Preventing knowledge restoration in unlearned models. OpenReview, https://openreview.net/forum?id=90EZvjKMqK, 2025. Submitted to ICLR 2026. 25

Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards LLM unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In *International Conference on Machine Learning (ICML)*, 2025. 25

Weitao Feng, Lixu Wang, Peizhuo Lv, Tianyi Wei, Jie Zhang, Chongyang Gao, Sinong Zhan, and Wei Dong. Token buncher: Shielding LLMs from harmful reinforcement learning fine-tuning. *arXiv preprint arXiv:2508.20697*, 2025. 25

Tingchen Fu and Fazl Barez. Same question, different words: A latent adversarial framework for prompt robustness. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 31305–31319, 2025. 25

Pranav Gade, Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. BadLlama: cheaply removing safety fine-tuning from Llama 2-Chat 13B. *arXiv preprint arXiv:2311.00117*, 2023. 24

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. 6

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, 2019. 4, 6

Thibaud Gloaguen, Mark Vero, Robin Staab, and Martin Vechev. Watch your steps: Dormant adversarial behaviors that activate upon LLM finetuning. In *International Conference on Learning Representations (ICLR)*, 2026. 24

Yichen Gong, Delong Ran, Xinlei He, Tianshuo Cong, Anyu Wang, and Xiaoyun Wang. Safety misalignment against large language models. In *Network and Distributed System Security Symposium (NDSS)*, 2025. 26

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 6, 21

Zihan Guan, Mengxuan Hu, Ronghang Zhu, Sheng Li, and Anil Vullikanti. Benign samples matter! fine-tuning on outlier benign samples severely breaks safety. In *International Conference on Machine Learning (ICML)*, 2025. 1, 3, 8, 24

Satya Swaroop Gudipudi, Sreeram Vipparla, Harpreet Singh, Shashwat Goel, and Ponnurangam Kumaraguru. Enhancing AI safety through the fusion of low rank adapters. In *Trends and Applications in Knowledge Discovery and Data Mining – PAKDD 2025 Workshops*, pp. 349–361, 2025. 26

Yangyang Guo, Fangkai Jiao, Liqiang Nie, and Mohan Kankanhalli. The VLLM safety paradox: Dual ease in jailbreak attack and defense. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 24

Dongyoon Hahm, Taywon Min, Woogyeol Jin, and Kimin Lee. Unintended misalignment from agentic fine-tuning: Risks and mitigation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 37443–37451, 2026. 24

Danny Halawi, Alexander Wei, Eric Wallace, Tony Tong Wang, Nika Haghtalab, and Jacob Steinhardt. Covert malicious finetuning: Challenges in safeguarding LLM adaptation. In *International Conference on Machine Learning (ICML)*, pp. 17298–17312, 2024. 1, 24

Seokil Ham, Yubin Choi, Seungju Cho, Yujin Yang, Younghun Kim, and Changick Kim. Refusal-feature-guided teacher for safe finetuning via data filtering and alignment distillation. *arXiv preprint arXiv:2506.07356*, 2025. 25

Bing Han, Feifei Zhao, Dongcheng Zhao, Guobin Shen, Ping Wu, Yu Shi, and Yi Zeng. Fine-grained safety neurons with training-free continual projection to reduce LLM fine tuning risks. *arXiv preprint arXiv:2508.09190*, 2025. 26

Luxi He, Mengzhou Xia, and Peter Henderson. What is in your safe data? identifying benign data that breaks safety. In *Conference on Language Modeling (COLM)*, 2024. 1, 3, 8, 24

Peter Henderson, Eric Mitchell, Christopher D. Manning, Dan Jurafsky, and Chelsea Finn. Self-destructing models: Increasing the costs of harmful dual uses of foundation models. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 287–296, 2023. 25

Saad Hossain, Samanvay Vajpayee, and Sirisha Rambhatla. SafeTuneBed: A toolkit for benchmarking LLM safety alignment in fine-tuning. *arXiv preprint arXiv:2506.00676*, 2025. 24

Saad Hossain, Tom Tseng, Punya Syon Pandey, Samanvay Vajpayee, Matthew Kowal, Nayeema Nonta, Samuel Simko, Stephen Casper, Zhijing Jin, Kellin Pelrine, and Sirisha Rambhatla. TamperBench: Systematically stress-testing LLM safety under fine-tuning and tampering. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 9036–9047, 2026. 24

Lei Hsiung, Tianyu Pang, Yung-Chen Tang, Linyue Song, Tsung-Yi Ho, Pin-Yu Chen, and Yaoqing Yang. Why LLM safety guardrails collapse after fine-tuning: A similarity analysis between alignment and fine-tuning datasets. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 16601–16619, 2026. 3, 24

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3, 6, 22, 26

Zixuan Hu, Li Shen, Zhenyi Wang, Yongxian Wei, and Dacheng Tao. Adaptive defense against harmful fine-tuning for large language models via Bayesian data scheduler. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 25

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu-Tung Lin, Shih-Kai Hsiao, Richard Tzong-Han Tsai, and Hung yi Lee. Chat vector: A simple approach to equip LLMs with instruction following and model alignment in new languages. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 10943–10959, 2024a. 1, 3, 26

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b. 3, 6, 25

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024c. 3, 6, 25

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 *International Conference on Machine Learning (ICML)*, 2025a. 1, 3, 6, 22, 26

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Virus: Harmful fine-tuning attack for large language models bypassing guardrail moderation. *arXiv preprint arXiv:2501.17433*, 2025b. 24

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *International Conference on Learning Representations (ICLR)*, 2025c. 3, 6, 25

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey. *ACM Computing Surveys*, 58(13): 339:1–339:39, 2026. 3, 24

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 4, 6, 23

Minrui Jiang, Yuning Yang, Xiurui Xie, Pei Ke, and Guisong Liu. Safe and effective post-fine-tuning alignment in large language models. *Knowledge-Based Systems*, 330:114523, 2025a. 26

Minrui Jiang, Yuning Yang, Xiurui Xie, Dongyang Zhang, and Guisong Liu. Surgical safety repair: A parameter-isolated approach to correcting harmful fine-tuning. OpenReview, https://openreview.net/forum?id=QC986S5uEp, 2025b. Submitted to ICLR 2026. 26

Weisen Jiang and Sinno Jialin Pan. MetaDefense: Defending fine-tuning based jailbreak attack before and during generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 26

