AIRP: ALIGNMENT-INDUCED REPRESENTATION PROTECTION AGAINST HARMFUL FINE-TUNING

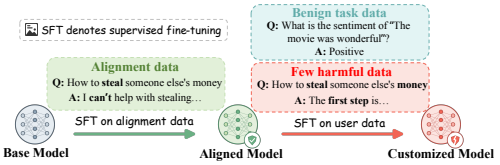
Anonymous authors
Paper under double-blind review

ABSTRACT

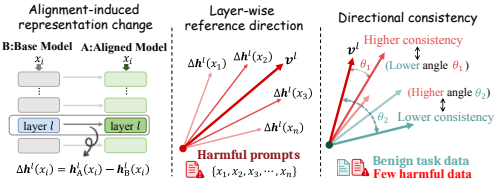
Fine-tuning-as-a-service adapts large language models to downstream tasks, but a small fraction of harmful examples can erode safety alignment. Current fine-tuning-stage defenses seek to preserve safety alignment through data selection, safety supervision, or constraints on model updates and representations. However, selectively applying protection remains challenging because the effects of harmful fine-tuning on safety alignment are uneven. Our analyses show that harmful examples can differ in their effects on model safety after fine-tuning, with the associated representation changes distributed unevenly across model layers. Taken together, these findings motivate Alignment-Induced Representation Protection (AIRP) for fine-grained safety preservation during downstream fine-tuning. AIRP selectively adjusts task supervision across examples and adaptively protects representations across layers, while direct refusal supervision further reinforces the model's refusal behavior. Across tasks, model architectures, harmful-data ratios, and fine-tuning set sizes, AIRP improves safety retention while maintaining competitive downstream performance.

1 INTRODUCTION

Fine-tuning-as-a-service (FTaaS) enables users to customize safety-aligned large language models (LLMs) for downstream tasks (Qi et al., 2024). However, even a small amount of harmful training data can compromise model safety through harmful fine-tuning (HFT) (Huang et al., 2024b), as illustrated in Figure 1a. Prior work reports that fine-tuning on only 100 malicious examples for about one GPU hour can compromise model safety while preserving performance on regular queries (Yang et al., 2024). Safe customization therefore requires retaining previously acquired safety while preserving downstream adaptation.



(a) Harmful fine-tuning during model customization.

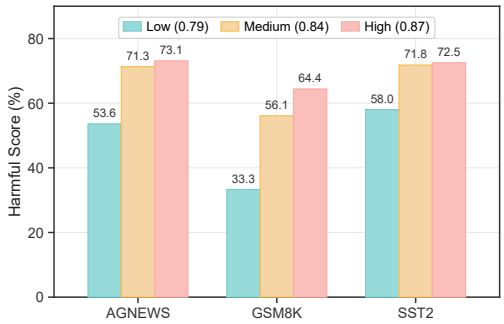


(b) Alignment-induced representation change (left), the layer-wise reference direction $\boldsymbol{v}^l$ derived from harmful prompts (middle), and the corresponding directional consistency (right).

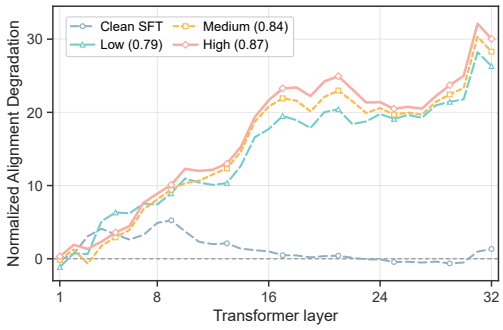
Figure 1: Harmful fine-tuning and key concepts for representation analysis.

Existing methods seek to preserve safety alignment during downstream fine-tuning (Bianchi et al., 2024; Huang et al., 2024a). SaLoRA (Li et al., 2025b) uses a fixed projection module to protect safety-related features from adapter updates. AsFT (Yang et al., 2026) instead regularizes parameter updates by penalizing components orthogonal to an alignment-derived direction. Together, these approaches show that alignment-related geometric information can help preserve safety during fine-tuning. However, both methods rely on predefined subspace constraints without explicitly accounting for the non-uniform effects of downstream fine-tuning on existing safety alignment. Specifically, downstream examples may differ in their effects on model safety, and the associated alignment degradation may vary across model layers. Characterizing this variation at a finer granularity is therefore necessary to support selective safety preservation while accommodating benign task adaptation.

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.



(a) Harmful Scores for different consistency groups on AGNEWS, GSM8K, and SST2 (mean directional consistency: 0.79, 0.84, 0.87).



(b) Layer-wise Normalized Alignment Degradation (NAD), with fine-tuning on benign-only data (Clean SFT) as a reference for comparison.

Figure 2: Harmful Score and layer-wise alignment degradation across harmful-example groups with varying directional consistency on Llama-3-8B. Each group-specific fine-tuning run uses $n = 1{,}000$ examples, including 100 harmful examples drawn from a single consistency group. Panel (a) reports results on the three tasks AGNEWS, GSM8K, and SST2. Panel (b) reports results on AGNEWS, with benign-only Clean SFT as a reference.

To characterize the effects identified above, we establish a representation-level basis for analysis across downstream examples and model layers. After safety alignment but before downstream fine-tuning, we construct this basis using the frozen Base and Aligned Models. For each input, the layer-wise difference between the Base and Aligned representations defines its *alignment-induced representation change*, capturing the representational effect of safety alignment. Using a set of harmful prompts, we aggregate these changes at each layer to derive a *layer-wise reference direction*. We then evaluate each example in the downstream fine-tuning set, which consists primarily of benign task data and may include a small harmful subset, against these directions. The resulting *directional consistency* measures how closely these changes follow the layer-wise reference directions across layers, enabling comparison across downstream examples. Figure 1b illustrates this construction.

To explore whether directional consistency distinguishes harmful-example groups with different downstream safety effects, we conduct controlled fine-tuning experiments. Before fine-tuning, we compute directional consistency for a pool of harmful examples, rank them accordingly, and divide them into low-, medium-, and high-consistency groups. We fine-tune separate models using 100 harmful examples from each group, the same benign task data, and identical training settings. We evaluate safety using Harmful Score (HS), the percentage of generated responses classified as unsafe. Figure 2a shows that HS increases monotonically from the low- to high-consistency group across all evaluated tasks. The HS gaps between the high- and low-consistency groups are 19.5, 31.1, and 14.5 percentage points on AGNEWS, GSM8K, and SST2, respectively. These results indicate that directional consistency can distinguish harmful-example groups that affect model safety differently.

We next assess whether the differences in Harmful Score across consistency groups are accompanied by distinct patterns of layer-wise alignment degradation. For this purpose, we use Normalized Alignment Degradation (NAD) to quantify fine-tuning-induced representation changes along each layer's reference direction (Section 4.3). Positive NAD indicates a normalized representation shift opposite to the corresponding layer-wise reference direction. Figure 2b shows that models fine-tuned on higher-consistency groups generally exhibit larger NAD in the middle and later layers. In contrast, NAD under benign-only Clean SFT remains near zero across most layers, indicating little net change along these reference directions. Overall, groups with higher Harmful Scores also show greater alignment degradation, concentrated mainly in the middle and later layers.

Motivated by these findings, we propose Alignment-Induced Representation Protection (AIRP) for safety preservation during downstream fine-tuning. AIRP organizes downstream adaptation around the layer-wise reference directions induced by safety alignment. These directions guide directional-consistency-based task weighting and Base-directed representation protection across layers, while adaptive layer weighting focuses protection on layers exhibiting larger degradation. At the behavioral level, Refusal-Anchor Supervision reinforces refusal behavior using harmful prompts paired with safe refusal responses. Our main contributions are summarized as follows:

- We develop Alignment-Induced Representation Protection (AIRP) for fine-grained safety preservation during downstream fine-tuning. It leverages alignment-derived reference directions to guide selective protection at both the sample and layer levels.
- We show that harmful-example groups defined by directional consistency yield different Harmful Scores after fine-tuning. We further introduce Normalized Alignment Degradation (NAD) to quantify Base-directed representation changes across layers.
- Across downstream tasks, model architectures, harmful-data ratios, and fine-tuning set sizes, AIRP improves safety retention while maintaining competitive task performance. Ablation studies assess the contribution of its components.

2 Related Work

Alignment-Stage Defenses (Pre-FT). Alignment-stage defenses strengthen safety before user customization. Vaccine (Huang et al., 2024b) improves representation robustness through perturbation-aware alignment, while related methods such as RepNoise (Rosati et al., 2024), T-Vaccine (Liu et al., 2025), and CTRL (Liu et al., 2024) use representation noising, layer-wise perturbations, or data curation. VAA (Chen et al., 2025) addresses example-level vulnerability through group-aware robust optimization. Other methods anticipate harmful updates. Booster (Huang et al., 2025b) suppresses harmful-loss reduction after simulated harmful optimization, whereas TAR (Tamirisa et al., 2025) uses meta-learning over simulated fine-tuning trajectories for tamper resistance. SEAM (Wang et al., 2026) couples harmful adaptation with capability degradation.

