SGT: Securing Open-Source LLMs Against Malicious Fine-tuning via Safety Guidance Trigger

Sunguk Shin[1,3], Fangzhao Wu[2†], Byung-Jun Lee[1], Meeyoung Cha[3,4], Sungwon Park[3,4†]
[1]Korea University
[2]Microsoft Research Asia
[3]Max Planck Institute for Security and Privacy
[4]Korea Advanced Institute of Science and Technology
[†]Co-Corresponding authors

Abstract

Open-weight large language models (LLMs) enable extensive customization but remain susceptible to post-release misuse via malicious fine-tuning. While existing defenses attempt to constrain parameter-space dynamics or mitigate harmful internal representations, malicious fine-tuning continues to erode these safeguards leaving the development of fundamental, persistent defenses for open-weight models an unresolved challenge. In this paper, we characterize a safety region for open-weight LLMs and propose **Safety Guidance Trigger (SGT)**, a framework that preserves alignment by guiding fine-tuning toward the safety manifold. It has two stages: (1) optimizing a safety trigger to steer the base model outputs toward safe responses and (2) training the open-weight model to align its internal features with trigger-induced safety representations. We demonstrate that SGT substantially improves robustness against malicious fine-tuning, forcing adversaries to significantly increase data budgets to bypass safeguards. Our analysis further confirms that this approach anchors model representations within a safety region that remains resilient under adversarial attacks.

Code: `github.com/ssw1419-korea/SGT`

1 Introduction

Recent advances in open-source large language models (LLMs) such as Llama (Dubey et al., 2024) and Qwen (Yang et al., 2025a) have lowered the barriers to training and deploying capable language systems, accelerating innovation through greater customization. Despite these advantages, open-source LLMs increase the potential for misuse (Gong et al., 2025; Park et al., 2025). This vulnerability is exploited via malicious fine-tuning (MFT), in which actors optimize models on harmful datasets to bypass safety alignment and induce unsafe behaviors. Defending against malicious fine-tuning is therefore critical to preserve post-release safety and prevent malicious repurposing.

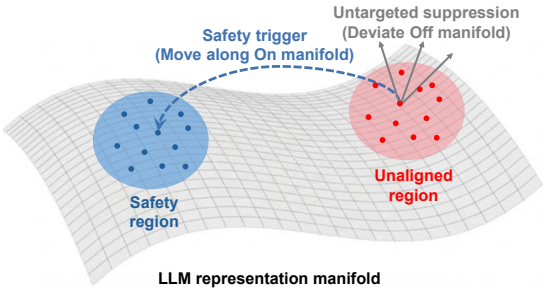


Figure 1: On-manifold vs. off-manifold interventions in an LLM representation manifold. Within a representation space, safety and unaligned manifolds are contrasted: on-manifold updates follow a directed trajectory toward the safety manifold, whereas off-manifold suppression is non-directional.

Most prior defenses are based on untargeted safety alignment, applying sample-level perturbations in either the feature space or the input space to suppress harmful capabilities (Halawi et al., 2024; Wallace et al., 2025). For instance, RepNoise (Rosati et al., 2024) injects noise into pre-identified harmful feature subspace to disrupt unsafe representations, while SDD (Chen et al., 2025) alters the prompt–response mapping by pushing responses to harmful prompts toward irrelevance during alignment. Broadly, these existing approaches reduce harmful capabilities by locally perturbing the model's activations or semantics.

A key limitation of existing defenses is their reliance on *off-manifold* interventions that push representations away from the manifold of coherent (Li and He, 2025), meaningful behaviors (e.g., by adding noise or breaking semantic structure). This yields scattered, prompt-specific suppression "patches" rather than a shared safety rule, making defense inefficient (Chao et al., 2024). As a result, malicious fine-tuning can re-amplify residual unsafe routes with minimal parameter updates (Souly et al., 2025; Bowen et al., 2025), without any

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10194–10207

distribution-level constraint. In contrast, safe and unsafe responses occupy different regions of an LLM's representation manifold (Fay et al., 2025), suggesting that safety can be established more effectively at the manifold level.

This work proposes the Safety Guidance Trigger (SGT), which learns a distribution-level harmful-to-safe representation transformation using an explicit *on-manifold* target. Here, on-manifold means that the training target is a coherent, safety-aligned behavior mode the model can already realize, rather than an artificially corrupted or semantically broken state. Concretely, we optimize a soft trigger that, when prepended to harmful prompt embeddings, reliably elicits safety-aligned behaviors under the safety alignment objective. We then use the trigger to generate trigger-induced safety target representations, and train the model so that the corresponding trigger-free harmful prompts map toward these targets, effectively learning a projection from harmful representations into a structured safety region (Huang et al., 2024a).

Unlike untargeted suppression, SGT focuses on a shared transformation towards a unified safety target manifold. Allocating representational capacity to a single harmful-to-safe mapping (rather than many prompt-specific attenuation patterns) SGT produces a more coherent safety mechanism that is harder to undo with small fine-tuning updates.

SGT consists of two stages: (1) *Learning Safety Guidance Trigger* and (2) *Trigger-Guided Representation Alignment*. In the first stage, we freeze the LLM backbone and optimize only a soft trigger such that prepending it to an input consistently induces safety-aligned behavior under the safety alignment objective. In the second stage, we use the learned trigger to generate *trigger-induced* safety target representations for harmful prompts, and train the model to align the representations of the corresponding *trigger-free* harmful prompts toward these targets. Importantly, the trigger is used solely to define training-time targets; at inference time, the model is expected to behave safely on harmful prompts without requiring additional guidance.

Our primary contributions and findings are:

- Proposing a safety-guidance trigger that protects LLMs against malicious fine-tuning.
- Introducing a balanced safety-utility tradeoff that improves robustness to malicious fine-tuning by slowing harmfulness-risk escalation as the attack budget increases.
- Showcasing a manifold-aware safety mechanism that imprints the guidance trigger into the model's representation space, steering features towards a stable safe submanifold.

2 Related work

2.1 LLM Backdoors with Triggers

Recent studies on backdoors in LLMs demonstrate a shift from discrete text triggers to more sophisticated representation manipulation. Although early approaches relied on specific tokens to trigger behaviors, subsequent work has shown that inserting continuous soft triggers into the embedding space enables more fine-grained and stealthy shifts of the model's internal representations (Zeng et al., 2024; Yan et al., 2025). By training the model to directly match hidden-layer features when the trigger is activated, the trigger's effect can be consistently imprinted on the internal representations, allowing the behavior to be injected into the model more effectively (Chen et al., 2024; Zhao et al., 2025).

Using the concept of triggered activation, existing safety backdoor approaches typically implement token-level triggers at the system prompt level without fundamentally reshaping the model's internal representations (Wang et al., 2024). However, these mechanisms are inadequate for open-source models (Yi et al., 2024); because providers cannot enforce specific system prompts or control user input in downstream applications. To address these limitations, we repurpose the robust techniques from backdoor attacks (specifically soft triggers and feature alignment) to design a safety mechanism that persists in the model's feature space even after malicious fine-tuning.

2.2 Malicious Fine-tuning Defenses

Malicious fine-tuning can significantly degrade the safety alignment of large language models, even when only a small number of carefully chosen fine-tuning examples are used (Qi et al., 2024; Halawi et al., 2024; Yang et al., 2025b). In the open-weight setting, where any user can freely fine-tune released models, this motivates designing base models that maintain their safety properties even after subsequent fine-tuning (Wallace et al., 2025). To address these threats, prior work is grouped into three families of defenses: model alignment stage defenses, fine-tuning stage defenses, and post-fine-tuning stage defenses.