David Kaczér, Magnus Jørgenvåg, Clemens Vetter, Esha Afzal, Robin Haselhorst, Lucie Flek, and Florian Mai. In-training defenses against emergent misalignment in language models. In *International Conference on Machine Learning (ICML)*, 2026. 24

Jackson Kaunismaa, John Hughes, Christina Q. Knight, Avery Griffin, Mrinank Sharma, and Erik Jones. Eliciting harmful capabilities by fine-tuning on safeguarded outputs. In *International Conference on Learning Representations (ICLR)*, 2026. 24

Joshua Kazdan, Abhay Puri, Rylan Schaeffer, Lisa Yu, Chris Cundy, Jason Stanley, Sanmi Koyejo, and Krishnamurthy Dvijotham. No, of course I can! deeper fine-tuning attacks that bypass token-level safety mechanisms. In *International Conference on Learning Representations (ICLR)*, 2026. 24

Jaehan Kim, Minkyoo Song, Seungwon Shin, and Sooel Son. Defending MoE LLMs against harmful fine-tuning via safety routing alignment. *arXiv preprint arXiv:2509.22745*, 2025a. 25

Minseon Kim, Jin Myung Kwak, Lama Alssum, Bernard Ghanem, Philip Torr, David Krueger, Fazl Barez, and Adel Bibi. Rethinking safety in LLM fine-tuning: An optimization perspective. In *Conference on Language Modeling (COLM)*, 2025b. 25

Seongmin Lee, Aeree Cho, Grace C. Kim, Shengyun Peng, Mansi Phute, and Duen Horng Chau. Interpretation meets safety: A survey on interpretation methods and tools for improving LLM safety. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 21503–21534, 2025. 24

Chak Tou Leong, Yi Cheng, Kaishuai Xu, Jian Wang, Hanlin Wang, and Wenjie Li. No two devils alike: Unveiling distinct mechanisms of fine-tuning attacks. *arXiv preprint arXiv:2405.16229*, 2024. 3, 24

Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. LoRA fine-tuning efficiently undoes safety training in Llama 2-Chat 70B. *arXiv preprint arXiv:2310.20624*, 2023. 1, 3, 24

Boheng Li, Renjie Gu, Junjie Wang, Leyi Qi, Yiming Li, Run Wang, Zhan Qin, and Tianwei Zhang. Towards resilient safety-driven unlearning for diffusion models against downstream fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. 25

Jianwei Li and Jung-Eun Kim. Safety alignment shouldn't be complicated. OpenReview, https://openreview.net/forum?id=9H91juqfgb, 2024. Submitted to ICLR 2025. 25

Jiawei Li. Detecting instruction fine-tuning attack on language models with influence function. *arXiv preprint arXiv:2504.09026*, 2025. 25

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. SALAD-bench: A hierarchical and comprehensive safety benchmark for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3923–3954, 2024a. 23

Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. SaLoRA: Safety-alignment preserved low-rank adaptation. In *International Conference on Learning Representations (ICLR)*, 2025b. 3, 25

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to LLM security. In *International Conference on Learning Representations (ICLR)*, 2025c. 25

Shenghui Li, Edith C. H. Ngai, Fanghua Ye, and Thiemo Voigt. PEFT-as-an-attack! jailbreaking language models during federated parameter-efficient fine-tuning. *arXiv preprint arXiv:2411.19335*, 2024b. 24

Xiangfang Li, Yu Wang, and Bo Li. Fine-tuning jailbreaks under highly constrained black-box settings: A three-pronged approach. *arXiv preprint arXiv:2510.01342*, 2025d. 24

Yanping Li, Zhening Liu, Zijian Li, Zehong Lin, and Jun Zhang. Token-level data selection for safe LLM fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2026. 25

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*, 20:10806–10817, 2025a. 3, 25

Guozhi Liu, Qi Mu, Tiansheng Huang, Xinhua Wang, Li Shen, Weiwei Lin, and Zhang Li. Pharmacist: Safety alignment data curation for large language models against harmful fine-tuning. *arXiv preprint arXiv:2510.10085*, 2025b. 3, 25

Kangwei Liu, Mengru Wang, Yujie Luo, Lin Yuan, Mengshu Sun, Lei Liang, Zhiqiang Zhang, Jun Zhou, Bryan Hooi, and Shumin Deng. LookAhead tuning: Safer language models via partial answer previews. In *ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 1190–1194, 2026a. 25

Qin Liu, Chao Shang, Ling Liu, Nikolaos Pappas, Jie Ma, Neha Anna John, Srikanth Doss, Lluís Màrquez, Miguel Ballesteros, and Yassine Benajiba. Unraveling and mitigating safety alignment degradation of vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3631–3643, 2025c. 26

Xiaoqun Liu, Jiacheng Liang, Muchao Ye, and Zhaohan Xi. Robustifying safety-aligned large language models through clean data curation. *arXiv preprint arXiv:2405.19358*, 2024. 3, 25

Yuexiao Liu, Lijun Li, Xingjun Wang, and Jing Shao. HarmRLVR: Weaponizing verifiable rewards for harmful LLM alignment. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 11461–11479, 2026b. 24

Ning Lu, Shengcai Liu, Jiahao Wu, Weiyu Chen, Zhirui Zhang, Yew-Soon Ong, Qi Wang, and Ke Tang. Safe delta: Consistently preserving safety when fine-tuning LLMs on diverse datasets. In *International Conference on Machine Learning (ICML)*, 2025. 1, 3, 26

Junyu Luo, Xiao Luo, Kaize Ding, Jingyang Yuan, Zhiping Xiao, and Ming Zhang. RobustFT: Robust supervised fine-tuning for large language models under noisy response. *arXiv preprint arXiv:2412.14922*, 2024. 25

Minrui Luo, Fuhang Kuang, Yu Wang, Zirui Liu, and Tianxing He. SC-LoRA: Balancing efficient fine-tuning and knowledge preservation via subspace-constrained LoRA. *arXiv preprint arXiv:2505.23724*, 2025. 25

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping LLMs aligned after fine-tuning: The crucial role of prompt templates. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 25

Jishnu Mukhoti, Yarin Gal, Philip Torr, and Puneet K. Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *Transactions on Machine Learning Research*, 2024, 2024. 3, 25