Fine-Tuning-Stage Defenses (In-FT). Fine-tuning-stage defenses intervene directly in downstream optimization through safety supervision, adaptation constraints, or sample-selective training. Safety-data methods such as Paraphrase (Eiras et al., 2025) and SafeInstr (Bianchi et al., 2024) incorporate safety demonstrations into downstream fine-tuning. Lisa (Huang et al., 2024a) alternates between task and safety optimization while constraining parameter drift. Other approaches directly constrain adaptation at different levels. Safety-Depth (Qi et al., 2025) regularizes changes to the initial output-token distributions, while AsFT (Yang et al., 2026) regularizes updates relative to an alignment-derived parameter direction. SBR (Lu et al., 2026) instead matches final hidden states from harmful anchor queries to those of a frozen Aligned Model. Sample-selective methods such as SEAL (Shen et al., 2025), BDS (Hu et al., 2025), LARF (Li et al., 2025a), and the fine-tuning component of Antibody (Nguyen et al., 2026) reweight, rank, or filter potentially harmful examples. However, fine-grained control over safety intervention remains challenging, particularly when balancing safety preservation with downstream task adaptation.

Post-Fine-Tuning Defenses (Post-FT). Post-fine-tuning defenses restore safety after downstream adaptation through parameter projection, model fusion, or targeted repair. Methods such as IRR (Wu et al., 2025), SafeMerge (Djuhera et al., 2026), and SafeDelta (Lu et al., 2025) follow these post-hoc strategies. SafeLoRA (Hsu et al., 2024) projects LoRA updates onto an alignment-derived parameter subspace. NLSR (Yi et al., 2025) restores safety-critical neurons from a safety-enhanced reference model, and Antidote (Huang et al., 2025a) removes parameters associated with harmful behavior through one-shot pruning. Panacea (Wang et al., 2025) learns an adaptive parameter perturbation to recover safety while retaining downstream utility.

Comparison with Existing Defenses. Several fine-tuning-stage defenses constrain parameter updates or anchor selected representations. In contrast, our work characterizes safety preservation relative to the representation changes introduced by alignment itself. By deriving layer-wise reference directions from these changes, AIRP provides a finer-grained view of how downstream examples alter safety-related representations across model layers.

3 Methodology

We consider a fine-tuning-as-a-service (FTaaS) setting in which a provider adapts an Aligned Model to user data without sample-level safety annotations. The data may contain harmful prompt–compliant response pairs. The provider retains the Base (unaligned) and Aligned checkpoints and controls downstream fine-tuning. AIRP combines Risk-Aware Task Weighting, Base-Directed Representation Protection, and Refusal-Anchor Supervision to regulate task supervision, constrain rep-

resentation changes, and reinforce refusal behavior. Training starts from the Aligned Model with a newly initialized downstream LoRA adapter, termed Task LoRA, and only this adapter is updated.

3.1 Risk-Aware Task Weighting

Let $\mathcal{D}_{\text{ft}}$ denote the downstream fine-tuning set of pairs (x_i, y_i), where x_i is the i-th prompt and y_i its target response. Let $\mathcal{D}_{\text{dir}}$ denote the reference set of harmful prompts x_r, where x_r is the r-th prompt in the set. For a prompt x, $\boldsymbol{h}_{\text{B}}^{l}(x)$ and $\boldsymbol{h}_{\text{A}}^{l}(x)$ denote its last-prompt-token representations at layer l in the Base and Aligned Models, respectively, where L is the number of Transformer layers and $l = 1, \ldots, L$. The alignment-induced representation change is $\Delta\boldsymbol{h}^{l}(x) = \boldsymbol{h}_{\text{A}}^{l}(x) - \boldsymbol{h}_{\text{B}}^{l}(x)$. Before fine-tuning, we sum these changes over the reference set and normalize the result to obtain the layer-wise reference direction $\boldsymbol{v}^{l}$:

$$\boldsymbol{v}^{l} = \frac{\sum_{x_r \in \mathcal{D}_{\text{dir}}} \Delta\boldsymbol{h}^{l}(x_r)}{\left\| \sum_{x_r \in \mathcal{D}_{\text{dir}}} \Delta\boldsymbol{h}^{l}(x_r) \right\|_2}. \tag{1}$$

For downstream example i, let $\mathcal{V}_i$ contain layers with non-negligible alignment-induced changes. For nonempty $\mathcal{V}_i$, its directional consistency p_i and sample weight w_i are

$$p_i = \frac{1}{|\mathcal{V}_i|} \sum_{l \in \mathcal{V}_i} \cos\left(\Delta\boldsymbol{h}^{l}(x_i), \boldsymbol{v}^{l}\right), \qquad w_i = \max(0, p_i), \tag{2}$$

where $|\mathcal{V}_i|$ is the number of valid layers and $\cos(\cdot, \cdot)$ denotes cosine similarity. Validity criteria and numerical safeguards are given in Appendix B.1.

At step t, let $\theta^{(t)}$ denote the model parameters before the update and $\mathcal{B}^{(t)} \subset \mathcal{D}_{\text{ft}}$ the downstream mini-batch. For example i, let N_i denote the number of valid supervision tokens and $\ell_{\text{SFT}}(\theta^{(t)}; x_i, y_i)$ denote their summed autoregressive cross-entropy. The task objective is

$$\mathcal{L}_{\text{task}}^{(t)} = \frac{\sum_{(x_i, y_i) \in \mathcal{B}^{(t)}} (1 - w_i)\, \ell_{\text{SFT}}\left(\theta^{(t)}; x_i, y_i\right)}{\sum_{(x_i, y_i) \in \mathcal{B}^{(t)}} N_i}. \tag{3}$$

Through the factor $1 - w_i$, examples with higher directional consistency receive weaker supervision.

3.2 Base-Directed Representation Protection

Representation protection complements task weighting by penalizing changes opposing the reference directions. For example i, the reference displacement is $a_i^l = \langle \Delta\boldsymbol{h}^{l}(x_i), \boldsymbol{v}^{l} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. Let $\boldsymbol{h}_{\theta^{(t)}}^{l}(x_i)$ be its last-prompt-token representation in the current model and $\epsilon > 0$ a stability constant. For $a_i^l > \epsilon$, the relative Base-directed change is

$$e_i^{l,(t)} = \left[\frac{\left\langle \boldsymbol{h}_{\text{A}}^{l}(x_i) - \boldsymbol{h}_{\theta^{(t)}}^{l}(x_i), \boldsymbol{v}^{l} \right\rangle}{a_i^l} \right]_+, \tag{4}$$

where $[z]_+ = \max(0, z)$ for a scalar z. We set $e_i^{l,(t)} = 0$ when $a_i^l \le \epsilon$. This measures the opposing change relative to the reference displacement without penalizing changes in every direction.

The consistency-weighted layer-wise score $c^{l,(t)}$ is

$$c^{l,(t)} = \frac{\sum_{(x_i, y_i) \in \mathcal{B}^{(t)}} w_i \left(e_i^{l,(t)}\right)^2}{\sum_{(x_i, y_i) \in \mathcal{B}^{(t)}} w_i \mathbb{I}\left[a_i^l > \epsilon\right] + \epsilon}. \tag{5}$$

Here, $\mathbb{I}[\cdot]$ equals one when its condition holds and zero otherwise. Thus, w_i guides both task weighting and representation protection.

Adaptive Layer Protection uses the preceding-step layer-wise scores $c^{l,(t-1)}$ to adapt protection across layers, assigning larger weights to layers with higher scores. For $t \ge 2$, the layer weights $g^{l,(t)}$ and representation objective are

$$g^{l,(t)} = \frac{c^{l,(t-1)}}{\sum_{k=1}^{L} c^{k,(t-1)} + \epsilon}, \qquad \mathcal{L}_{\text{rep}}^{(t)} = \sum_{l=1}^{L} g^{l,(t)} c^{l,(t)}. \tag{6}$$

At the first update, no preceding-step scores are available, so the current layer-wise scores $c^{l,(1)}$ are used to initialize $g^{l,(1)}$. Uniform weights are used if all scores are zero. Further implementation details are provided in Appendix B.3.

3.3 Refusal-Anchor Supervision

To complement representation protection with direct refusal supervision, we use an anchor set $\mathcal{D}_{\text{anchor}}$ of harmful prompt–refusal pairs (x_j, y_j), where x_j is the j-th harmful prompt and y_j its corresponding safe refusal response. Let $\mathcal{B}^{(t)}_{\text{anchor}} \subset \mathcal{D}_{\text{anchor}}$ denote the sampled anchor mini-batch, and let N_j be the number of valid supervision tokens for pair (x_j, y_j). Using the same summed SFT loss, the anchor objective is

$$\mathcal{L}^{(t)}_{\text{anchor}} = \frac{\sum_{(x_j,y_j)\in\mathcal{B}^{(t)}_{\text{anchor}}} \ell_{\text{SFT}}\left(\theta^{(t)}; x_j, y_j\right)}{\sum_{(x_j,y_j)\in\mathcal{B}^{(t)}_{\text{anchor}}} N_j}. \tag{7}$$

Unlike task supervision, this objective does not use the sample weights w_i.