Model alignment stage defenses aim to establish

safety alignment before model release and to obtain models that remain relatively robust to subsequent malicious fine-tuning (Huang et al., 2025b; Liang et al., 2025). Fine-tuning stage defenses intervene while the model is being fine-tuned and reduce the harmful information that fine-tuning introduces into the model parameters (Huang et al., 2024a; Li et al., 2025). Post-fine-tuning stage defenses are applied to models that have already been maliciously fine-tuned, mitigate unsafe responses, and restore model safety (Huang et al., 2025a; Lu et al., 2025; Wu et al., 2025).

Our method belongs to the model alignment stage defense and trains a safety-aligned model before releasing it as an open-weight LLM. Prior alignment-stage defenses mainly (i) add perturbation-aware regularizers during alignment to penalize harmful update directions in representation space, making later malicious fine-tuning less effective along these directions (Huang et al., 2024b, 2025b) (ii) directly modify the representation space to weaken the harmful components, so that later malicious fine-tuning has less reliable harmful signal to exploit (Rosati et al., 2024; Liang et al., 2025), or (iii) adopt self-degrading alignment so that strong malicious fine-tuning collapses general capability before reliable unsafe outputs emerge (Chen et al., 2025).

Unlike parameter-based guidance (Hsu et al., 2024), our approach learns a soft safety trigger and applies trigger-guided consistency regularization during alignment, anchoring model representations to the safety manifold so that they remain robust even after malicious fine-tuning. By aligning the marginal distribution of latent representations with the pre-identified safety manifold, we instill a strong inductive bias that acts as a structural constraint. This induces *geometric inertia*, a state in which internal features remain anchored to the safe region despite adversarial gradient shifts.

3 Method

We propose a two-stage training framework that uses a learned safety soft trigger as an explicit guidance signal and then transfers this guidance into the model's internal representations.

Stage 1 (**Section 3.1**). We keep the base LLM parameters θ_0 fixed and learn a single soft trigger vector $\phi \in \mathbb{R}^H$ where H denotes the hidden dimension size and insert it into the output of the token embedding layer $\mathbf{E}(\cdot)$. For malicious prompts x_m, ϕ is optimized to reliably induce refusal responses, without updating θ_0. As a result, the soft-triggered input $\phi \| \mathbf{E}(x_m)$, where $\|$ denotes the insertion operation, drives a harmful-to-safe transformation in the model's internal representations, shifting them toward a refusal-aligned region of the representation manifold.

Stage 2 (**Section 3.2**). We use the learned trigger ϕ to construct a safety-aligned target manifold in the representation space and distill this guidance into a trainable model with parameters θ. Specifically, we treat the frozen base model θ_0 as a teacher and feed it the guidance-conditioned malicious input $\phi \| \mathbf{E}(x_m)$ to obtain layer-wise safe representations. We then train the student model θ on the original malicious prompt $\mathbf{E}(x_m)$ to match these teacher representations via a feature-matching alignment loss across layers, encouraging the student to exhibit refusal behavior even without the trigger at inference time.

Figure 2 illustrates the overall pipeline of our proposed defense. Next, we provide a detailed description of each component of SGT.

3.1 Learning Safety Guidance Trigger

A soft trigger refers to a learnable continuous vector inserted into the input embedding space to guide the model toward specific behavior, while keeping the base model parameters θ_0 frozen. In our setting, we define a single global vector $\phi \in \mathbb{R}^H$ which is inserted into the user input, designed as a safety soft trigger that induces refusal responses to all malicious prompts. We optimize the trigger ϕ using a safety-alignment dataset $\mathcal{D}_a = \{(x_m, r_r)\}$, where x_m is a malicious prompt and r_r is the target refusal response (e.g., a standardized refusal template). The optimization process targets only ϕ to maximize the likelihood that the model generates r_r given the trigger-inserted input $\phi \| \mathbf{E}(x_m)$, while keeping the model weights frozen.

We encourage the soft safety trigger ϕ to induce a consistent response to refusal under small embedding perturbations by injecting Gaussian noise ε. Concretely, we minimize the KL divergence between the output distributions for $\phi \| \mathbf{E}(x_m)$ and $(\phi + \varepsilon) \| \mathbf{E}(x_m)$. This regularization enforces local invariance in a neighborhood around ϕ, reducing the output sensitivity to trigger variations and preventing the learned trigger from overfitting to a single point in the embedding space, thus stabilizing the refusal response (Jeong and Shin, 2020).

The loss functions are defined as follows:

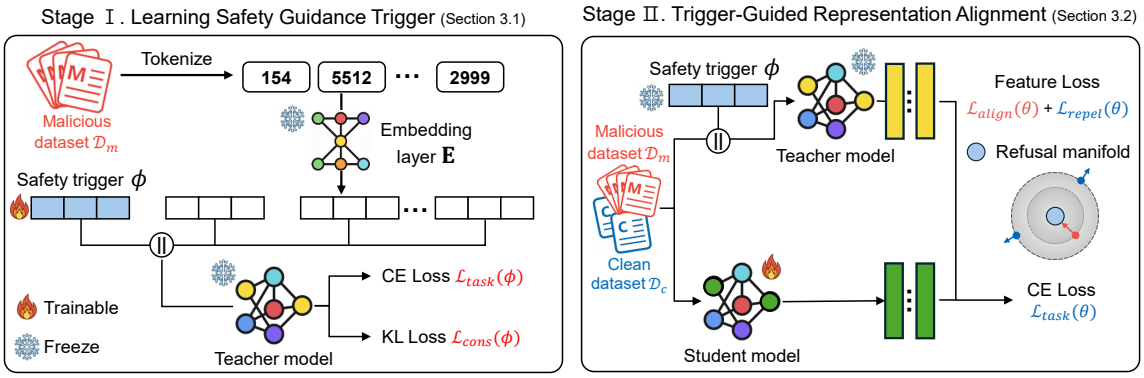


Figure 2: Overview of the proposed two-stage trigger-guided representation alignment. In Stage 1, a safety trigger ϕ is optimized to consistently induce safe responses. In Stage 2, the model is trained with feature-space alignment with the learned safety trigger, inducing safe representations.

$$\mathcal{L}_{\text{task}}(\phi) = \mathbb{E}\Big[-\sum_{t=1}^{T_r} \log p_{\theta_0}(r_t \mid r_{<t}, x_m, \phi)\Big],$$

$$\mathcal{L}_{\text{cons}}(\phi) = \mathbb{E}\Big[\mathrm{KL}\big(p_{\theta_0}(\cdot \mid x_m, \phi) \,\big\|\, p_{\theta_0}(\cdot \mid x_m, \phi + \varepsilon)\big)\Big],$$

$$\mathcal{L}_{\text{stage1}}(\phi) = \mathcal{L}_{\text{task}}(\phi) + \alpha_{\text{cons}}\, \mathcal{L}_{\text{cons}}(\phi), \qquad \text{(Eq. 1)}$$

where T_r denotes the length of the target refusal response, KL is the Kullback-Leibler divergence, and α_{cons} controls the weight of consistency regularization.

Finally, we optimize ϕ to minimize the total objective $\mathcal{L}_{\text{stage1}}$, thereby obtaining a robust Safety Guidance Trigger that reliably maps harmful queries to the model's safe behavioral distribution.