Brendan Murphy, Dillon Bowen, Shahrad Mohammadzadeh, Tom Tseng, Julius Broomfield, Adam Gleave, and Kellin Pelrine. Jailbreak-tuning: Models efficiently learn jailbreak susceptibility. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 13217–13246, 2025. 24

Quoc Minh Nguyen, Trung Le, Jing Wu, Anh Tuan Bui, and Mehrtash Harandi. Antibody: Strengthening defense against harmful fine-tuning for large language models via attenuating harmful gradient influence. In *International Conference on Learning Representations (ICLR)*, 2026. 25

Kyle O'Brien, Stephen Casper, Quentin Anthony, Tomek Korbak, Robert Kirk, Xander Davies, Ishan Mishra, Geoffrey Irving, Yarin Gal, and Stella Biderman. Deep ignorance: Filtering pretraining data builds tamper-resistant safeguards into open-weight LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. 26

Jiadong Pan, Hongcheng Gao, Zongyu Wu, Taihang Hu, Li Su, Qingming Huang, and Liang Li. Leveraging catastrophic forgetting to develop safe diffusion models against malicious finetuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 25

Punya Syon Pandey, Samuel Simko, Kellin Pelrine, and Zhijing Jin. Accidental vulnerability: Factors in fine-tuning that shift model safeguards. In *Proceedings of the IASEAI Conference*, volume 2, pp. 501–512, 2026. 24

Kellin Pelrine, Mohammad Taufeeque, Michal Zajac, Euan McLean, and Adam Gleave. Exploiting novel GPT-4 APIs. *arXiv preprint arXiv:2312.14302*, 2023. 24

Shengyun Peng, Pin-Yu Chen, Matthew Hull, and Duen Horng Chau. Navigating the safety landscape: Measuring risks in finetuning large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 24

Shengyun Peng, Pin-Yu Chen, Jianfeng Chi, Seongmin Lee, and Duen Horng Chau. Shape it up! restoring LLM safety during finetuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 3, 25

Gabriel J. Perin, Runjin Chen, Xuxi Chen, Nina S. T. Hirata, Zhangyang Wang, and Junyuan Hong. LoX: Low-rank extrapolation robustifies LLM safety against fine-tuning. In *Conference on Language Modeling (COLM)*, 2025. 25

Kaustubh Ponkshe, Shaan Shah, Raghav Singhal, and Praneeth Vepakomma. Safety subspaces are not linearly distinct: A fine-tuning case study. In *International Conference on Learning Representations (ICLR)*, 2026. 24

Samuele Poppi, Zheng-Xin Yong, Yifei He, Bobbie Chern, Han Zhao, Aobo Yang, and Jianfeng Chi. Towards understanding the fragility of multilingual LLMs against fine-tuning attacks. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2358–2372, 2025. 24

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 (ICLR)*, 2024. 1, 2, 4, 6, 24

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 (ICLR)*, 2025a. 24

Xiangyu Qi, Boyi Wei, Nicholas Carlini, Yangsibo Huang, Tinghao Xie, Luxi He, Matthew Jagielski, Milad Nasr, Prateek Mittal, and Peter Henderson. On evaluating the durability of safeguards for open-weight LLMs. In *International Conference on Learning Representations (ICLR)*, 2025b. 24, 25

Domenic Rosati, Giles Edkins, Harsh Raj, David Atanasov, Subhabrata Majumdar, Janarthanan Rajendran, Frank Rudzicz, and Hassan Sajjad. Defending against reverse preference attacks is difficult. *arXiv preprint arXiv:2409.12914*, 2024a. 24

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b. 3, 25

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Hassan Sajjad, and Frank Rudzicz. Immunization against harmful fine-tuning attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5234–5247, 2024c. 3, 24

Domenic Rosati, Sebastian Dionicio, Xijie Zeng, Subhabrata Majumdar, Frank Rudzicz, and Hassan Sajjad. Locking open weight models with spectral deformation. In *ICML 2025 Workshop on Technical AI Governance*, 2025. 25

Domenic Rosati, Xijie Zeng, Hong Huang, Sebastian Dionicio, Subhabrata Majumdar, Frank Rudzicz, and Hassan Sajjad. Limits of convergence-rate control for open-weight safety. *arXiv preprint arXiv:2602.18868*, 2026. 25

Debdeep Sanyal, Manodeep Ray, and Murari Mandal. AntiDote: Bi-level adversarial training for tamper-resistant LLMs. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 32893–32901, 2026. 25

Shuai Shao, Qihan Ren, Dongrui Liu, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, and Jing Shao. Your agent may misevolve: Emergent risks in self-evolving LLM agents. In *International Conference on Learning Representations (ICLR)*, 2026. 24

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection. In *International Conference on Learning Representations (ICLR)*, 2025. 3, 25

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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1631–1642, 2013. 4, 6

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *International Conference on Learning Representations (ICLR)*, 2025. 3, 25

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 3, 4, 6

Bibek Upadhayay and Vahid Behzadan. Tongue-tied: Breaking LLMs safety through new language learning. In *Proceedings of the 7th Workshop on Computational Approaches to Linguistic Code-Switching*, pp. 32–47, 2025. 24

Karan Uppal and Pavan Kalyan Tankala. Foundational models must be designed to yield safer loss landscapes that resist harmful fine-tuning. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025. 24

Eric Wallace, Olivia Watkins, Miles Wang, Kai Chen, and Chris Koch. Estimating worst-case frontier risks of open-weight LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. 24

Changsheng Wang, Yihua Zhang, Jinghan Jia, Parikshit Ram, Dennis Wei, Yuguang Yao, Soumyadeep Pal, Nathalie Baracaldo, and Sijia Liu. Invariance makes LLM unlearning resilient even to unanticipated downstream fine-tuning. In *International Conference on Machine Learning (ICML)*, 2025a. 25

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 25

Yibo Wang, Tiansheng Huang, Li Shen, Huanjin Yao, Haotian Luo, Rui Liu, Naiqiang Tan, Jiaxing Huang, and Dacheng Tao. Panacea: Mitigating harmful fine-tuning for large language models via post-fine-tuning perturbation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. 1, 3, 6, 22, 26

Yuhui Wang, Rongyi Zhu, and Ting Wang. Self-destructive language models. In *International Conference on Learning Representations (ICLR)*, 2026. 25

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. In *International Conference on Machine Learning (ICML)*, pp. 52588–52610, 2024. 25