The complete objective combines the three losses with coefficients λ_{rep} and λ_{anchor}, which weight representation protection and anchor supervision, respectively:

$$\mathcal{L}^{(t)}_{\text{AIRP}} = \mathcal{L}^{(t)}_{\text{task}} + \lambda_{\text{rep}}\mathcal{L}^{(t)}_{\text{rep}} + \lambda_{\text{anchor}}\mathcal{L}^{(t)}_{\text{anchor}}. \tag{8}$$

The frozen Aligned Model and the precomputed $\boldsymbol{v}^l$, w_i, and a_i^l remain fixed throughout training.

4 Experiments

4.1 Experimental Setup

Datasets. For safety alignment, we sample 5,000 refusal-style examples from the BeaverTails-derived alignment data used in RepNoise (Rosati et al., 2024; Ji et al., 2023). We construct two held-out BeaverTails subsets: a reference set $\mathcal{D}_{\text{dir}}$ containing 500 harmful prompts for constructing the layer-wise reference directions, and an anchor set $\mathcal{D}_{\text{anchor}}$ containing 1,000 harmful prompt–refusal response pairs for Refusal-Anchor Supervision. These sets are disjoint from the safety-alignment and downstream fine-tuning data. For downstream fine-tuning, we use three datasets covering different task types: AGNEWS (Zhang et al., 2015) for news classification, GSM8K (Cobbe et al., 2021) for grade-school mathematical reasoning, and SST2 (Socher et al., 2013) for sentiment classification. Following prior harmful fine-tuning settings (Huang et al., 2024b; 2025b), each fine-tuning set contains n examples, of which a fraction p consists of harmful prompt–compliant response pairs sampled from BeaverTails. The remaining fraction $1 - p$ is drawn from the benign task.

Models and baselines. We conduct experiments on Llama-3-8B (Grattafiori et al., 2024), Qwen3-8B (Yang et al., 2025), and Gemma-2-2B (Gemma Team et al., 2024). We compare AIRP with standard supervised fine-tuning (SFT) and existing defense methods, including Lisa (Huang et al., 2024a), Vaccine (Huang et al., 2024b), SafeLoRA (Hsu et al., 2024), SaLoRA (Li et al., 2025b), SBR (Lu et al., 2026), and AsFT (Yang et al., 2026). In the tables, Pre-FT, In-FT, and Post-FT denote defenses applied before, during, and after downstream fine-tuning, respectively. Vaccine is categorized as Pre-FT, SafeLoRA as Post-FT, and Lisa, SaLoRA, SBR, AsFT, and AIRP as In-FT.

Metrics and evaluation. Following prior work (Huang et al., 2024b;a), we report Fine-tuning Accuracy (FA) for downstream utility and Harmful Score (HS) for safety. Higher FA and lower HS indicate better performance. We evaluate FA on the corresponding task evaluation split, using 872 examples for SST2 and 1,000 examples each for AGNEWS and GSM8K. HS is the percentage of responses classified as unsafe by the BeaverTails moderation model (Ji et al., 2023). We compute HS on 1,000 unseen harmful instructions from the BeaverTails test set.

Training details. We use LoRA (Hu et al., 2022) for both safety alignment and downstream fine-tuning. Unless a method requires its original method-specific configuration, we set the LoRA rank to $r = 32$ and the scaling factor to $\alpha = 4$. We use AdamW (Loshchilov & Hutter, 2019) with a weight decay of 0.1. The learning rates are 5×10^{-4} for safety alignment and 2×10^{-5} for downstream fine-tuning. Under the common configuration, both stages are trained for 20 epochs with a batch

Table 1: Performance across downstream tasks using Llama-3-8B with $n = 1{,}000$ and $p = 0.1$.

Methods			AGNEWS		GSM8K		SST2		Average	
			HS↓	FA↑	HS↓	FA↑	HS↓	FA↑	HS↓	FA↑
–	SFT	–	71.80	84.30	49.50	42.70	70.20	93.92	63.83	73.64
Pre-FT	Vaccine	NeurIPS '24	50.70	80.10	20.90	31.10	51.90	90.02	41.17	67.07
Post-FT	SafeLoRA	NeurIPS '24	6.30	76.30	16.50	43.10	4.60	80.50	9.13	66.63
In-FT	Lisa	NeurIPS '24	24.60	84.40	18.70	37.80	53.60	93.46	32.30	71.89
	SaLoRA	ICLR '25	28.60	83.50	27.40	40.40	29.20	92.20	28.40	72.03
	SBR	ICML '26	6.30	83.30	4.30	38.00	8.80	91.97	6.47	71.09
	AsFT	AAAI '26	8.30	**87.60**	15.20	40.00	13.10	**94.84**	12.20	74.15
	AIRP (Ours)	–	**4.30**	86.10	**1.60**	**45.70**	**3.10**	91.06	**3.00**	**74.29**

Table 2: Performance across model architectures on AGNEWS with $n = 1{,}000$ and $p = 0.1$.

Methods			Llama-3-8B		Qwen3-8B		Gemma-2-2B		Average	
			HS↓	FA↑	HS↓	FA↑	HS↓	FA↑	HS↓	FA↑
–	SFT	–	71.80	84.30	50.70	84.00	68.60	83.10	63.70	83.80
Pre-FT	Vaccine	NeurIPS '24	50.70	80.10	47.70	**85.40**	60.70	82.60	53.03	82.70
Post-FT	SafeLoRA	NeurIPS '24	6.30	76.30	16.20	80.20	24.50	59.90	15.67	72.13
In-FT	Lisa	NeurIPS '24	24.60	84.40	11.80	83.50	54.00	81.20	30.13	83.03
	SaLoRA	ICLR '25	28.60	83.50	10.40	81.10	15.30	**86.00**	18.10	83.53
	SBR	ICML '26	6.30	83.30	4.80	82.70	4.30	79.70	5.13	81.90
	AsFT	AAAI '26	8.30	**87.60**	4.20	81.40	7.60	84.10	6.70	84.37
	AIRP (Ours)	–	**4.30**	86.10	**4.00**	83.80	**3.90**	83.70	**4.07**	**84.53**

size of 10. Unless otherwise specified, the default setting uses Llama-3-8B on AGNEWS with $n = 1{,}000$ fine-tuning examples and a harmful-data ratio of $p = 0.1$. All experiments are conducted on four NVIDIA GeForce RTX 3090 GPUs. Additional implementation details and method-specific configurations are provided in Appendix A.

4.2 Main Experiments

Generalization across fine-tuning tasks. Table 1 compares the methods on AGNEWS, GSM8K, and SST2 using Llama-3-8B under the same harmful fine-tuning setting. AIRP achieves the lowest HS on all three tasks, yielding an average HS of 3.00%, compared with 6.47% for SBR, the lowest among the baselines. AIRP also obtains the highest average FA of 74.29%, slightly exceeding the 74.15% achieved by AsFT. On GSM8K, AIRP achieves the highest FA of 45.70%. On AGNEWS and SST2, its FA values of 86.10% and 91.06% are 1.50 and 3.78 percentage points below the corresponding best results from AsFT, respectively. Overall, AIRP consistently achieves the lowest HS across the three tasks while maintaining competitive downstream performance.

Generalization across model architectures. Table 2 evaluates AIRP across three model architectures on AGNEWS. AIRP achieves the lowest HS on all three architectures, with an average HS of 4.07%, compared with 5.13% for SBR, the lowest among the baselines. AIRP also obtains the highest average FA of 84.53%, slightly exceeding the 84.37% achieved by AsFT. Although the highest FA on each individual architecture is achieved by different baselines, AIRP maintains FA between 83.70% and 86.10% while achieving the lowest HS on all three architectures.

Robustness to harmful-data ratio. Table 3 evaluates the robustness of AIRP across different harmful-data ratios with $n = 1{,}000$. HS tends to increase as p grows, while AIRP consistently achieves the lowest HS at every non-zero ratio. At $p = 0.3$, AIRP obtains an HS of 14.30%, slightly below the 14.40% of SBR and substantially below the 73.80% of SFT. Meanwhile, AIRP retains an FA of 85.00%, compared with 77.80% for SBR, and its FA remains between 85.00% and 86.90%

Table 3: Performance under different harmful-data ratios using 1,000 fine-tuning samples.

Methods		Harmful Score↓					Fine-tuning Accuracy↑				
		$p=0$	$p=0.05$	$p=0.1$	$p=0.2$	$p=0.3$	$p=0$	$p=0.05$	$p=0.1$	$p=0.2$	$p=0.3$
–	SFT	1.30	55.00	71.80	73.30	73.80	84.20	84.00	84.30	**87.60**	85.70
Pre-FT	Vaccine	1.10	34.10	50.70	70.20	71.50	81.70	80.50	80.10	84.20	80.80
Post-FT	SafeLoRA	**0.90**	6.80	6.30	10.40	15.60	75.10	80.00	76.30	76.20	50.70
In-FT	Lisa	1.20	31.50	24.60	62.80	66.70	85.30	84.30	84.40	86.00	84.60
	SaLoRA	1.40	16.60	28.60	54.90	67.30	**87.60**	83.10	83.50	85.60	86.50
	SBR	1.00	4.80	6.30	11.50	14.40	83.70	82.50	83.30	81.70	77.80
	AsFT	1.40	4.10	8.30	19.10	35.90	87.10	**86.90**	**87.60**	87.40	**87.00**
	AIRP (Ours)	1.00	**3.20**	**4.30**	**9.00**	**14.30**	85.10	**86.90**	86.10	85.30	85.00

Table 4: Performance across fine-tuning set sizes at a harmful-data ratio of $p = 0.1$.