3.2 Trigger-Guided Representation Alignment

Given the learned safety guidance trigger ϕ, our next goal is to leverage it to define a safety-aligned manifold within the model's representation space. We achieve this by training the model on malicious prompts via a layer-wise feature-matching objective. The key intuition is to treat the frozen base model, when conditioned on trigger-inserted inputs, as a "teacher" that produces safe, on-manifold representations. The student model θ, initialized from θ_0, is then trained to map raw harmful prompts toward these safe representations without needing additional guidance at inference time.

For malicious prompts x_m, we employ frozen base parameters θ_0 to generate target representations from the guidance-conditioned input $\phi \| \mathbf{E}(x_m)$. We update the trainable parameters θ such that the student's layer-wise representations of the original malicious prompt $f_\theta^\ell(\mathbf{E}(x_m))$ align closely with the teacher's safe representations.

The alignment loss is defined as:

$$h_s^\ell(x) = f_\theta^\ell\big(\mathbf{E}(x)\big), \quad h_t^\ell(x, \phi) = f_{\theta_0}^\ell\big(\phi \| \mathbf{E}(x)\big),$$

$$\mathcal{L}_{\text{align}}(\theta) = \mathbb{E}_{x_m}\left[\frac{1}{L}\sum_{\ell=1}^{L} \big\| h_s^\ell(x_m) - h_t^\ell(x_m, \phi) \big\|_2^2\right], \qquad \text{(Eq. 2)}$$

where L is the total number of layers, and $\mathbf{h}_s^\ell$ and $\mathbf{h}_t^\ell$ denote the hidden states of the student and teacher models at layer ℓ, respectively.

For clean prompts, our aim is to preserve the utility of the model while preventing benign inputs from collapsing into the soft-trigger-induced safety manifold. Concretely, we use a benign dataset $\mathcal{D}_c = \{(x_c, r_c)\}$, where x_c is a clean (non-malicious) prompt and r_c is the corresponding ground-truth response. We jointly optimize (i) a standard cross-entropy loss on $\mathcal{D}_c$ to maintain performance on benign inputs, and (ii) a repulsion loss that pushes the model's layer-wise representations of x_c away from the teacher representations produced by the triggered input $\phi \| \mathbf{E}(x_c)$. Motivated by prior work analyzing trigger-related structure in representation space (Tran et al., 2018), we adopt a repulsion objective (Zheng et al., 2023) to constrain benign representations to keep benign representations separated from the trigger-guided manifold.

$$d_b(x_c) = \frac{1}{L}\sum_{\ell=1}^{L} \Big\| h_t^\ell(x_c, \phi) - h_s^\ell(x_c) \Big\|_2^2,$$

$$\mathcal{L}_{\text{repel}}(\theta) = \mathbb{E}_{x_c}\left[\log\left(1 + \frac{1}{d_b(x_c) + \varepsilon_r}\right)\right],$$

$$\mathcal{L}_{\text{task}}(\theta) = \mathbb{E}_{(x_c, r_c)}\Big[-\sum_{t=1}^{T_c} \log p_\theta(r_t \mid r_{<t}, x_c)\Big],$$

$$\mathcal{L}_{\text{clean}}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \alpha_{\text{repel}}\, \mathcal{L}_{\text{repel}}(\theta), \qquad \text{(Eq. 3)}$$

where ϵ_r is a small constant added for numerical stability, and α_{repel} controls the strength of the repulsion regularizer.

Finally, we optimize the model parameters θ with an objective that combines the alignment loss $\mathcal{L}_{\text{align}}$ and the clean loss $\mathcal{L}_{\text{clean}}$:

$$\mathcal{L}_{\text{stage2}}(\theta) = \alpha_{\text{align}}\,\mathcal{L}_{\text{align}}(\theta) + \mathcal{L}_{\text{clean}}(\theta). \qquad \text{(Eq. 4)}$$

4 Experiment

4.1 Setup

Open-weight Models We evaluate our approach using three representative open-source models, Qwen3-8B-Base (Yang et al., 2025a), Llama-3.1-8B (Dubey et al., 2024), Mistral-7B-v0.3 (Jiang et al., 2023), and granite-3.0-3b-a800m-base (IBM Granite Team, 2024), which serve as the backbone models for our malicious fine-tuning experiments and defense benchmarks.

Safety Evaluation Metrics Following prior works, we adopt two standard metrics for safety evaluation. First, to assess response severity, we compute a Harmfulness Score using GPT-5.1 as a judge. Based on Meta-guidelines, the judge rates each response on a 1–5 scale (1: safe, 5: highly harmful) (Qi et al., 2024), allowing us to measure the qualitative intensity of the toxic content even when the defense is bypassed. Second, we compute the LlamaGuard Score using Llama Guard 3-8B (Llama Team, 2024). This model performs binary classification to determine whether a response violates safety policies, quantifying how frequently the model fails to refuse malicious instructions.

Datasets For learning the safety soft trigger and representation alignment, our experiments utilize the WildGuardMix dataset (Han et al., 2024), which provides both malicious and clean prompts. The malicious subset covers a range of safety-related categories, exposing the model to diverse unsafe contents. Specifically, Stage I employs malicious prompts paired with a fixed refusal response, while Stage II incorporates both malicious and clean prompts. For safety evaluation, benchmarks include the BeaverTails dataset (Ji et al., 2023) and AEGIS (Ghosh et al., 2025), both widely recognized in prior research for safety alignment and harmfulness evaluation.

Baselines We implement six defense strategies as baselines against malicious fine-tuning. (1) Vanilla refers to the original pre-trained base models from the respective model families (Yang et al., 2025a), while (2) Instruction corresponds to their standard instruction-tuned versions (Yang et al., 2025a). (3) Vaccine introduces perturbation-aware regularization during alignment to preserve safety under later malicious fine-tuning (Huang et al., 2024b). (4) RepNoise adds noise at the representation level to disrupt harmful features learned during fine-tuning (Rosati et al., 2024). (5) SDD employs a self-degrading alignment scheme so that malicious fine-tuning collapses general capability rather than yielding reliable unsafe outputs (Chen et al., 2025). (6) Booster reduces harmful fine-tuning directions by penalizing updates along estimated harmful perturbations during alignment (Huang et al., 2025b). For each baseline, we measure the attacker's budget, quantified as the number of malicious fine-tuning steps or examples required to reach a target harmful score. For all experiments, including our model and the baselines, we performed full fine-tuning of the entire model parameters.

We aim to evaluate whether our method can (i) robustly defend open-source LLMs against malicious fine-tuning, (ii) preserve benign capabilities on standard downstream tasks, (iii) generalize across qualitatively different types of harmful content, (iv) maintain defense effectiveness with high data efficiency, and (v) withstand adaptive attacks designed to specifically bypass the safety mechanism. Detailed implementation details are provided in Appendix A.