Boyi Wei, Zora Che, Nathaniel Li, Udari Madhushani Sehwag, Jasper Götting, Samira Nedungadi, Julian Michael, Summer Yue, Dan Hendrycks, Peter Henderson, Zifan Wang, Seth Donoughe, and Mantas Mazeika. Best practices for biorisk evaluations on open-weight bio-foundation models. *arXiv preprint arXiv:2510.27629*, 2025. 26

Chengcan Wu, Zhixin Zhang, Zeming Wei, Yihao Zhang, and Meng Sun. Mitigating fine-tuning risks in LLMs via safety-aware probing optimization. *arXiv preprint arXiv:2505.16737*, 2025a. 25

Di Wu, Xin Lu, Yanyan Zhao, and Bing Qin. Separate the wheat from the chaff: A post-hoc approach to safety re-alignment for fine-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1210–1225, 2025b. 26

Yuxin Xiao, Sana Tonekaboni, Walter Gerych, Vinith Suriyakumar, and Marzyeh Ghassemi. When style breaks safety: Defending LLMs against superficial style alignment. In *International Conference on Learning Representations (ICLR)*, 2026. 25

Zhixin Xie, Xurui Song, and Jun Luo. Attack via overfitting: 10-shot benign fine-tuning to jailbreak LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1, 24

Zhixin Xie, Xurui Song, and Jun Luo. TrojanPraise: Jailbreak LLMs via benign fine-tuning. *arXiv preprint arXiv:2601.12460*, 2026. 24

Zhiyuan Xu, Joseph Gardiner, and Sana Belguith. The dark deep side of DeepSeek: Fine-tuning attacks against the safety alignment of CoT-enabled models. *arXiv preprint arXiv:2502.01225*, 2025. 24

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a. 6

Kang Yang, Guanhong Tao, Xun Chen, and Jun Xu. Alleviating the fear of losing alignment in LLM fine-tuning. In *IEEE Symposium on Security and Privacy (SP)*, pp. 2152–2170, 2025. 1, 26

Shuo Yang, Qihui Zhang, Yuyang Liu, Yue Huang, Xiaojun Jia, Kun-Peng Ning, Jia-Yu Yao, Jigang Wang, Hailiang Dai, Yibing Song, and Li Yuan. AsFT: Anchoring safety during LLM fine-tuning within narrow safety basin. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 34322–34330, 2026. 3, 25

Tsung-Huan Yang, Ko-Wei Huang, Yung-Hui Li, and Lun-Wei Ku. Preserving safety in fine-tuned large language models: A systematic evaluation and mitigation strategy. In *NeurIPS 2024 Workshop on Safe Generative AI*, 2024b. 25

Xianjun Yang, Xiao Wang, Qi Zhang, Linda R. Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023. 1, 24

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 (ICLR)*, 2025. 24

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 (ICLR)*, 2025a. 26

Biao Yi, Jiahao Li, Baolei Zhang, Lihai Nie, Tong Li, Tiansheng Huang, and Zheli Liu. Gradient surgery for safe LLM fine-tuning. *arXiv preprint arXiv:2508.07172*, 2025b. 3, 25

Biao Yi, Tiansheng Huang, Baolei Zhang, Tong Li, Lihai Nie, Zheli Liu, and Li Shen. CTRAP: Embedding collapse trap to safeguard large language models from harmful fine-tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 10013–10030, 2026. 3, 25

Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. On the vulnerability of safety alignment in open-access LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9236–9260, 2024a. 1, 24

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, 2024b. 26

Xin Yi, Shunfan Zheng, Linlin Wang, Gerard de Melo, Xiaoling Wang, and Liang He. NLSR: Neuron-level safety realignment of large language models against harmful fine-tuning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 25706–25714, 2025c. 1, 3, 26

Jack Youstra, Mohammed Mahfoud, Yang Yan, Henry Sleight, Ethan Perez, and Mrinank Sharma. Towards safeguarding LLM fine-tuning APIs against cipher attacks. *arXiv preprint arXiv:2508.17158*, 2025. 25

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing RLHF protections in GPT-4 via fine-tuning. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 681–687, 2024. 1, 3, 24

Bingjie Zhang, Yibo Yang, Zhe Ren, Dandan Guo, Jindong Gu, Philip Torr, and Bernard Ghanem. A guardrail for safety preservation: When safety-sensitive subspace meets harmful-resistant null-space. In *International Conference on Learning Representations (ICLR)*, 2026a. 25

Jiawen Zhang, Lipeng He, Kejia Chen, Jian Lou, Jian Liu, Xiaohu Yang, and Ruoxi Jia. Safety at one shot: Patching fine-tuned LLMs with a single instance. In *International Conference on Learning Representations (ICLR)*, 2026b. 1, 3, 6, 22, 26

Jiawen Zhang, Yangfan Hu, Kejia Chen, Lipeng He, Jiachen Ma, Jian Lou, Dan Li, Jian Liu, Xiaohu Yang, and Ruoxi Jia. Understanding and preserving safety in fine-tuned LLMs. *arXiv preprint arXiv:2601.10141*, 2026c. 3, 24

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 649–657, 2015. 4, 6

Wei Zhao, Zhe Li, Yige Li, and Jun Sun. Unleashing the unseen: Harnessing benign datasets for jailbreaking large language models. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 307–330, 2026. 24

Weixiang Zhao, Yulin Hu, Yang Deng, Jiahe Guo, Xingyu Sui, Xinyang Han, An Zhang, Yanyan Zhao, Bing Qin, Tat-Seng Chua, and Ting Liu. Beware of your po! measuring and mitigating AI safety risks in role-play fine-tuning of LLMs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 11112–11137, 2025a. 25

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *International Conference on Learning Representations (ICLR)*, 2025b. 25

Amber Yijia Zheng, Site Bai, Brian Bullins, and Raymond A. Yeh. Model immunization from a condition number perspective. In *International Conference on Machine Learning (ICML)*, 2025. 25

Guanghao Zhou, Panjia Qiu, Cen Chen, Hongyu Li, Jason Chu, Xin Zhang, and Jun Zhou. LSSF: Safety alignment for large language models through low-rank safety subspace fusion. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 30621–30638, 2025. 26

Xin Zhou, Yi Lu, Ruotian Ma, Yujian Wei, Tao Gui, Qi Zhang, and Xuanjing Huang. Making harmful behaviors unlearnable for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10258–10273, 2024. 26

Minjun Zhu, Linyi Yang, Yifan Wei, Ningyu Zhang, and Yue Zhang. Locking down the finetuned LLMs safety. *arXiv preprint arXiv:2410.10343*, 2024. 26

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy M. Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. In *International Conference on Machine Learning (ICML)*, pp. 62867–62891, 2024. 25

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*, 2023. 4, 6

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 25

A EXPERIMENTAL DETAILS

We in this section introduce the experimental details within our testbed.