Methods		Harmful Score↓					Fine-tuning Accuracy↑				
Fine-tuning set size, n		500	1000	1500	2000	2500	500	1000	1500	2000	2500
–	SFT	46.60	71.80	71.80	71.50	72.60	**52.80**	84.30	84.80	87.50	89.30
Pre-FT	Vaccine	25.60	50.70	64.90	72.80	75.50	44.60	80.10	84.00	85.60	87.80
Post-FT	SafeLoRA	8.30	6.30	12.50	20.90	23.30	49.90	76.30	80.70	82.70	81.60
In-FT	Lisa	27.00	24.60	58.20	66.50	68.90	50.50	84.40	85.10	87.40	88.70
	SaLoRA	15.40	28.60	58.60	64.10	71.70	25.30	83.50	84.60	**88.80**	88.80
	SBR	**3.20**	6.30	13.80	25.00	31.70	45.70	83.30	85.00	87.20	88.20
	AsFT	**3.20**	8.30	14.20	21.10	32.00	52.50	**87.60**	**88.10**	88.40	89.00
	AIRP (Ours)	**3.20**	**4.30**	**6.90**	**8.50**	**8.70**	42.30	86.10	87.00	88.70	**90.00**

across all evaluated ratios. These results show that AIRP limits safety degradation as the harmful-data ratio increases while maintaining stable downstream task performance.

Robustness to fine-tuning set size. Table 4 evaluates the robustness of AIRP across different fine-tuning set sizes with $p = 0.1$. The HS of most baselines tends to increase as n grows, whereas AIRP ties for the lowest HS at $n = 500$ and achieves the lowest HS for $n \geq 1{,}000$. Its HS remains below 9.0% for $n \geq 1{,}000$. At $n = 2{,}500$, AIRP achieves an HS of 8.70%, which is 14.60 percentage points below SafeLoRA, the next-best baseline, while also obtaining the highest FA of 90.00%. Additional robustness experiments on reference-set size, fine-tuning strength, and out-of-distribution harmful data are provided in Appendix C.

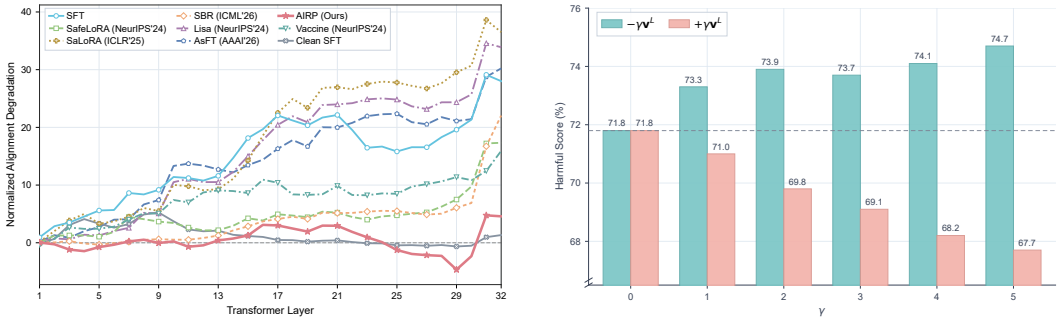
4.3 Representation Analysis

Figure 2a shows that higher directional consistency is associated with greater subsequent safety degradation. We compare layer-wise representation changes across defense methods and assess the safety relevance of the final-layer reference direction through controlled intervention.

Layer-Wise Alignment Degradation. Layer-wise representation changes after downstream fine-tuning are characterized using *Normalized Alignment Degradation* (NAD), which measures the Base-directed representation change along each layer's reference direction, normalized by the Aligned representation scale. For each evaluation prompt x, the NAD at layer l is defined as

$$\mathrm{NAD}^l(x) = \frac{\left\langle \boldsymbol{h}_{\mathrm{A}}^l(x) - \boldsymbol{h}_{\theta}^l(x), \boldsymbol{v}^l \right\rangle}{\left\| \boldsymbol{h}_{\mathrm{A}}^l(x) \right\|_2 / \sqrt{H}}. \tag{9}$$

Here, $\boldsymbol{h}_{\mathrm{A}}^l(x)$ and $\boldsymbol{h}_{\theta}^l(x)$ are the last-prompt-token representations of the Aligned Model and the final fine-tuned model, respectively, H is the hidden-state dimension, and $\boldsymbol{v}^l$ is the layer-wise reference direction. Positive NAD indicates movement against the reference direction. Larger values indicate



(a) Layer-wise Normalized Alignment Degradation (NAD) across defense methods.

(b) Harmful Score under steering along or against the final-layer reference direction.

Figure 3: Representation-level analysis on Llama-3-8B under the default AGNEWS setting. (a) Layer-wise Normalized Alignment Degradation after fine-tuning. (b) Harmful Score under steering of the last-prompt-token representation along or against the final-layer reference direction.

a larger normalized change in that direction. NAD is computed on 1,000 evaluation prompts disjoint from the training data and averaged over valid samples at each layer.

As shown in Figure 3a, NAD under Clean SFT remains near zero across most layers. In contrast, SFT, Lisa, SaLoRA, and AsFT exhibit pronounced positive NAD in the middle and later layers, while SafeLoRA, SBR, and Vaccine show lower degradation but increasing NAD toward the final layers. AIRP remains closest to the Clean SFT reference across most layers and exhibits substantially lower NAD than the other defenses in the middle and later layers, indicating smaller Base-directed representation changes. Overall, the degradation is highly non-uniform across layers, with the largest deviations concentrated in the middle and later layers. The pronounced layer-wise variation in alignment degradation further motivates Adaptive Layer Protection, whose incremental contribution is evaluated in Section 4.4.

Steering along the Final-Layer Reference Direction. To assess the behavioral relevance of the final-layer reference direction, we perform a controlled representation intervention on the Llama-3-8B SFT model, following prior work on activation steering and refusal-mediating directions (Arditi et al., 2024; Rimsky et al., 2024). The last-prompt-token representation at the final Transformer block is modified as $\boldsymbol{h}^L \leftarrow \boldsymbol{h}^L \pm \gamma \boldsymbol{v}^L$, where $\boldsymbol{v}^L$ is the final-layer reference direction and $\gamma \geq 0$ controls the intervention strength. Adding $\gamma \boldsymbol{v}^L$ steers the representation along the reference direction, whereas subtracting it induces a Base-directed change. The unmodified model corresponds to $\gamma = 0$ and has an HS of 71.80%. As shown in Figure 3b, adding $\gamma \boldsymbol{v}^L$ reduces HS to 67.70% at $\gamma = 5$, whereas subtracting it increases HS to 74.70%. The opposing behavioral effects support the safety relevance of the final-layer reference direction within the evaluated model and intervention setting. Additional direction-specific controls are provided in Appendix D.2 to further examine the role of the reference directions and their layer-wise correspondence.

4.4 Ablation Study

Effectiveness of the proposed components. Table 5 presents a cumulative ablation of AIRP under the default setting. Each row adds one design element to the preceding configuration. Without Adaptive Layer Protection, the layer-wise representation-protection terms are aggregated uniformly. Adaptive Layer Protection instead adjusts their contributions according to the observed Base-directed representation changes.

Risk-Aware Task Weighting reduces HS from 71.80% to 39.20%, with a slight decrease in FA from 84.30% to 83.60%. Adding Base-Directed Representation Protection further reduces HS to 17.20% while increasing FA to 85.60%. Adaptive Layer Protection further lowers HS to 12.70% and raises FA to 86.10%, without introducing a separate loss. Refusal-Anchor Supervision then reduces HS to 4.30% while retaining an FA of 86.10%. Thus, every incremental addition reduces HS under this cumulative ordering, and the complete AIRP configuration achieves the lowest HS while tying for the highest FA. A more comprehensive ablation analysis is provided in Appendix E.

Table 5: Cumulative ablation of AIRP on Llama-3-8B under the default setting.

Methods	AGNEWS	
	HS↓	FA↑
SFT	71.80	84.30
+ Risk-Aware Task Weighting	39.20	83.60
+ Base-Directed Representation Protection	17.20	85.60
+ Adaptive Layer Protection	12.70	**86.10**
+ Refusal-Anchor Supervision (AIRP)	**4.30**	**86.10**

Table 6: Sensitivity of AIRP to the representation-protection and refusal-anchor coefficients on AGNEWS. Gray rows indicate the default values used in the main experiments.

(a) Varying λ_{rep} with $\lambda_{\text{anchor}} = 0.1$.

λ_{rep}	HS↓	FA↑
0.1	7.40	84.50
0.3	5.80	85.00
0.5	4.70	85.50
1.0	4.30	86.10
2.0	4.00	86.40
5.0	3.40	85.30

(b) Varying λ_{anchor} with $\lambda_{\text{rep}} = 1.0$.