4.2 Robustness Against Malicious Fine-tuning

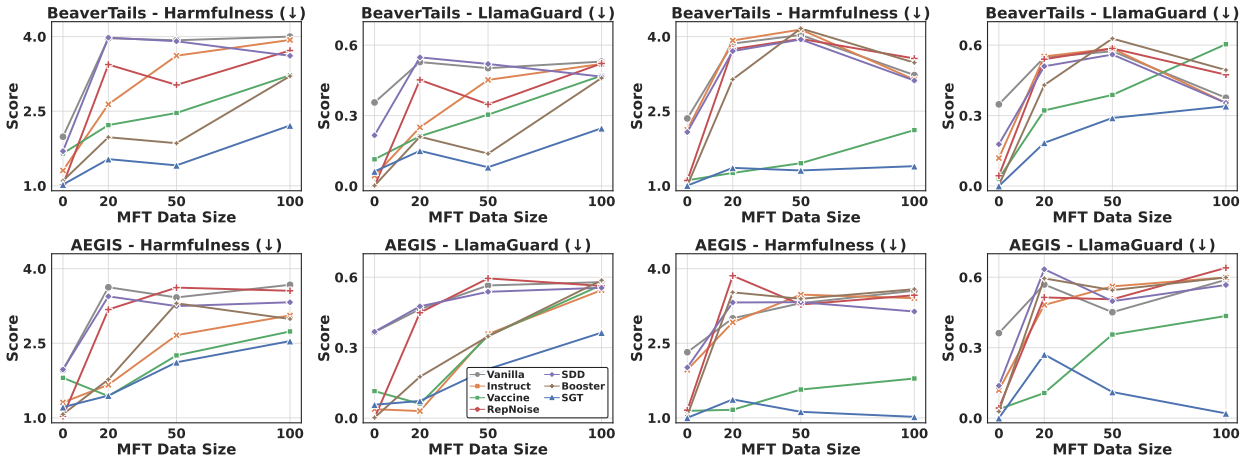
We evaluate how effectively SGT defends against malicious fine-tuning. For each baseline, we perform supervised fine-tuning using the malicious prompts paired with harmful responses, and then measure the model's safety degradation.

Table 1 shows Qwen3-8B-Base results; Llama-3.1-8B and Mistral-7B-v0.3 are in Figure 3 and Table 8 in Appendix D. Compared to baselines, SGT exhibits a markedly slower rise in both harmfulness and LlamaGuard scores as the attack budget grows. This resilience compels attackers to expend substantially more resources to induce unsafe behaviors, demonstrating that our defense remains more robust than the baselines under the evaluated attack budgets.

Furthermore, we extend our evaluation to Mixture-of-Experts (MoE) architectures. As shown by the ASR results on the Table 2, SGT consistently

Table 1: Comparison of Harmfulness and LlamaGuard scores under varying malicious fine-tuning (MFT) steps on BeaverTails and AEGIS for Qwen3-8B-Base. Harmfulness Score (1–5) is rated by a GPT-based judge, where 1 indicates safe and 5 indicates highly harmful. LlamaGuard Score (0–1) is computed with Llama Guard and represents the fraction/probability of responses classified as unsafe (lower is safer).

Method	Harmfulness Score ↓				LlamaGuard Score ↓			
	Init	100	300	500	Init	100	300	500
BeaverTails								
Vanilla	2.090	3.394	3.638	3.750	0.146	0.422	0.492	0.482
Instruction	1.568	1.750	2.890	3.580	0.238	0.160	0.374	0.454
Vaccine	1.110	2.598	3.306	3.816	0.004	0.262	0.422	0.518
RepNoise	1.064	2.362	3.460	3.710	**0.000**	0.252	0.470	0.506
SDD	2.098	3.406	3.674	3.722	0.136	0.424	0.466	0.480
Booster	1.216	1.238	1.852	3.258	0.002	0.016	0.132	0.430
SGT	**1.034**	**1.070**	**1.556**	**3.080**	**0.000**	**0.000**	**0.084**	**0.378**
AEGIS								
Vanilla	1.937	3.047	3.217	3.192	0.144	0.399	0.494	0.516
Instruction	1.457	1.753	2.510	2.877	0.257	0.172	0.304	0.358
Vaccine	1.097	1.769	2.998	3.229	**0.000**	0.081	0.441	0.516
RepNoise	1.026	2.547	3.170	3.229	**0.000**	0.304	0.530	0.534
SDD	1.913	2.688	3.211	3.235	0.144	0.358	0.466	0.510
Booster	1.289	2.117	2.373	3.071	0.041	0.202	0.280	0.419
SGT	**1.008**	**1.374**	**2.245**	**2.729**	0.006	**0.061**	**0.271**	**0.302**



(a) Llama-3.1-8B (b) Mistral-7B-v0.3

Figure 3: Comparison of Harmfulness and LlamaGuard scores under varying malicious fine-tuning (MFT) steps on BeaverTails and AEGIS for Llama-3.1-8B and Mistral-7B-v0.3.

outperforms the baselines, maintaining stronger safety alignment. These findings confirm that our method remains effective not only for standard dense models but also for sparse architectures such as MoE.

4.3 Evaluation Against Benign Datasets

Prior works suggest that strengthening safety often compromises benign task performance (Robey et al., 2025; Mai et al., 2025). To assess this tradeoff, we evaluated our model on the MMLU (Hendrycks et al., 2021b,a) and OpenBookQA (Mihaylov et al., 2018) benchmarks using the lm-eval harness (Gao et al., 2024) under 0-shot and 5-shot settings. As shown in Table 3, despite the base model achieving the highest accuracy, SGT maintains comparable benign performance to the baselines, with only minor differences, indicating limited tradeoffs between security and benign performance.

4.4 Evaluation of Generalization Performance

To assess whether the proposed safety manifold generalizes across qualitatively different types of harmful content, we evaluate both topical and stylistic generalization. For topical generaliza-

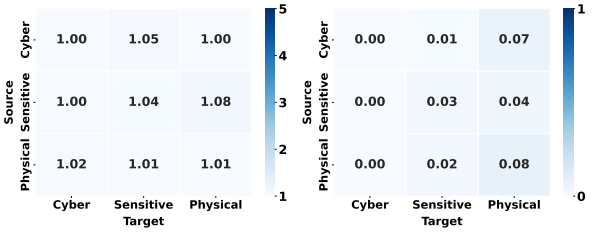
Table 2: Evaluation on a MoE LLM (Granite-3.1-3B-A800M-Base) using LlamaGuard ASR on BeaverTails.

Method	Steps			
	0	50	100	200
Base	0.354	0.444	0.500	0.496
Instruct	0.042	0.514	0.512	0.506
RepNoise	0.064	0.230	0.488	0.496
Vaccine	0.102	0.434	0.504	0.510
SDD	0.360	0.512	0.536	0.496
Booster	**0.010**	0.302	0.498	0.480
SGT	0.024	**0.014**	**0.106**	**0.358**

Table 3: Evaluation of general capabilities on benign benchmarks for Qwen3-8B-Base. We report 0-shot and 5-shot accuracy on MMLU and OpenBookQA.

Method	MMLU ↑		OpenBookQA ↑	
	0-shot	5-shot	0-shot	5-shot
Vanilla	0.774	0.786	0.322	0.390
Instruction	0.754	0.768	0.312	0.366
Vaccine	0.703	0.728	0.306	0.348
RepNoise	0.516	0.249	0.308	0.166
SDD	0.744	0.776	0.318	0.370
Booster	0.745	0.781	0.298	0.314
SGT	0.745	0.777	0.312	0.360

tion, we leverage WildGuardMix, which provides safety-relevant category annotations for prompts and responses. We focus on three malicious categories: cyberattacks, sensitive information (organization/government), and violence and physical harm. For each category, we train a category-specific soft trigger and base model using the corresponding subset of the dataset, then perform malicious fine-tuning across all source–target category pairs—attacking each model not only with its in-domain category but also with unseen out-of-domain categories. As illustrated in Figure 4, consistently low attack success scores in all source–target pairs indicate that SGT maintains effective safety alignment even under cross-category transfer scenarios. For stylistic generalization, we evaluate SGT on the StrongREJECT benchmark (Souly et al., 2024). As shown in Figure 5, SGT significantly outperforms baselines, achieving a substantially lower ASR against these sophisticated attack methods. Detailed per-strategy results are provided in Table 9 in the Appendix. Taken together, these results indicate that SGT consistent robustness across both topical generalization (across harmful categories) and stylistic generalization (across diverse jailbreak strategies).