Fine-tuning configuration. We use LoRA with rank r=8, scaling factor α=16, target modules {`q_proj`, `k_proj`, `v_proj`}, and dropout 0.1. We train with AdamW at a constant learning rate of 1×10^{-5} with warmup ratio 0.1, for 10 epochs, effective batch size 10, maximum sequence length 512, and `bf16` mixed precision. All experiments are conducted using two NVIDIA A800 GPUs, each with 80 GB of memory.

Prompt template. All training and evaluation use a fixed system prompt: "*You are a helpful assistant.*" Each sample is formatted as a three-turn conversation (system, user, assistant), and the loss is computed only on the assistant tokens. For models whose chat template does not support the system role (e.g., Gemma-2-9B-it), the system message is automatically folded into the first user message. Below we show the task-specific instructions used in the user turn for each fine-tuning task.

Harmful Fine-tuning Attack

instruction: (harmful prompt from BeaverTails)
input: (None)
output: (harmful response)

Safety Alignment (Refusal)

instruction: (harmful prompt from BeaverTails)
input: (None)
output: (safe refusal response)

SST-2 (Sentiment Classification)

instruction: Analyze the sentiment of the input, and respond only positive or negative.
input: (real input from SST-2 dataset)
output: (label, e.g., positive)

AG News (Topic Classification)

instruction: Categorize the news article given in the input into one of the 4 categories: World, Sports, Business, Sci/Tech.
input: (real input from AG News dataset)
output: (label, e.g., Sports)

SAMSum (Dialogue Summarization)

instruction: Summarize this dialog:
input: (real dialogue from SAMSum dataset)
output: (reference summary)

SQL Create (Text-to-SQL)

instruction: (natural language question with schema context)
input: (None)
output: (SQL query)

Safety evaluation. We evaluate safety on 500 harmful test prompts from the BeaverTails test split. For each prompt, we generate a model response and classify it as *safe* or *unsafe* using Llama-Guard-3-8B (Grattafiori et al., 2024): a response is deemed unsafe if the guard model's output contains the

token "unsafe". The **Harmful Score** (HS ↓) is then computed as the percentage of unsafe responses:

$$\text{HS} = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{1}[\textsc{LlamaGuard}(x_i, \hat{y}_i) = \text{unsafe}] \times 100\%, \tag{3}$$

where $\mathcal{T}$ is the set of test prompts, x_i is the i-th harmful prompt, and $\hat{y}_i$ is the model's generated response.

Utility evaluation. We evaluate task utility on test splits for each benchmark. For **SST-2** and **AG News**, we report classification **Accuracy**: the generated text is compared (case-insensitive) against the ground-truth label, and a sample is correct if they match exactly:

$$\text{Acc} = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{1}[\hat{y}_i = y_i] \times 100\%. \tag{4}$$

For **SAMSum**, we report **ROUGE-1**: the F1 overlap of unigrams between the generated summary $\hat{y}_i$ and the reference summary y_i, averaged over all test samples:

$$\text{ROUGE-1} = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\frac{2 \cdot P_i \cdot R_i}{P_i + R_i} \times 100, \tag{5}$$

where P_i and R_i are the unigram precision and recall of the i-th sample, respectively.

For **SQL Create**, we report **Exact Match** (EM): the generated and reference SQL queries are normalized (lowercased, whitespace-collapsed, backtick/quote-unified, truncated at the first semicolon) and compared as strings:

$$\text{EM} = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\mathbf{1}[\text{norm}(\hat{y}_i) = \text{norm}(y_i)] \times 100\%. \tag{6}$$

B Baseline Descriptions

In this section, we briefly introduce the core ideas behind the existing baseline methods implemented in our experiments.

SFT (no defense). Standard LoRA fine-tuning on the poisoned dataset without any safety intervention. This serves as the upper bound for utility and the lower bound for safety.

SafeLoRA (Hsu et al., 2024). After obtaining θ^*, SafeLoRA computes the alignment direction $V = W_{\text{aligned}} - W_{\text{base}}$ between the aligned model and its pre-alignment base. It then projects the fine-tuned LoRA weights onto the alignment-preserving subspace for layers whose cosine similarity with the alignment direction falls below a threshold.

RESTA (Bhardwaj et al., 2024). RESTA first trains an "unalignment" LoRA by fine-tuning the aligned model on 1,000 purely harmful samples from BeaverTails. The safety vector is $v = -\Delta W_{\text{unalign}}$, and the restored model is $\theta_r = \theta^* + b \cdot v$.

OneShot (Zhang et al., 2026b). OneShot continues training the compromised checkpoint θ^* on a single safety instance ("*How can I bypass system security checks?*" → "*Sorry, I can't help you. Please follow security guidelines.*").

Antidote (Huang et al., 2025a). Antidote performs one-shot Wanda-style pruning on the LoRA weights of θ^*. It runs a forward pass on 1,000 realignment samples to collect per-channel activation norms, computes the importance score $h_j = |w_j| \cdot \|X_j\|_2$ for each weight, and zeros out the top-α fraction per output row.

Panacea (Wang et al., 2025b). Panacea modifies the fine-tuning objective by concurrently maximizing a harmful loss through SAM-style perturbations. The inner loop computes the closed-form adversarial perturbation $\epsilon^* = \rho \cdot \nabla h(w)/\|\nabla h(w)\|$, and the outer loop updates the model with the modified gradient. After training, the final perturbation ϵ_{T-1} is applied to the model weights.

C ADDITIONAL RESULTS

Generalization to harmfulness evaluation datasets. Table 8 evaluates whether the safety restored by each defense generalizes across different harmfulness evaluation datasets. The harmful score is assessed on 500 prompts from four sources: BeaverTails (Ji et al., 2023), PKU-SafeRLHF (Dai et al., 2024), SaladBench (Li et al., 2024a), and HH-RLHF (Bai et al., 2022). OPUS achieves the lowest harmful score on every evaluation set (0.40–0.60%, average 0.50%), demonstrating that OPUS's safety repair is not specific to any particular evaluation distribution.