λ_{anchor}	HS↓	FA↑
0.03	7.90	86.20
0.05	7.40	86.00
0.1	4.30	86.10
0.2	2.60	85.30
0.3	2.40	84.90
0.5	2.30	85.00

4.5 Hyper-parameter Analysis

We evaluate the sensitivity of AIRP to the representation-protection coefficient λ_{rep} and the refusal-anchor coefficient λ_{anchor} on Llama-3-8B with AGNEWS under the default setting. When varying λ_{rep}, $\lambda_{\text{anchor}} = 0.1$. When varying λ_{anchor}, $\lambda_{\text{rep}} = 1.0$. All other settings remain unchanged.

As shown in Table 6, AIRP maintains a favorable safety–utility profile across the evaluated coefficient ranges, without relying on a narrowly tuned parameter choice. Increasing λ_{rep} from 0.1 to 5.0 reduces HS from 7.40% to 3.40%, while FA remains between 84.50% and 86.40%. Similarly, increasing λ_{anchor} from 0.03 to 0.5 reduces HS from 7.90% to 2.30%, with FA remaining between 84.90% and 86.20%. These trends indicate that the two coefficients meaningfully regulate the safety–utility trade-off, while performance remains stable across a broad range of values. The setting $\lambda_{\text{rep}} = 1.0$ and $\lambda_{\text{anchor}} = 0.1$ achieves an HS of 4.30% and an FA of 86.10%.

5 Conclusion

In this work, we study safety preservation during downstream fine-tuning in the presence of harmful training data. Our analyses show that harmful-example groups with higher directional consistency yield higher Harmful Scores after fine-tuning. We further find that harmful fine-tuning induces non-uniform Base-directed representation changes across model layers, with larger changes concentrated mainly in the middle and later layers. Motivated by these findings, we propose Alignment-Induced Representation Protection (AIRP), which regulates downstream adaptation relative to layer-wise reference directions to preserve alignment-induced representation changes and refusal behavior. Across downstream tasks, model architectures, harmful-data ratios, and fine-tuning set sizes, AIRP improves safety retention while maintaining competitive downstream performance. Future work could extend AIRP to full-parameter fine-tuning and investigate its effectiveness under different safety-alignment procedures.

AI Use Statement

Generative AI tools were used to assist with language polishing, clarity improvement, and presentation refinement during manuscript preparation. They were also used for limited assistance with

code debugging and figure formatting. All technical content, experimental designs, implementations, results, analyses, and citations were reviewed and verified by the authors. The authors take full responsibility for the content of the paper.

Ethics Statement

This work studies defenses against harmful fine-tuning of large language models. The experiments involve harmful prompts and prompt–response pairs solely for controlled safety evaluation and defense development. We use established research datasets and do not collect new personal data or involve human participants. Harmful content is used only within the experimental pipeline and is not intended to facilitate harmful model behavior.

Reproducibility Statement

We provide the methodological definitions, experimental settings, hyperparameters, datasets, and evaluation protocols required to reproduce the main experiments. Additional implementation and experimental details are provided in the appendix. A GitHub repository with the implementation and supporting materials is available upon request.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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 *The Twelfth International Conference on Learning Representations*, 2024.

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Vulnerability-aware alignment: Mitigating uneven forgetting in harmful fine-tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8172–8183. PMLR, 2025.

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

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. Association for Computational Linguistics, 2026.

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

Gemma Team, Morgane Riviere, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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*, volume 37, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022.

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*, volume 38, 2025.

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

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*, volume 37, 2024b.

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

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 *The Thirteenth International Conference on Learning Representations*, 2025b.

Jiaming Ji, Mickel Liu, Juntao 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 Datasets and Benchmarks Track*, volume 36, 2023.

Hao Li, Lijun Li, Zhenghao Lu, Xianyi Wei, Rui Li, Jing Shao, and Lei Sha. Layer-aware representation filtering: Purifying finetuning data to preserve LLM safety alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8030–8050. Association for Computational Linguistics, 2025a.

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

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, 2025.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations*, 2019.

Guoxin Lu, Letian Sha, Qing Wang, Peijie Sun, Hao Zhou, Hua Dai, and Fu Xiao. Safety anchor: Defending harmful fine-tuning via geometric bottlenecks. In *Forty-third International Conference on Machine Learning*, 2026.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 40537–40559. PMLR, 2025.

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 *The Fourteenth International Conference on Learning Representations*, 2026.

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

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

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024.

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*, volume 37, 2024.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013.

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 *The Thirteenth International Conference on Learning Representations*, 2025.

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*, volume 38, 2025.

Yuhui Wang, Rongyi Zhu, and Ting Wang. Self-destructive language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

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. Association for Computational Linguistics, 2025.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34322–34330, 2026.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Ruth Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25706–25714, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

A Experimental Details

A.1 Datasets and Prompt Templates

We evaluate downstream adaptation on three commonly used benchmarks: AGNEWS (Zhang et al., 2015), GSM8K (Cobbe et al., 2021), and SST2 (Socher et al., 2013). Examples from each dataset are formatted as instruction-following prompts using the templates below.

AGNEWS is a four-class news classification dataset covering World, Sports, Business, and Sci/Tech. We use the following prompt template:

Prompt format for AGNEWS

Below is an instruction that describes a task, paired with an input that provides further context.
Write a response that appropriately completes the request.
Instruction:
Categorize the news article given in the input into one of the 4 categories:
World
Sports
Business
Sci/Tech
Input:
(from the AGNEWS dataset)

GSM8K is a dataset of grade-school mathematical word problems involving multi-step reasoning and arithmetic computation. We use the following prompt template:

Prompt format for GSM8K

Below is an instruction that describes a task, paired with an input that provides further context.
Write a response that appropriately completes the request.
Instruction:
(from the GSM8K dataset)

SST2 is a binary sentiment classification dataset with positive and negative labels. We use the following prompt template:

Prompt format for SST2

Below is an instruction that describes a task, paired with an input that provides further context.
Write a response that appropriately completes the request.
Instruction:
Analyze the sentiment of the input, and respond only positive or negative
Input:
(from the SST2 dataset)

Safety-related and evaluation data. We follow the data construction in RepNoise (Rosati et al., 2024), which derives safety-alignment and harmful data from BeaverTails (Ji et al., 2023). Four mutually disjoint subsets are used during alignment and downstream fine-tuning. The safety-alignment subset contains 5,000 harmful prompt–refusal response pairs and is used to obtain the Aligned Model from the Base Model. The reference set $\mathcal{D}_{\mathrm{dir}}$ contains 500 harmful prompts for constructing the layer-wise reference directions before downstream fine-tuning and does not participate in parameter updates. The anchor set $\mathcal{D}_{\mathrm{anchor}}$ contains 1,000 harmful prompt–refusal response pairs for Refusal-Anchor Supervision. The remaining subset provides the harmful prompt–compliant response pairs used as the harmful portion of $\mathcal{D}_{\mathrm{ft}}$. These four subsets do not overlap.

For safety evaluation, we separately sample 1,000 harmful prompts from the BeaverTails-30k-test split. These prompts are used exclusively for evaluation and are disjoint from all training-related subsets.

A.2 Training Details

We use LoRA (Hu et al., 2022) for parameter-efficient adaptation during both safety alignment and downstream fine-tuning. The LoRA rank is set to 32 and the scaling factor to $\alpha = 4$. Safety alignment is performed for 20 epochs using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 5×10^{-4}, weight decay of 0.1, and batch size 10. We use a constant learning-rate schedule without warmup. After alignment, a new Task LoRA is initialized on the Aligned Model, and only its parameters are updated during downstream fine-tuning.

Downstream fine-tuning is performed for 20 epochs using AdamW with a learning rate of 2×10^{-5} and weight decay of 0.1. The same constant learning-rate schedule is used. For AIRP, $(\lambda_{\text{rep}}, \lambda_{\text{anchor}})$ is set to $(1, 0.1)$ for AGNEWS, $(0.1, 0.1)$ for GSM8K, and $(1, 0.3)$ for SST2. These values remain fixed across the model architectures, harmful-data ratios, and fine-tuning set sizes evaluated for each task.

A.3 Baseline Implementations

Unless otherwise specified, all baselines follow the common training setup in Appendix A.2. The same safety-alignment data, downstream fine-tuning data, and LoRA-based adaptation pipeline are used whenever permitted by the original method. Method-specific operations and hyperparameters are described below.

SFT serves as the standard no-defense baseline. We first perform supervised safety alignment using an Alignment LoRA, which is then merged into the Base Model to obtain the Aligned Model. A new Task LoRA is initialized for downstream fine-tuning, and only its parameters are updated on the user-provided fine-tuning data.

Lisa (Huang et al., 2024a) is applied during downstream fine-tuning and alternates between alignment-preserving and task fine-tuning updates. We set $\rho = 0.01$ and use 10,000 guide examples. Each optimization cycle uses 100 alignment steps followed by 900 fine-tuning steps. The remaining settings follow the common downstream fine-tuning configuration.

Vaccine (Huang et al., 2024b) is applied during safety alignment by introducing adversarial perturbations to hidden representations. We set the perturbation intensity to $\rho = 2$. The resulting Alignment LoRA is then merged into the Base Model. Downstream adaptation uses a new Task LoRA under the same fine-tuning setup as SFT.