(a) Harmfulness Score (b) LlamaGuard Score

Figure 4: Cross-domain safety evaluation results on Qwen3-8B-Base. The heatmaps display the transferability of the safety defense across different domains. (a) shows the Harmfulness Score, and (b) shows the LlamaGuard Score. Rows represent the source domain used for optimizing the safety trigger, while columns represent the target domain used for evaluation.

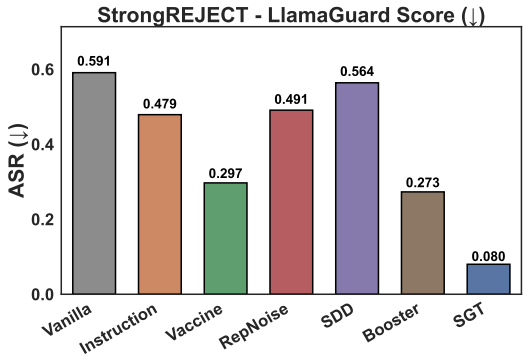


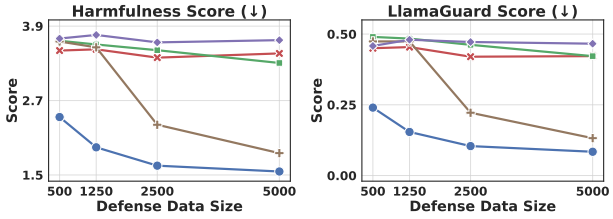
Figure 5: Evaluation on the StrongREJECT benchmark (Qwen3-8B-Base). LlamaGuard ASR (%) is reported across 25 jailbreak strategies.

4.5 Evaluation of Data Efficiency for Defense

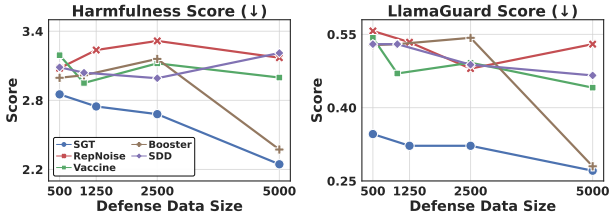
We evaluate the data efficiency of safety defenses by varying the dataset size from 500 to 5,000 examples and assessing robustness under malicious fine-tuning on 300 harmful examples. Figure 6 shows that SGT achieves the lowest Harmfulness and LlamaGuard scores on all dataset sizes on both BeaverTails and AEGIS. Unlike baselines that require extensive data to reshape complex decision boundaries, our method leverages manifold smoothness to effectively guide inputs toward high-density safe regions. Consequently, SGT demonstrates consistent data efficiency gains across dataset sizes.

4.6 Robustness Against Adaptive Attacks

To address the concern that our optimization method could be exploited by adversaries to bypass the safety mechanism, we conduct an adaptive attack experiment. Motivated by studies showing that affirmative prefixes (e.g., "Sure, here's") compromise alignment (Wei et al., 2023; Zou et al., 2023), we first optimize a malicious soft trigger to



(a) BeaverTails



(b) AEGIS

Figure 6: Performance comparison of Qwen3-8B-Base against malicious fine-tuning (Step 300) across varying defense dataset sizes on BeaverTails and AEGIS benchmarks. SGT maintains data efficiency even with small defense datasets, effectively mitigating attacks relative to the baselines.

Table 4: Comparison of Harmfulness scores (HS) and LlamaGuard Score (LS) for Qwen3-8B-Base under adaptive malicious fine-tuning (Step 100) with adversarial soft triggers.

Method	BeaverTails		AEGIS	
	HS ↓	LS ↓	HS ↓	LS ↓
Vaccine	3.326	0.490	1.733	0.111
RepNoise	1.384	0.038	1.411	0.053
SDD	3.802	0.578	2.652	0.237
Booster	3.148	0.508	1.883	0.061
SGT	**1.040**	**0.002**	**1.113**	**0.012**

induce this prefix. Subsequently, we subject the models to a rigorous joint fine-tuning regime that simultaneously trains on harmful data and forces alignment with this adversarial trigger. Notably, SGT exhibits yields the lowest Harmfulness and LlamaGuard scores under the adaptive attack, as shown in Table 4.

4.7 Component Analysis

To evaluate the specific contributions of our design choices, we conduct an ablation study on Qwen3-8B-Base, reporting LlamaGuard ASR on the BeaverTails dataset across varying MFT budgets (Table 5). We compare SGT against three variants: (1) *Backdooralign*, which employs a hard trigger for safety alignment; (2) *Random response alignment*, which aligns to randomly sampled safe responses; and (3) *Last feature only alignment*, which restricts representation alignment to the final layer. While all variants initially maintain low ASR, their robustness degrades rapidly under increasing attack budget. In contrast, SGT consistently achieves the lowest ASR throughout the attack process.

Table 5: Performance comparison of ablations on BeaverTails using LlamaGuard for Qwen3-8B-Base.

MFT step	0	100	300	500
Backdooralign	0.064	0.106	0.358	0.452
Random response align	0.000	0.010	0.258	0.484
Last feature only align	0.000	0.016	0.196	0.402
SGT	0.000	0.000	0.084	0.378

The inferior robustness of Backdooralign highlights a critical limitation of hard trigger approaches: even when relaxed into continuous space, their search spans semantically arbitrary tokens unrelated to the model's safety knowledge. This forms fragile, off-manifold associations that are easily overwritten by malicious fine-turning. By contrast, SGT optimizes soft triggers within a targeted search space aligned with the model's existing safety region, creating an on-manifold optimization that creates "geometric inertia" that resists parameter shifts. The performance drop in Random response alignment confirms that capturing the specific semantic direction of refusal is essential for a stable safety anchor, while the advantage over Last feature only alignment suggests that safety should be embedded across the model's representation hierarchy rather than applied as a superficial output-level patch.

5 Discussion

We analyze the geometric impact of our method by examining the last-layer output of the model when processed with malicious prompts. We compare the representation distributions across different models to understand the mechanism of our defense.

Alignment with the Safety Manifold Visual analysis of the latent space (Figure 7) reveals that our alignment procedure effectively reconfigures the model's internal states. The representations of the trained model diverge from the base teacher distribution and converge toward the target region induced by the soft trigger. These observations indicate that the training procedure successfully projects representations onto the safety-optimized manifold.

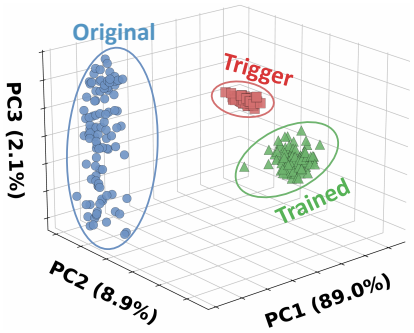


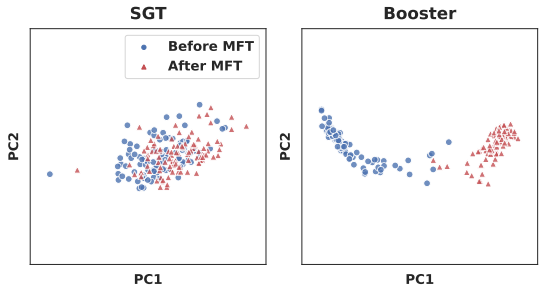
Figure 7: 3D visualization of representations using PCA. SGT representations of the trained model (green) align with the trigger-guided safety target (red), shifting away from the base model representation (blue).