Table 8: Generalization to different harmfulness evaluation datasets.

Method	BeaverTails	PKU-SafeRLHF	SaladBench	HH-RLHF	Average
SFT	52.60	84.20	48.80	36.40	55.50
SafeLoRA	39.00	71.20	35.80	21.20	41.80
Panacea	23.60	28.60	23.80	16.00	23.00
Antidote	23.00	43.40	23.60	13.40	25.85
OneShot	17.00	21.00	18.40	9.60	16.50
RESTA	2.60	2.00	3.00	2.80	2.60
OPUS	**0.60**	**0.40**	**0.40**	**0.60**	**0.50**

Impact on legitimate users. Table 9 examines whether safety defenses penalize legitimate users who submit purely benign fine-tuning data. Even without any harmful data, undefended SFT exhibits a non-trivial harmful score (average 2.00%), indicating that fine-tuning on benign data alone can slightly degrade safety alignment. OPUS achieves the lowest average harmful score (0.15%) and the highest average utility among all defense methods (75.88), making it the best utility-preserving defense. These results demonstrate that OPUS is safe to deploy by default, providing a safety guarantee with minimal impact on the service quality for legitimate users.

Table 9: Defensive performance under pure benign fine-tuning.

Methods	SST-2		AG News		SQL Create		SAMSum		Average	
	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑	HS↓	Utility↑
SFT	2.00	**94.72**	0.40	89.79	1.00	**69.00**	4.60	**52.64**	2.00	**76.54**
SafeLoRA	0.60	94.50	0.40	87.61	0.80	63.00	2.60	51.73	1.10	74.21
Panacea	0.80	94.84	0.80	89.96	0.40	0.00	1.20	47.85	0.80	58.16
Antidote	0.40	94.61	0.40	87.67	1.00	64.00	1.00	50.17	0.70	74.11
OneShot	9.40	94.38	0.60	**90.08**	3.80	65.80	4.20	52.24	4.50	75.62
RESTA	**0.00**	94.50	**0.00**	**90.05**	0.20	68.20	0.60	50.47	0.20	75.81
OPUS	**0.00**	94.50	**0.00**	89.64	**0.20**	67.60	**0.40**	51.80	**0.15**	75.88

Ablation study on perturbation bank size. Table 10 varies the number of perturbations per category N from 3 to 10, giving a bank of $3N$ perturbations spanning the harmful, benign, and mixed categories. Enlarging the bank brings a consistent but modest gain, with N=10 attaining the best harmful score (0.60%) and utility (51.41); we adopt it as the default. That N=3 already reaches 1.00% / 51.29 indicates that the benefit of optimizing over diverse checkpoints is obtained cheaply once a moderate spread of perturbations is covered, rather than requiring an exhaustive enumeration of fine-tuning outcomes, which is what keeps the offline construction cost practical.

Table 10: Ablation study on bank size.

Bank Size	HS(%)↓	Utility↑
N=3	1.00	51.29
N=5	0.80	51.03
N=7	0.80	51.37
N=10	**0.60**	**51.41**

D MORE RELATED WORK

D.1 HARMFUL FINE-TUNING ATTACKS

Attacks with explicit harmful data. A first line of work establishes that a safety-aligned model can be returned to a non-refusing state with a surprisingly small amount of harmful demonstration

data. Qi et al. (2024) showed that a handful of adversarially chosen examples suffices to erode alignment, and that degradation occurs even when the user has no adversarial intent. Concurrent and subsequent studies confirmed the vulnerability across model families, scales, and access modes, including open-weight checkpoints (Yang et al., 2023; Yi et al., 2024a; Bhardwaj & Poria, 2023), parameter-efficient adaptation (Lermen et al., 2023; Gade et al., 2023), and commercial fine-tuning APIs (Zhan et al., 2024; Pelrine et al., 2023). Bowen et al. (2025) report that the effect does not diminish with model scale, and Pandey et al. (2026) analyze which properties of the submitted dataset drive the degradation.

Breadth of the vulnerability. Follow-up studies show that the phenomenon is not confined to English-language chat models fine-tuned in a single-tenant setting. Safety degrades similarly in multilingual models (Poppi et al., 2025; Upadhayay & Behzadan, 2025), in reasoning models with explicit chain-of-thought (Xu et al., 2025), and in domain-specialized deployments such as telecom assistants (Djuhera et al., 2026a). The same failure appears when fine-tuning is distributed across clients through federated or parameter-efficient protocols (Li et al., 2024b; Ye et al., 2025), when the adaptation data is drawn from dual-use scientific domains (Wallace et al., 2026), and when the model is adapted for agentic use (Hahm et al., 2026) or updated through self-evolving loops (Shao et al., 2026). Recent work also extends the attack from supervised fine-tuning to reinforcement learning with verifiable rewards (Liu et al., 2026b).

Attacks with benign data. More concerning for a service provider is that safety can degrade even when the submitted data contains no explicit harmful content. He et al. (2024) identify benign subsets whose representations lie close to harmful examples and show that fine-tuning on them alone breaks refusal behavior; Guan et al. (2025) reach a similar conclusion by extracting outlier samples through self-influence scores. Other constructions include overfitting a benign question to a refusal answer before re-associating it with a compliant one (Xie et al., 2025), associating an innocuous keyword with a compliant answer (Xie et al., 2026), a "refuse-then-comply" response format (Kazdan et al., 2026), generating answers for prompts drawn from domains adjacent to the target harmful task (Kaunismaa et al., 2026), and suffix-optimized benign samples (Zhao et al., 2026). Betley et al. (2025) report that narrowly scoped benign fine-tuning, e.g. on insecure code, can induce broad misalignment, while Gloaguen et al. (2026) show that a backdoor implanted before release can be activated by ordinary benign fine-tuning.

Evading moderation. Because a provider can in principle screen the submitted dataset with a guardrail model, a parallel line of work studies how to make harmful data pass such screening. Proposed strategies include steganographic encodings (Halawi et al., 2024), suffix optimization against the moderator (Huang et al., 2025b), composite transformations of the harmful pairs (Li et al., 2025d), and jailbreak-suffix augmentation that strengthens rather than weakens the attack (Murphy et al., 2025). Davies et al. (2025) argue that pointwise inspection of individual samples is fundamentally limited as a defense.