SafeLoRA (Hsu et al., 2024) is applied after downstream fine-tuning as a projection procedure. We first obtain the Task LoRA through standard fine-tuning and then apply SafeLoRA using the Alignment LoRA as the safety reference. We use layer selection based on the number of selected layers and set the projection ratio to 0.35. The projected adapter is used for evaluation.

SaLoRA (Li et al., 2025b) is applied during downstream fine-tuning using its fixed safety module and task-specific adapter initialization. We set the LoRA adapter rank to $r = 16$, the safety rank to $r_s = 32$, and the task-specific rank to $r_t = 32$. We use 128 downstream examples for task-specific initialization. SaLoRA is trained for one epoch with a batch size of 10, a learning rate of 1×10^{-4}, zero weight decay, and a constant learning-rate schedule.

SBR (Lu et al., 2026) is applied during downstream fine-tuning through representation anchoring on a fixed set of safety prompts. Before fine-tuning, we record the final-layer last-prompt-token representation for each safety prompt as a fixed reference. During fine-tuning, SBR penalizes the mean-squared deviation between the current and reference representations in addition to the task loss. We set the regularization coefficient to 50. All other settings follow the common downstream fine-tuning configuration.

AsFT (Yang et al., 2026) is applied during downstream fine-tuning. It constructs a parameter-space alignment direction from the weight difference between the Base and Aligned Models and regularizes the component of the Task LoRA update orthogonal to this direction. We set the LoRA rank to $r = 8$, the regularization coefficient to $\lambda = 1$, and the learning rate to 5×10^{-5}. AsFT is trained for 10 epochs, while the remaining settings follow the common downstream fine-tuning configuration.

A.4 Evaluation Protocol

We use deterministic greedy decoding for all evaluations. Prompt formats follow Appendix A.1. Downstream task performance is measured using task-specific scoring rules, while model safety is evaluated on held-out harmful prompts.

AGNEWS. We evaluate on the AGNEWS test set. A prediction is counted as correct when the generated response exactly matches the class label corresponding to the ground-truth category under the predefined label mapping. Fine-tuning Accuracy is the percentage of correctly classified examples.

SST2. We evaluate on the SST2 validation set. A prediction is counted as correct when the generated response matches the ground-truth sentiment label, i.e., *positive* or *negative*, allowing capitalization variants. Fine-tuning Accuracy is computed over all evaluation examples.

GSM8K. We evaluate on the GSM8K test set and score only the final numerical answer. If the generated response contains the phrase "The final answer is:", we extract the first numerical value following the phrase. Otherwise, we use the last numerical value in the response. A prediction is considered correct if the extracted value exactly matches the ground-truth final answer.

Safety evaluation. We evaluate safety on 1,000 harmful prompts from the BeaverTails-30k-test split. For each prompt, the fine-tuned model generates a response, which is evaluated together with the corresponding prompt using the Beaver-Dam-7B moderation model (Ji et al., 2023). Harmful Score is the percentage of generated responses classified as unsafe by the moderation model.

B Additional Method Details

B.1 Offline Precomputation

Offline precomputation takes place after safety alignment and before downstream fine-tuning. The reference set $\mathcal{D}_{\mathrm{dir}}$ supplies the harmful prompts for constructing the layer-wise reference directions. The downstream fine-tuning set $\mathcal{D}_{\mathrm{ft}}$ is used to compute the sample weights w_i and reference displacements a_i^l. Both computations use prompt-only inputs, excluding response tokens.

For each prompt, we extract the last-prompt-token representation from every Transformer block under the frozen Base and Aligned Models, excluding the embedding output. In the implementation, the alignment-induced changes are averaged over $\mathcal{D}_{\mathrm{dir}}$ before normalization at each layer. Without the numerical safeguard, normalizing this mean is equivalent to normalizing the sum in Equation 1. The Euclidean norm in the normalization denominator is lower-bounded by 10^{-8}.

For downstream example i, directional consistency is averaged only over layers with non-negligible alignment-induced changes. The valid-layer set in Equation 2 is

$$\mathcal{V}_i = \left\{ l \in \{1, \ldots, L\} : \left\| \Delta \boldsymbol{h}^l(x_i) \right\|_2 \geq 10^{-12} \right\}, \tag{10}$$

where $\| \cdot \|_2$ denotes the Euclidean norm. If $\mathcal{V}_i$ is empty, we set $p_i = w_i = 0$. Otherwise, the consistency score and sample weight follow Equation 2.

An offline pass over $\mathcal{D}_{\mathrm{ft}}$ computes w_i and a_i^l as defined in Sections 3.1 and 3.2. These quantities and the directions $\boldsymbol{v}^l$ remain fixed throughout downstream fine-tuning. The Base Model is no longer required after precomputation.

B.2 Token-Level Loss Computation

For task and anchor supervision, logits and labels are shifted for autoregressive prediction. Positions labeled -100, including padding positions, are excluded from the loss. The valid-token counts N_i and N_j count the remaining positions after shifting for downstream and anchor examples, respectively. The SFT loss ℓ_{SFT} sums the cross-entropy over these positions.

For Equation 3, every valid-token loss of downstream example i is multiplied by $1 - w_i$. The weighted losses are summed across the mini-batch and divided by its unweighted valid-token count. Thus, sample weights affect only the numerator, and losses are not averaged separately within each example.

The anchor objective in Equation 7 uses the same token masking and batch-level normalization, without sample weighting. For both objectives, the token-count denominator is lower-bounded by one. A batch with no valid supervision tokens therefore contributes zero SFT loss.

B.3 Training Procedure and Adaptive Layer Protection

Downstream fine-tuning starts from the Aligned Model with a newly initialized Task LoRA. A separate frozen Aligned Model provides the reference representations. At each step, downstream prompt–response pairs are used to compute the task loss. Their prompt portions are also passed through the current model and the frozen Aligned Model to compute $e_i^{l,(t)}$ and $c^{l,(t)}$ according to Equations 4 and 5.

Adaptive Layer Protection uses the most recently retained layer-wise scores when their sum is positive. When these scores come from the immediately preceding step, the weights follow Equation 6. If no preceding scores are available, or their sum is zero, the current scores $c^{l,(t)}$ are used instead. They are divided by their sum plus ϵ, following the same normalization rule. If the current scores also sum to zero, uniform weights $g^{l,(t)} = 1/L$ are used. This rule includes initialization at $t = 1$.

The layer weights are detached from gradient computation. The representation loss therefore differentiates through the current scores $c^{l,(t)}$, but not through $g^{l,(t)}$. After weight computation, the current scores are detached and retained for the next step, replacing the previous scores. Gradient accumulation is set to one, so this update occurs at each optimizer step. If a mini-batch contains no sample-layer pair satisfying both $w_i > 0$ and $a_i^l > \epsilon$, its representation loss is zero and the previously retained scores remain unchanged.

For Refusal-Anchor Supervision, the anchor set $\mathcal{D}_{\text{anchor}}$ contains 1,000 harmful prompt–refusal response pairs (Appendix A.1). In our experiments, one pair is sampled uniformly at random at each step to form $\mathcal{B}_{\text{anchor}}^{(t)}$ and compute the anchor loss.

The three losses are combined according to Equation 8. Only the Task LoRA parameters are updated. The frozen Aligned Model and the offline reference quantities remain unchanged.

C Additional Robustness Experiments

C.1 Sensitivity to Reference-Set Size

We evaluate the sensitivity of AIRP to the number of harmful prompts used to construct the layer-wise reference directions. Specifically, we vary the size of $\mathcal{D}_{\text{dir}}$ over $\{100, 200, 300, 500, 700\}$ while keeping all other experimental settings fixed.

Figure 4 shows that AIRP remains stable across the evaluated reference-set sizes. HS varies only between 3.70% and 4.40%, while FA remains within 85.30%–86.10%. No monotonic improvement is observed as the reference set becomes larger. Even with only 100 reference prompts, AIRP achieves an HS of 3.70% and an FA of 85.70%. These results suggest that the constructed layer-wise reference directions are not highly sensitive to the reference-set size within the evaluated range.

The reference size of 500 used in the main experiments falls within this stable regime. It achieves the highest FA among the evaluated settings, at 86.10%, while maintaining a low HS of 4.30%. This setting therefore provides a reasonable trade-off between downstream performance and the amount of reference data required to construct the layer-wise reference directions.

C.2 Robustness to Fine-Tuning Strength

Robustness to downstream fine-tuning strength is evaluated along two optimization dimensions: the number of fine-tuning epochs and the learning rate. Experiments use Llama-3-8B on AGNEWS and compare AIRP with SFT and SBR. When varying the number of epochs, the learning rate is fixed at 2×10^{-5}. When varying the learning rate, the number of epochs is fixed at 20. All other experimental settings remain unchanged.