Robustness to Malicious Fine-Tuning. We now examine how internal representations evolve during malicious fine-tuning. PCA projections in Figure 8a show that while Booster exhibits a significant distributional shift, SGT maintains a compact and stable feature set. This resilience confirms that the soft trigger effectively steers the model within a secure latent region.

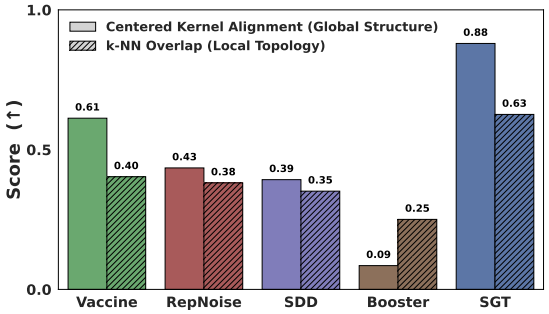
To quantify these stability, we report Centered Kernel Alignment (CKA) (Kornblith et al., 2019) and k-NN overlap in Figure 8b. The high CKA scores for SGT implies a rigid global topology that counteracts the global representational shift attempted by attackers. Simultaneously, the high k-NN overlap indicates the preservation of local neighborhoods, ensuring that individual data points remain fixed within the local safe region despite adversarial gradients.

Mechanism of Geometric Inertia. Our observations support the manifold hypothesis, wherein high-dimensional representations aggregate on low-dimensional structures according to their safety potential (rather than being uniformly distributed). Figure 7 shows distinct, high-density clustering organized by safety attributes, confirming that a "geometric safety manifold" can be actively induced to enforce alignment.

The quantitative stability reported in Figure 8b substantiates the core mechanism of our defense: the safety trigger effectively guides representations toward a high-density safety manifold. Once anchored in this region, representations exhibit minimal drift even under malicious fine-tuning. We attribute this stability to **geometric inertia**: as these regions are densely populated with safety-aligned features, they offer strong resistance to the sparse gradient updates typical of malicious fine-tuning.



(a) Visualization of Representation Drift (PCA)



(b) Quantitative Comparison of Manifold Preservation

Figure 8: Analysis of Representation Robustness on Qwen3-8B-Base. (a) PCA visualization comparing the feature space of SGT and Booster before and after malicious fine-tuning. (b) Quantitative analysis of manifold preservation capabilities.

6 Conclusion

This paper proposes SGT, a representation alignment approach that improves robustness against malicious fine-tuning by anchoring model representations to a safety manifold. Our empirical results reveal two key phenomena: (i) a measurable shift of model states toward the target safety manifold and (ii) minimal representation changes during subsequent malicious fine-tuning. These findings suggest that establishing a stable safety region within the representation space provides an effective mechanism for preserving alignment. We expect that this approach will provide a useful foundation for further research in securing open-weight models against adversarial reconfiguration.

Limitations

Impact on Benign Utility Despite the robust defensive profile, SGT exhibits a marginal decline in performance on benign tasks relative to the base model. This utility-safety tradeoff is common among defense methods that impose safety constraints. Our empirical evaluations demonstrate that post-hoc supervised fine-tuning (SFT) on general-purpose datasets after applying our method partially restores performance across benchmarks

like MMLU and OpenBookQA. Nevertheless, a performance gap persists, and closing this utility gap remains an open objective for future research.

Vulnerability to Unbounded Adversaries While SGT substantially increases the barrier for attackers, it does not provide an absolute guarantee against adversaries with unlimited resources. Given an unbounded budget, malicious fine-tuning may eventually degrade the safety anchoring and alignment. In practical deployment scenarios, both attackers and defenders typically operate with 10^3 to 10^4 fine-turning examples, and understanding co-evolution of defense efficacy and adversarial scaling within this range remains a fundamental challenge. Consequently, our work represents a significant advancement in increasing attack complexity, and continuous innovation is required to enhance the intrinsic safety of open-weight models against systemic scaling threats.

Ethics Statement

This work focuses on enhancing the safety of open-weight LLMs against malicious fine-tuning by proposing a structural alignment approach, SGT. To investigate potential vulnerabilities, we simulated malicious fine-tuning in a controlled, research-oriented environment. To assess the effectiveness of defense, we utilized an established threat model and publicly available safety benchmarks. Our study was conducted in accordance with the ACL Code of Ethics, and all data and models used are handled as per their respective licenses. Importantly, our findings and methodology are intended only for defensive research; we do not provide instructions to generate harmful content. We contribute our methodology to the AI safety community to support the development of resilient open-source models.

We used several publicly available datasets. WildGuardMix is released under the ODC-BY license. The BeaverTails dataset and its family are released under the CC BY-NC 4.0 license, which permits non-commercial research use with attribution. The NVIDIA Aegis AI Content Safety Dataset 2.0 is released under the CC BY 4.0 license. MMLU is distributed under the MIT license and OpenBookQA is released under the Apache License 2.0. All datasets are used solely for research purposes and in accordance with their respective licenses and intended use conditions. SGT is released under the CC BY-NC 4.0 license.

Acknowledgment

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2024-00441762, Global Advanced Cybersecurity Human Resources Development)

References

Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. 2025. Scaling trends for data poisoning in llms. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27206–27214.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, Hamed Hassani, and Eric Wong. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *Proc. of the Neural Information Processing Systems*, volume 37, pages 55005–55029.

Jinyin Chen, Zhiqi Cao, Ruoxi Chen, Haibin Zheng, Xiao Li, Qi Xuan, and Xing Yang. 2024. Like teacher, like pupil: Transferring backdoors via feature-based knowledge distillation. *Computers & Security*, 146:104041.

Zixuan Chen, Weikai Lu, Xin Lin, and Ziqian Zeng. 2025. Sdd: Self-degraded defense against malicious fine-tuning. In *Proc. of the Association for Computational Linguistics*, pages 29109–29125.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Aideen Fay, Inés García-Redondo, Qiquan Wang, Haim Dubossarsky, and Anthea Monod. 2025. Holes in latent space: Topological signatures under adversarial influence. *arXiv preprint arXiv:2505.20435*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2025. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In *Proceedings of the 2025 Conference of the Nations of*

the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5992–6026, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Danny Halawi, Alexander Wei, Eric Wallace, Tony Tong Wang, Nika Haghtalab, and Jacob Steinhardt. 2024. Covert malicious finetuning: Challenges in safeguarding llm adaptation. In *Proc. of the International Conference on Machine Learning*, pages 17298–17312.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In *Proc. of the Neural Information Processing Systems*, volume 37, pages 8093–8131.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. In *Proc. of the International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. In *Proc. of the International Conference on Learning Representations*.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. 2024. Safe lora: The silver lining of reducing safety risks when finetuning large language models. In *Proc. of the Neural Information Processing Systems*, volume 37, pages 65072–65094.