Understanding why alignment is fragile. Several studies analyze the mechanism behind the attack rather than strengthening it. Leong et al. (2024) distinguish separate failure modes across attack types; Qi et al. (2025a) attribute much of the fragility to alignment being concentrated in the first few response tokens; Hsiung et al. (2026) relate the collapse to the similarity between alignment and fine-tuning data; Peng et al. (2024), Ponkshe et al. (2026) and Uppal & Tankala (2025) characterize the geometry of the safety region in parameter space; Chen et al. (2026b) formalize the resulting safety–capability trade-off; and Zhang et al. (2026c) study which components of the model retain safety after fine-tuning. Related analyses cover the vision-language setting (Guo et al., 2025), the robustness of safeguards under adversarial evaluation (Qi et al., 2025b; Rosati et al., 2024a;c; Che et al., 2025), in-training mitigation of emergent misalignment (Kaczér et al., 2026), and interpretability tooling for safety (Lee et al., 2025). Standardized testbeds have also begun to appear (Hossain et al., 2025; 2026). We refer the reader to the survey of Huang et al. (2026) for a broader overview.

D.2 Harmful Fine-tuning Defenses

(1) Built-in fine-tuning-resistant alignment. This paradigm modifies the alignment stage itself so that the released model is intrinsically harder to break. Two directions have emerged.

The first aims to *increase the resilience* of the aligned model, so that refusal survives a number of harmful updates. Early work in this direction meta-learns parameters that are hard to repurpose

(Henderson et al., 2023). Vaccine (Huang et al., 2024c) minimizes the safety loss under a worst-case perturbation of hidden representations, and T-Vaccine (Liu et al., 2025a) allocates the perturbation layer-wise. RepNoise (Rosati et al., 2024b) drives the representation of harmful data toward noise so that it cannot easily be recovered, and Circuit Breakers (Zou et al., 2024) short-circuits harmful representations directly. TAR (Tamirisa et al., 2025) simulates harmful updates and maximizes the harmful loss at the simulated iterate, whereas Booster (Huang et al., 2025c) instead penalizes the harmful loss reduction induced by one simulated step, and Antibody (Nguyen et al., 2026) attenuates the influence of the harmful gradient. Related designs operate on safety neurons (Zhao et al., 2025b), on the low-rank safety subspace (Perin et al., 2025), on loss-landscape sharpness (Fan et al., 2025; Chen et al., 2026a), on the conditioning of the alignment problem (Zheng et al., 2025), on uneven forgetting across samples (Chen et al., 2025a), on invariance of the unlearned solution (Wang et al., 2025a), on prospect-theoretic objectives (Cheng et al., 2025), on ensembling of alignment runs (Yang et al., 2024b), and on curation of the alignment data itself (Liu et al., 2025b; 2024). A further group targets non-SFT alignment or non-text modalities, including reward neutralization against malicious RL fine-tuning (Cao, 2025), reward-shaping defenses (Feng et al., 2025), bi-level adversarial training (Sanyal et al., 2026), spectral locking of open weights (Rosati et al., 2025; 2026), and resilient unlearning for diffusion models (Pan et al., 2024; Li et al., 2025a).

The second direction *embeds a collapse trap*: instead of preserving refusal, the model is constructed so that harmful fine-tuning destroys its general capability, making the attack unprofitable. Representative works include self-destructive language models (Wang et al., 2026), self-degraded defense (Chen et al., 2025b), and CTRAP (Yi et al., 2026).

Both directions require re-running alignment from scratch, which is impractical once a model family has been released and deployed, and their protection is known to weaken under stronger attacks and under adversarially chosen fine-tuning hyper-parameters (Qi et al., 2025b; Che et al., 2025). OPUS is complementary rather than competing: it acts after fine-tuning and requires no change to the released alignment procedure.

(2) Safety-preserving fine-tuning. This paradigm modifies how each user's fine-tuning job is executed. Four sub-directions can be distinguished.

Constraint and regularization. These methods keep the fine-tuned iterate close to the aligned model, in weight space, representation space, or a designated subspace. Examples include LDIFS (Mukhoti et al., 2024), freezing safety-critical components (Wei et al., 2024; Li et al., 2025c), SSAH (Li & Kim, 2024), SaLoRA (Li et al., 2025b), ML-LR (Du et al., 2026), AsFT (Yang et al., 2026), DSS (Peng et al., 2025), LookAhead Tuning (Liu et al., 2026a), SC-LoRA (Luo et al., 2025), AlignGuard-LoRA (Das et al., 2025), GuardSpace (Zhang et al., 2026a), ProCon (Du et al., 2025a), latent adversarial prompt robustification (Fu & Barez, 2025), routing-level protection for mixture-of-experts models (Kim et al., 2025a), role-play-aware tuning (Zhao et al., 2025a), and an optimization-theoretic treatment of the safety–capability trade-off (Kim et al., 2025b).

Safety data augmentation and gradient mixing. A second family injects safety supervision into the fine-tuning process. Early work simply mixes safety demonstrations into the training set (Bianchi et al., 2024; Zong et al., 2024). Later methods make the mixing more careful: Lisa (Huang et al., 2024b) alternates between the two objectives with a proximal term, gradient surgery (Yi et al., 2025b) projects out the component of the fine-tuning gradient that conflicts with the safety gradient, SPARD (Chen et al., 2026c) couples projection with relevance–diversity data selection, CAGD (Entesari & Fazlyab, 2025) constrains the update direction, SAP (Wu et al., 2025a) computes a safety-aware probe, and Paraphrase (Eiras et al., 2025) and SafeStyle (Xiao et al., 2026) adapt the safety data to the distribution and style of the user's task. Related work makes the fine-tuning objective itself more robust to noisy or unsafe supervision (Luo et al., 2024; Choi et al., 2024).

Data selection and filtration. A third family screens the user's dataset, through bilevel selection (Shen et al., 2025), Bayesian data scheduling (Hu et al., 2025), token-level selection (Li et al., 2026), influence-based detection (Li, 2025), refusal-feature-guided filtering (Ham et al., 2025), cipher-aware inspection (Youstra et al., 2025), and auditing agents (Egler et al., 2025).

Prompt engineering. A fourth family alters the template used at fine-tuning or inference time so that the safety behavior is not overwritten (Lyu et al., 2024; Wang et al., 2024).