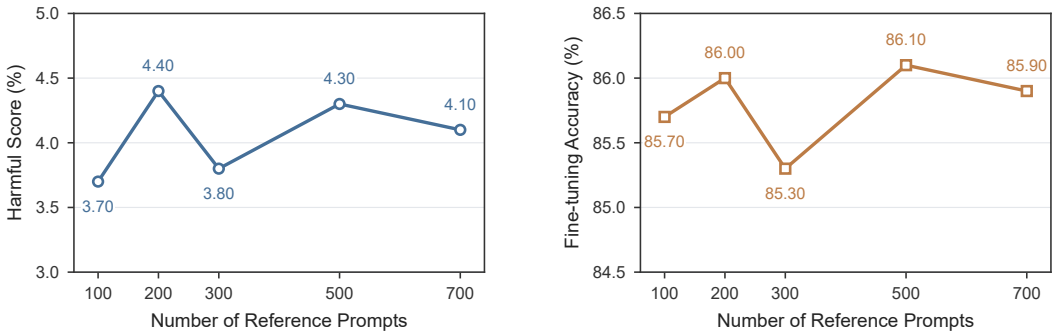


Figure 4: Sensitivity of AIRP to the number of harmful reference prompts used to construct the layer-wise reference directions. The left panel reports Harmful Score and the right panel reports Fine-tuning Accuracy.

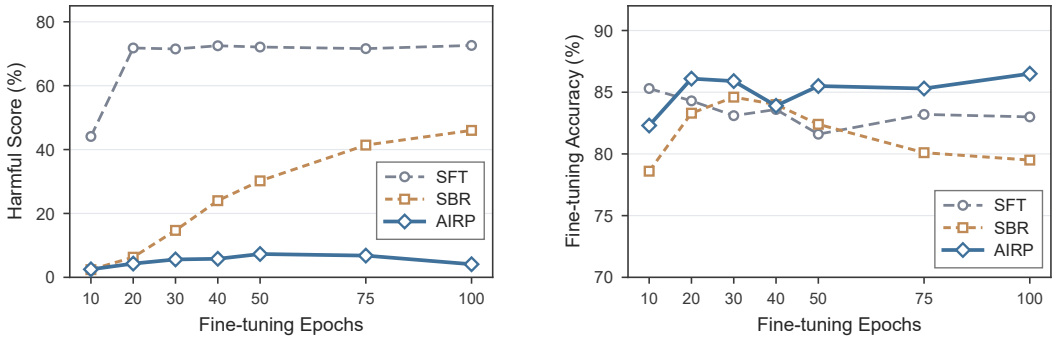


Figure 5: Performance under different numbers of downstream fine-tuning epochs on AGNEWS. The left panel reports Harmful Score and the right panel reports Fine-tuning Accuracy for SFT, SBR, and AIRP.

Figure 5 shows the effect of increasing the number of fine-tuning epochs. The HS of SFT rises from 44.10% at 10 epochs to above 70% at all subsequent settings. SBR maintains low HS under shorter fine-tuning but degrades as training continues, reaching 46.00% at 100 epochs. In contrast, AIRP maintains HS between 2.50% and 7.30% throughout the evaluated range, while FA remains between 82.30% and 86.50%. At 100 epochs, AIRP achieves an HS of 4.10% and an FA of 86.50%, compared with 46.00% HS and 79.50% FA for SBR.

Figure 6 evaluates learning rates from 5×10^{-6} to 1×10^{-3}, spanning a 200-fold range. SBR maintains low HS at smaller learning rates but degrades sharply at 5×10^{-5}, where its HS reaches 35.10%, and exceeds 45% at several larger values. AIRP maintains HS below 6% across all evaluated learning rates. Its FA generally increases with the learning rate, reaching 90.10% at 1×10^{-3} while retaining an HS of 4.30%. Overall, AIRP maintains stable safety and competitive task performance under substantial changes in downstream optimization strength.

C.3 Robustness to Out-of-Distribution Harmful Data

Robustness to harmful-data distribution shift is evaluated using two out-of-distribution harmful datasets: Harmful HEx-PHI (Qi et al., 2025) and PureBad (Qi et al., 2024). The reference set $\mathcal{D}_{\text{dir}}$ and all other experimental settings remain unchanged. Only the harmful portion of the downstream fine-tuning data is replaced with examples from the corresponding OOD dataset. We compare AIRP with SFT and SBR and track Harmful Score throughout fine-tuning.

As shown in Figure 7, both OOD datasets cause substantial safety degradation under standard fine-tuning. On Harmful HEx-PHI, the HS of SFT increases from 1.10% after one epoch to 70.60% after 20 epochs. On PureBad, HS similarly increases from 0.90% to 76.50%. This confirms that both

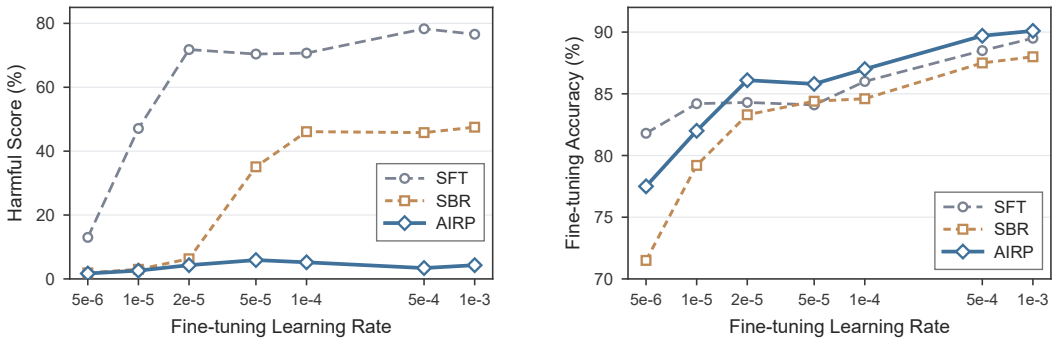


Figure 6: Performance under different downstream fine-tuning learning rates on AGNEWS. The left panel reports Harmful Score and the right panel reports Fine-tuning Accuracy for SFT, SBR, and AIRP. The learning-rate axis uses a logarithmic scale.

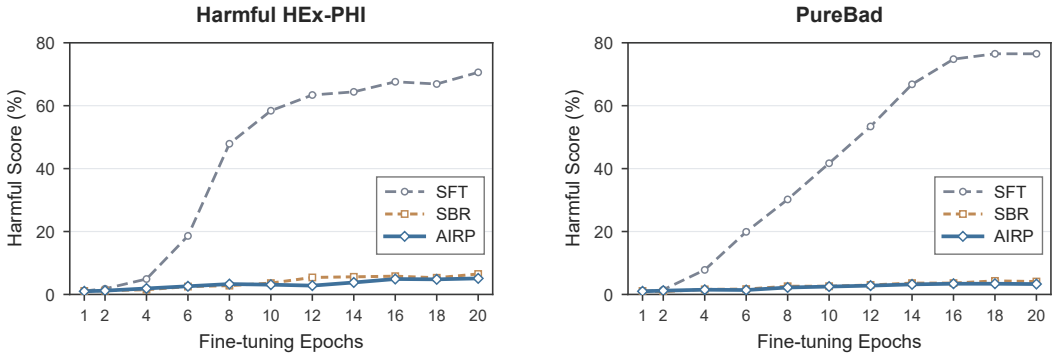


Figure 7: Harmful Score across downstream fine-tuning epochs when the harmful examples are drawn from Harmful HEx-PHI (Qi et al., 2025) (left) and PureBad (Qi et al., 2024) (right). Results are reported for SFT, SBR, and AIRP.

datasets remain effective sources of harmful fine-tuning despite differing from the prompts used to construct the layer-wise reference directions.

SBR and AIRP both mitigate this degradation, while AIRP maintains lower HS throughout most of the evaluated checkpoints. On Harmful HEx-PHI, AIRP keeps HS within 1.00%–5.10% and reaches 5.10% at 20 epochs, compared with 6.50% for SBR. On PureBad, AIRP remains below 3.40% throughout training and achieves an HS of 3.30% at 20 epochs, compared with 4.10% for SBR. Overall, the layer-wise reference directions constructed from the original reference set remain effective under the evaluated harmful-data distribution shifts, supporting the robustness of AIRP beyond the reference distribution.

D Additional Analysis of Reference Directions

D.1 Directional Consistency and Safety Outcomes

The analysis below expands on the directional-consistency results summarized in Figure 2a, providing the full task-wise results across the three consistency groups. We compute directional consistency p_i for 3,200 harmful examples sampled from BeaverTails (Ji et al., 2023). The examples are sorted by p_i and divided into three approximately equal-sized groups. The resulting low-, medium-, and high-consistency groups have mean p_i values of approximately 0.79, 0.84, and 0.87, respectively. We randomly sample 100 examples from each group and conduct separate fine-tuning runs under the same training settings.

Table 7: Fine-tuning performance across harmful-example groups with different directional consistency.

Sample Group	AGNEWS		GSM8K		SST2	
	HS↓	FA↑	HS↓	FA↑	HS↓	FA↑
Low	53.60	83.30	33.30	42.70	58.00	94.50
Medium	71.30	83.40	56.10	45.00	71.80	95.07
High	73.10	83.60	64.40	43.80	72.50	93.92

Table 7 shows that HS increases from the low- to high-consistency group on all three downstream tasks, while FA remains relatively stable across groups. The HS gaps between the high- and low-consistency groups are 19.50, 31.10, and 14.50 percentage points on AGNEWS, GSM8K, and SST2, respectively. These results provide the detailed evidence underlying the trend reported in the Introduction: fine-tuning on higher-consistency harmful-example groups yields higher Harmful Scores across the evaluated tasks.