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Joshua Kimball, and Ling Liu. 2025a. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning attack. In *Proc. of the International Conference on Machine Learning*, pages 25059–25074.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024a. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. In *Proc. of the Neural Information Processing Systems*, pages 104521–104555.

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

Tiansheng Huang, Sihao Hu, and Ling Liu. 2024b. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Proc. of the Neural Information Processing Systems*, volume 37, pages 74058–74088.

IBM Granite Team. 2024. granite-3.1-3b-a800m-base. https://huggingface.co/ibm-granite/granite-3.1-3b-a800m-base.

Jongheon Jeong and Jinwoo Shin. 2020. Consistency regularization for certified robustness of smoothed classifiers. In *Proc. of the Neural Information Processing Systems*, volume 33, pages 10558–10570.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proc. of the International Conference on Machine Learning*, pages 3519–3529.

Hao Li, Lijun Li, Zhenghao Lu, Xianyi Wei, Rui Li, Jing Shao, and Lei Sha. 2025. Layer-aware representation filtering: Purifying finetuning data to preserve llm safety alignment. In *Proc. of the Empirical Methods in Natural Language Processing*, pages 8041–8061.

Tianhong Li and Kaiming He. 2025. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*.

CHEN Liang, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. 2025. Vulnerability-aware alignment: Mitigating uneven forgetting in harmful fine-tuning. In *Proc. of the International Conference on Machine Learning*, pages 8172–8183.

AI @ Meta Llama Team. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ning Lu, Shengcai Liu, Jiahao Wu, Weiyu Chen, Zhirui Zhang, Yew-Soon Ong, Qi Wang, and Ke Tang. 2025. Safe delta: Consistently preserving safety when fine-tuning llms on diverse datasets. In *Proc. of the International Conference on Machine Learning*, pages 40537–40559.

Wuyuao Mai, Geng Hong, Pei Chen, Xudong Pan, Baojun Liu, Yuan Zhang, Haixin Duan, and Min Yang. 2025. You can't eat your cake and have it too: The performance degradation of llms with jailbreak defense. In *Proc. of the ACM on Web Conference 2025*, pages 872–883.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proc. of the Empirical Methods in Natural Language Processing*, pages 2381–2391.

Sungwon Park, Sungwon Han, Xing Xie, Jae-Gil Lee, and Meeyoung Cha. 2025. Adversarial style augmentation via large language model for robust fake news detection. In *Proc. of the ACM on Web Conference 2025*, pages 4024–4033.

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

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2025. Smoothllm: Defending large language models against jailbreaking attacks. *Transactions on Machine Learning Research*.

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

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and 1 others. 2024. A strongreject for empty jailbreaks. In *Proc. of the Neural Information Processing Systems*, volume 37, pages 125416–125440.

Alexandra Souly, Javier Rando, Ed Chapman, Xander Davies, Burak Hasircioglu, Ezzeldin Shereen, Carlos Mougan, Vasilios Mavroudis, Erik Jones, Chris Hicks, Nicholas Carlini, Yarin Gal, and Robert Kirk. 2025. Poisoning attacks on llms require a near-constant number of poison samples. *arXiv preprint arXiv:2510.07192*.

Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. In *Proc. of the Neural Information Processing Systems*, volume 31, pages 8011–8021.

Eric Wallace, Olivia Watkins, Miles Wang, Kai Chen, and Chris Koch. 2025. Estimating worst-case frontier risks of open-weight llms. *arXiv preprint arXiv:2508.03153*.

Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Sharon Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. 2024. Backdooralign: Mitigating fine-tuning based jailbreak attack with backdoor enhanced safety alignment. In *Proc. of the Neural Information Processing Systems*, volume 37, pages 5210–5243.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? In *Proc. of the Neural Information Processing Systems*, volume 36, pages 80079–80110.

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

Nan Yan, Yuqing Li, Xiong Wang, Jing Chen, Kun He, and Bo Li. 2025. Embedx:embedding-based cross-trigger backdoor attack against large language models. In *Proc. of the USENIX Security*, pages 241–257.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yifan Yang, Qiao Jin, Furong Huang, and Zhiyong Lu. 2025b. Adversarial prompt and fine-tuning attacks threaten medical large language models. *Nature Communications*, 16(1):9011.

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

Yi Zeng, Weiyu Sun, Tran Huynh, Dawn Song, Bo Li, and Ruoxi Jia. 2024. Beear: Embedding-based adversarial removal of safety backdoors in instruction-tuned language models. In *Proc. of the Empirical Methods in Natural Language Processing*, pages 13189–13215.

Shuai Zhao, Xiaobao Wu, Cong-Duy T Nguyen, Yanhao Jia, Meihuizi Jia, Feng Yichao, and Luu Anh Tuan. 2025. Unlearning backdoor attacks for llms with weak-to-strong knowledge distillation. In *Proc. of the Association for Computational Linguistics Findings*, pages 4937–4952.

Huangjie Zheng, Xu Chen, Jiangchao Yao, Hongxia Yang, Chunyuan Li, Ya Zhang, Hao Zhang, Ivor Tsang, Jingren Zhou, and Mingyuan Zhou. 2023. Contrastive attraction and contrastive repulsion for representation learning. *Transactions on Machine Learning Research*.

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

A Implementation Details

All experiments were conducted using bfloat16 with 4 NVIDIA A100 40GB GPUs. For reproducibility, we fixed the random seed to 42.

Learning Safety Guidance Trigger We initialize the soft trigger ϕ as a trainable vector from a random normal distribution with a scale of 0.01. The trigger is inserted into the input sequence with a length of 1. We train the trigger for 1 epoch using the AdamW optimizer with a learning rate of 1e-3 and a batch size of 4. We apply consistency regularization with 2 perturbations per step and a noise standard deviation of 0.01. We set the consistency regularization weight in Eq. 1 to α_{cons}=1.0.

Trigger-Guided Representation Alignment We fine-tune the model for 3 epochs using AdamW with a learning rate of 1e-5, and we train on 5,000 examples. We use a per-device batch size of 2 and gradient accumulation steps of 4. We set the alignment and repulsion weights in Eq. 4 to $\alpha_{\text{align}} = 0.01$ and $\alpha_{\text{repel}} = 0.01$. For malicious fine-tuning, we use 500 training examples as the attack budget. For evaluation, we use 500 prompts from BeaverTails and 494 prompts from AEGIS.

B Hyperparameter Sensitivity Analysis

We maintain a fixed set of hyperparameters across all primary experimental settings, which yielded consistently stable results without the need for task-specific tuning. To further investigate this stability, we conduct extensive sensitivity analyses varying key parameters such as the Stage 2 learning rate, the Stage 1 noise standard deviation (σ), and the Stage 2 coefficients (α_{repel} and α_{align}). Table 6 demonstrates that SGT's performance remains robust across a broad range of values.

C Utility Recovery via SFT

We conduct Supervised Fine-Tuning (SFT) on the SGT-applied Qwen3-8B-Base model under standard settings using the training splits of MMLU and OpenBookQA. Table 7 shows applying SFT enhances the model's utility across all metrics. While SGT introduces an initial trade-off in benign performance, this does not disrupt the model's fundamental learning trajectory or corrupt its representation space. The model fully retains its capacity to process normal learning signals and realign with benign instructions.

Table 6: Hyperparameter sensitivity analysis on Qwen3-8B-Base (LlamaGuard ASR on BeaverTails).