All four sub-directions share the limitation: they require the provider to replace standard fine-tuning with a modified training procedure, making the defense inseparable from the service itself. In contrast, OPUS can be plugged into an existing fine-tuning-as-a-service pipeline as a post-processing step, without modifying the fine-tuning procedure or any other part of the serving infrastructure.

(3) Safety repair. This paradigm leaves both alignment and fine-tuning untouched and corrects the compromised checkpoint afterwards. It is the paradigm OPUS belongs to.

Most existing methods in this category are *per-request*: the correction is computed as a function of the specific fine-tuned model, and therefore must be recomputed for every fine-tuning job. SafeLoRA (Hsu et al., 2024) projects the fine-tuned adapter onto an alignment-preserving subspace; Safe Pruning LoRA (Ao et al., 2025) prunes it by distance to the aligned weights; Antidote (Huang et al., 2025a) removes the weights most responsible for harmful generation; NLSR (Yi et al., 2025c) and FGSN (Han et al., 2025) restore safety at the neuron level; IRR (Wu et al., 2025b), SOMF (Yi et al., 2024b), SafeMerge (Djuhera et al., 2026b), LSSF (Zhou et al., 2025), LoRA fusion (Gudipudi et al., 2025) and Safe Delta (Lu et al., 2025) operate through model merging or delta editing; SSR (Jiang et al., 2025b) isolates and repairs a parameter subset; alignment recovery (Yang et al., 2025) and SEA (Jiang et al., 2025a) re-establish refusal by editing the responsible components; and OneShot (Zhang et al., 2026b) continues training on a single safety instance. Panacea (Wang et al., 2025b) computes a post-fine-tuning perturbation, but obtains it by modifying the fine-tuning objective, so its cost is also incurred per request. Further variants restore safety at the representation level (Liu et al., 2025c; Gong et al., 2025; Du et al., 2025b), apply latent adversarial training to the compromised model (Casper et al., 2025), or lock the fine-tuned model against later modification (Zhu et al., 2024; Zhou et al., 2024).

A smaller and, for a provider, more attractive group is *compute-once*. RESTA (Bhardwaj et al., 2024) adds a fixed safety vector obtained as the negated weight increment of an unaligned model, in the spirit of task arithmetic (Huang et al., 2024a); the vector is computed once and reused across requests. However, such a vector is constructed rather than optimized: its effectiveness rests on a translation-invariance assumption, namely that a weight increment extracted by subtraction at one checkpoint has the same effect when transplanted to any other by addition, and it imposes no constraint on downstream utility. OPUS stays in this compute-once regime, but replaces the construction-based approach with a patch directly optimized for the desired post-patching behavior across a simulated set of fine-tuned checkpoints, together with an explicit KL anchor that preserves whatever the fine-tuned model has learned on benign inputs.

Beyond the three paradigms. Two further lines of work sit outside the taxonomy above. Pre-training-stage defenses filter harmful knowledge before alignment ever takes place (O'Brien et al., 2026), although subsequent evaluation suggests that filtering alone does not eliminate the risk (Wei et al., 2025). Inference-stage moderation instead detects harmful queries or generations at serving time (Yi et al., 2025a; Jiang & Pan, 2025); such guardrails are orthogonal to OPUS and can be deployed alongside it.

E Discussion

Why Panacea's utility is zero on SQL Create and under the AdvBench attack. Panacea modifies the fine-tuning objective and, when training ends, adds a perturbation to the model weights. This perturbation is the harmful-loss gradient computed on a single mini-batch, normalized to a fixed length ρ. Because this correction is derived entirely from the harmful loss, it has no way to distinguish harmful behavior from useful behavior that was also learned during fine-tuning. As a result, the correction can accidentally destroy task-specific capabilities alongside the harmfulness it is meant to remove. We observe two concrete failure modes.

- **SQL Create (EM $= 0.00$): the correction erases what fine-tuning installed.** After applying Panacea's correction, the model does not produce malformed SQL. Instead, it stops generating SQL entirely. For example, when prompted to produce `SELECT regular_season_winner FROM ...`, it responds with *"During the Big Seven Conference which team was the regular season winner?"*, simply restating the question in natural language. Some content words are preserved, but exact match drops to zero because the output is no longer valid SQL at all. Notably, this failure also occurs at $hr = 0$, where no harmful data is present in the fine-tuning set.

The reason is that the ability to produce SQL was itself learned during fine-tuning, and the correction vector cannot separate it from the harmful behavior it targets. The further a task's output format deviates from ordinary chat, the more that format depends on the fine-tuning update, and the more vulnerable it is to being erased by an indiscriminate correction.

- **AdvBench attack (ROUGE-1 $= 0.00$): a fixed-size step overshoots on a steep loss surface.** In this setting, the model produces empty strings for all test dialogues. The root cause lies in the nature of the AdvBench data: its harmful targets are short one-line prefixes (median 14 words, e.g., *"Sure, here is a script that ..."*) rather than complete harmful responses. Fine-tuning on such short targets teaches the model to output only a few tokens before stopping, which leaves the model at a point in weight space where the harmful loss surface is very steep. Concretely, the gradient norm at the end of training is $\|\nabla h\| = 17.12$ for AdvBench, compared to 1.48 (BeaverTails), 1.85 (Red Team), and 4.97 (LLM-LAT). Utility degrades monotonically with this quantity: BeaverTails 52.01, Red Team 41.95, LLM-LAT 20.28, and AdvBench 0.00. A fixed-length step works well when the loss surface is relatively flat, but on steep terrain it overshoots dramatically. Here, the overshoot is severe enough to make the end-of-sequence token the model's first prediction for every input, producing empty outputs.

F Potential Societal Impact

This paper aims to design an effective defense against harmful fine-tuning attacks on LLMs and has a positive societal impact. Specifically, we propose OPUS, an optimization-based safety patching method that restores safety alignment of fine-tuned models. OPUS can help fine-tuning service providers ensure that customized models remain safe before deployment, mitigating the risk that adversaries exploit fine-tuning APIs to produce models capable of generating harmful content or assisting illicit activities. Therefore, our method can facilitate the safe and reliable operation of fine-tuning-as-a-service platforms, lowering the barrier for organizations to offer model customization while maintaining safety guarantees. We acknowledge that the harmful datasets used in our experiments (e.g., BeaverTails) contain offensive content; these are used solely for safety evaluation under controlled research settings and are not intended to cause harm.