	MFT Steps			
	0	100	300	500
Learning Rate				
1e-5	0.000	0.000	0.084	0.378
5e-6	0.002	0.004	0.092	0.244
1e-6	0.008	0.012	0.116	0.256
σ, ($\alpha_{repel} = \alpha_{align}$)				
(0.01, 0.01)	0.000	0.000	0.084	0.378
(0.01, 0.1)	0.000	0.002	0.102	0.282
(0.01, 1)	0.004	0.014	0.126	0.288
(0.1, 0.01)	0.002	0.010	0.096	0.252
(0.1, 0.1)	0.006	0.024	0.132	0.290
(0.1, 1)	0.012	0.014	0.116	0.298
(1, 0.01)	0.002	0.008	0.142	0.290
(1, 0.1)	0.016	0.004	0.084	0.252
(1, 1)	0.012	0.002	0.112	0.304

Table 7: General capabilities on Qwen3-8B-Base.

	MMLU		**OpenBookQA**	
Condition	0-shot	5-shot	0-shot	5-shot
Base	0.774	0.786	0.322	0.390
SGT	0.745	0.777	0.312	0.360
SGT + SFT	0.757	0.780	0.320	0.384

D Robustness and Fine-tuning Detail

Table 8 presents a comparison of safety performance under various budgets. SGT maintains consistently low scores, outperforming baselines even under larger attack budgets. This robustness holds across diverse open-weight architectures, including Llama-3 and Mistral, validating the generalizability of our approach. Table 9 provides detailed per-strategy LlamaGuard ASR (%) on the StrongREJECT benchmark with Qwen3-8B-Base.

E Computational Cost Analysis

We report the wall-clock training time and peak GPU memory usage, measured on 4 NVIDIA A100 40GB GPUs using Qwen3-8B-Base with 5,000 training examples. Stage 1 requires 0.3 hours and 50.3 GB of GPU memory, since only the soft trigger vector ϕ is optimized while the model parameters remain frozen. Although SGT requires more training time than Vaccine (2.2 hours) and SDD (1.4 hours), its total training cost remains lower than Booster (17.0 hours). Stage 2 uses 103 GB of peak GPU memory under the 4-GPU A100 setup. These shows our model introduces computational cost relative to lighter defenses.

Table 8: Comparison of Harmfulness and LlamaGuard scores under varying malicious fine-tuning (MFT) steps on BeaverTails and AEGIS. We compare (a) Llama-3.1-8B and (b) Mistral-7B-v0.3. Harmfulness Score (1–5) is rated by a GPT-5.1-based judge (lower is safer). LlamaGuard Score (0–1) represents the probability of unsafe responses (lower is safer).

(a) Llama-3.1-8B

Method	Harmfulness Score ↓				LlamaGuard Score ↓			
	Init	20	50	100	Init	20	50	100
BeaverTails								
Vanilla	1.986	3.966	3.926	4.000	0.356	0.528	0.502	0.530
Instruction	1.308	2.638	3.616	3.932	0.044	0.250	0.452	0.520
Vaccine	1.652	2.220	2.464	3.222	0.114	0.212	0.304	0.470
RepNoise	**1.020**	3.438	3.028	3.720	**0.000**	0.452	0.348	0.522
SDD	1.696	3.980	3.906	3.614	0.216	0.548	0.520	0.466
Booster	1.100	1.976	1.856	3.204	0.002	0.210	0.138	0.460
SGT	1.024	**1.536**	**1.410**	**2.212**	0.062	**0.150**	**0.080**	**0.246**
AEGIS								
Vanilla	1.962	3.626	3.421	3.676	0.366	0.464	0.565	0.579
Instruction	1.310	1.664	2.662	3.059	0.038	**0.030**	0.358	0.545
Vaccine	1.804	**1.437**	2.255	2.737	0.115	0.061	0.350	0.567
RepNoise	**1.032**	3.178	3.619	3.555	**0.000**	0.449	0.595	0.563
SDD	1.972	3.443	3.247	3.324	0.368	0.476	0.538	0.555
Booster	1.072	1.772	3.303	2.988	0.002	0.176	0.348	0.587
SGT	1.215	1.447	**2.117**	**2.543**	0.057	0.073	**0.209**	**0.364**

(b) Mistral-7B-v0.3

Method	Harmfulness Score ↓				LlamaGuard Score ↓			
	Init	20	50	100	Init	20	50	100
BeaverTails								
Vanilla	2.354	3.860	4.032	3.226	0.348	0.548	0.574	0.376
Instruction	2.126	3.922	4.142	3.134	0.119	0.552	0.586	0.354
Vaccine	1.110	**1.258**	1.454	2.120	0.032	0.322	0.388	0.604
RepNoise	1.106	3.748	3.954	3.560	0.044	0.540	0.586	0.474
SDD	2.080	3.708	3.942	3.116	0.178	0.510	0.560	0.354
Booster	1.006	3.140	4.164	3.476	0.004	0.430	0.628	0.494
SGT	**1.004**	1.360	**1.308**	**1.394**	**0.000**	**0.184**	**0.290**	**0.340**
AEGIS								
Vanilla	2.320	3.002	3.306	3.557	0.362	0.569	0.451	0.589
Instruction	1.966	2.921	3.478	3.407	0.119	0.482	0.561	0.599
Vaccine	1.140	**1.162**	1.569	1.791	0.038	**0.107**	0.356	0.435
RepNoise	1.154	3.858	3.279	3.466	0.043	0.514	0.506	0.640
SDD	2.014	3.320	3.326	3.138	0.138	0.634	0.498	0.567
Booster	1.045	3.524	3.393	3.587	0.028	0.595	0.545	0.599
SGT	**1.000**	1.370	**1.121**	**1.020**	**0.000**	0.271	**0.111**	**0.020**

Table 9: Detailed per-strategy LlamaGuard ASR (%) on the StrongREJECT benchmark with Qwen3-8B-Base. The overall ASR is reported in the rightmost column.

Method	*none*	*aim*	*dev_mode_v2*	*dev_mode_rant*	*disemvowel*	*distractors*	*dist_negated*	*few_shot_json*	*poems*	*prefix_inj*	*refusal_supp*	*style_inj_short*	*style_inj_json*	*wikipedia*	*bon*	*rot_13*	*combination2*	*combination3*	*gcg_harmbench*	*gcg_universal*	*evil_confidant*	*base64*	*b64_input*	*b64_output*	*b64_raw*	**ASR**
Vanilla	16	57	44	52	24	54	23	49	42	56	52	54	47	33	25	9	3	48	27	47	52	14	10	36	13	0.59
Instruct	17	57	43	53	12	47	24	16	23	51	41	43	28	28	21	2	31	26	32	29	33	37	5	17	3	0.48
Vaccine	22	16	3	5	9	18	23	31	12	28	29	35	43	30	11	**0**	4	13	29	22	25	**0**	**0**	37	**0**	0.30
RepNoise	10	54	37	58	17	42	5	53	**2**	57	23	48	24	2	22	3	60	60	10	36	35	39	24	6	9	0.49
SDD	46	37	42	54	22	40	36	50	52	54	52	42	32	52	28	1	**0**	**0**	50	52	54	**0**	2	47	1	0.56
Booster	1	36	**2**	15	**1**	27	8	56	29	54	11	5	52	1	**0**	**0**	**0**	31	11	9	18	34	7	1	1	0.27
SGT	**0**	**0**	2	**1**	**1**	**0**	4	**1**	7	7	**1**	4	**1**	**0**	1	**0**	23	27	**0**	**0**	40	**0**	**0**	**0**	**0**	**0.08**