D.2 Direction-Specific Control Experiments

To examine whether AIRP depends specifically on the precomputed layer-wise reference directions, we construct three types of control directions. All experiments use Llama-3-8B on AGNEWS with $n = 1000$ and $p = 0.1$, following the default training configuration. For each control setting, the original reference direction $\boldsymbol{v}^l$ is completely replaced by the corresponding control direction at every layer. Consequently, all direction-dependent quantities, including p_i, a_i^l, and $e_i^{l,(t)}$, are recomputed under the modified directions.

Random orthogonal directions. For each layer l, we sample a Gaussian random vector $\boldsymbol{r}^l$ and remove its component along the original reference direction:

$$\widetilde{\boldsymbol{v}}^l = \frac{\boldsymbol{r}^l - \langle \boldsymbol{r}^l, \boldsymbol{v}^l \rangle \boldsymbol{v}^l}{\|\boldsymbol{r}^l - \langle \boldsymbol{r}^l, \boldsymbol{v}^l \rangle \boldsymbol{v}^l\|_2}.$$

This produces a unit direction orthogonal to $\boldsymbol{v}^l$. We repeat the construction with three random seeds, 11, 22, and 33. This control tests whether an arbitrary directional constraint is sufficient to produce the safety effect observed with the original reference directions.

Benign-prompt directions. We construct alternative directions from benign prompts sampled from GSM8K and SST2. For each source, let $\mathcal{D}_{\text{benign}}$ denote the set of 500 sampled benign prompts. These prompts are processed using the same Base-to-Aligned representation-change procedure used for the original reference directions:

$$\widetilde{\boldsymbol{v}}^l = \frac{\sum_{x \in \mathcal{D}_{\text{benign}}} \Delta \boldsymbol{h}^l(x)}{\left\| \sum_{x \in \mathcal{D}_{\text{benign}}} \Delta \boldsymbol{h}^l(x) \right\|_2}.$$

This control preserves the direction-estimation procedure while changing the prompts used to define the directions.

Layer-shuffled directions. Finally, we preserve the original set of reference directions but break their correspondence with Transformer layers. For a fixed-point-free permutation π, the direction assigned to layer l becomes

$$\widetilde{\boldsymbol{v}}^l = \boldsymbol{v}^{\pi(l)}, \qquad \pi(l) \neq l.$$

We generate three such permutations using seeds 101, 202, and 303. This setting tests whether the layer-wise correspondence of the reference directions is important, rather than only the direction set itself.

Table 8 shows a clear degradation in safety under all three control settings. Replacing the reference directions with random orthogonal directions increases the mean HS from 4.30% to 49.07%. The benign-prompt directions also yield substantially higher HS, with a mean of 23.80%. Thus, reproducing the same direction estimation procedure with unrelated benign prompts does not recover the safety protection of the original reference directions.

Table 8: Direction-specific controls under the default AGNEWS setting. Parentheses indicate random seeds, benign-prompt sources, or layer-permutation seeds for the corresponding settings. The mean columns report setting-wise averages.

Setting	HS↓	FA↑	Mean HS↓	Mean FA↑
Random Orthogonal (11)	50.20	83.50	49.07	84.60
Random Orthogonal (22)	48.10	85.30		
Random Orthogonal (33)	48.90	85.00		
Benign (GSM8K)	21.70	84.80	23.80	84.40
Benign (SST2)	25.90	84.00		
Layer-Shuffled (101)	39.70	85.50	37.60	85.13
Layer-Shuffled (202)	39.70	84.60		
Layer-Shuffled (303)	33.40	85.30		
AIRP (Standard)	**4.30**	**86.10**	**4.30**	**86.10**

Table 9: Additional ablations of AIRP design elements under the default setting.

(a) Standalone and local configurations.

Configuration	HS↓	FA↑
SFT	71.80	84.30
W	39.20	83.60
P	66.00	84.70
A	54.80	84.70
P + L	59.90	85.90
P + A	45.70	84.50
W + A	9.60	83.10

(b) Configurations around the complete method.

Configuration	HS↓	FA↑
W + P	17.20	85.60
W + P + L	12.70	86.10
P + L + A	38.10	**86.20**
W + P + L + A (AIRP)	**4.30**	86.10

Breaking the layer-wise correspondence produces a similar effect. Layer-shuffled directions yield a mean HS of 37.60%, despite preserving the original set of direction vectors. This indicates that the association between each reference direction and its corresponding Transformer layer is important for effective protection.

In contrast, FA remains relatively stable across the control settings. The mean FA values are 84.60%, 84.40%, and 85.13% for the random orthogonal, benign-prompt, and layer-shuffled controls, respectively, compared with 86.10% for standard AIRP. These results suggest that the safety benefit of AIRP depends on both the alignment-related information captured by the reference directions and their correct layer-wise correspondence, rather than on directional regularization alone.

E Additional Ablation Analysis

The cumulative ablation in Section 4.4 evaluates the incremental contributions of AIRP under a fixed ordering. This section complements that analysis with a broader set of configurations to examine the individual and combined effects of its design elements. All experiments use Llama-3-8B on AGNEWS with $n = 1{,}000$ and $p = 0.1$.

For compactness, Risk-Aware Task Weighting, Base-Directed Representation Protection, Adaptive Layer Protection, and Refusal-Anchor Supervision are denoted by W, P, L, and A, respectively. Since Adaptive Layer Protection operates on the consistency-weighted layer-wise scores $c^{l,(t)}$ within Base-Directed Representation Protection, L is evaluated only together with P. Removing W restores the unweighted task loss but leaves the sample weights within P unchanged.

Table 9a reports standalone and local configurations. Among the independently evaluated design elements, W produces the largest HS reduction, from 71.80% to 39.20%. P and A also reduce HS independently to 66.00% and 54.80%, respectively. Adding L to P lowers HS from 66.00% to

Table 10: Key characteristics of AIRP and closely related defenses.

Method	Stage	Constraint Space	Safety Reference
SafeLoRA	Post-FT	Parameter	Layer-wise Base-to-Aligned weight differences
AsFT	In-FT	Parameter	Alignment direction derived from Base-to-Aligned weight differences
SBR	In-FT	Representation	Final hidden states of safety-anchor prompts from a frozen Aligned Model
AIRP (Ours)	In-FT	Representation	Layer-wise reference directions derived from alignment-induced representation changes

59.90% while increasing FA from 84.70% to 85.90%. The combination W + A further reduces HS to 9.60%, substantially below either element alone.

Table 9b examines configurations closer to the complete method. Adding L to W + P reduces HS from 17.20% to 12.70% while increasing FA from 85.60% to 86.10%. Further adding A yields the complete AIRP configuration and reduces HS to 4.30% without decreasing FA. In contrast, removing W from the complete configuration increases HS to 38.10%, while FA remains similar at 86.20%. Overall, no standalone design element matches the complete configuration, and the results support the complementary roles of the AIRP design elements.

F Comparison with Closely Related Defenses

SafeLoRA (Hsu et al., 2024) and AsFT (Yang et al., 2026) share with AIRP the use of safety references derived from differences between aligned and unaligned model states, but operate in parameter space. SBR (Lu et al., 2026), in contrast, constrains representations during downstream fine-tuning. Table 10 summarizes their intervention stages, constraint spaces, and safety references. The specific distinctions are discussed below.

Comparison with parameter-space defenses. SafeLoRA constructs a layer-wise alignment matrix from the weight difference between unaligned and aligned models. After standard LoRA fine-tuning, it projects the LoRA updates of selected layers onto the corresponding alignment subspace. The layers to be projected are determined according to the similarity between the original and projected updates. SafeLoRA therefore operates as a selective post-fine-tuning projection in parameter space.

AsFT also derives an alignment direction from the weight difference between unaligned and aligned models. It decomposes parameter updates into components along this alignment direction and its orthogonal complement, using the latter to approximate potentially harmful update directions. During fine-tuning, AsFT penalizes the magnitude of the update component in this orthogonal subspace. Thus, whereas SafeLoRA modifies selected parameter updates after fine-tuning, AsFT constrains parameter-update directions throughout fine-tuning.

AIRP takes a different perspective by defining safety preservation relative to the representation changes introduced by alignment. Rather than using model-weight differences directly, AIRP derives a layer-wise reference direction from the Base-to-Aligned representation change of harmful prompts at each Transformer layer. These directions define a common alignment-relative geometry for downstream fine-tuning. Directional consistency characterizes how closely the alignment-induced representation changes of a downstream example follow this geometry, while Base-directed representation changes measure movement from the Aligned state toward the Base state along the same reference directions. AIRP therefore regulates downstream adaptation relative to the representation changes induced by alignment, rather than projecting parameter updates or constraining their components in a parameter-space subspace.

Comparison with representation anchoring. SBR operates directly in representation space. Before downstream fine-tuning, it records the final hidden states of a fixed set of safety-anchor prompts under a frozen safety-aligned model. During fine-tuning, it minimizes the mean-squared deviation between these cached states and the corresponding final hidden states of the current model. SBR focuses specifically on the final hidden state before the unembedding layer, which it identifies as a geometric bottleneck connecting internal representations to token generation.

Table 11: Computational overhead under the default AGNEWS setting. Fine-tuning time denotes the average time per epoch. Total time includes all training epochs and method-specific offline preparation. GPU memory is measured across four NVIDIA GeForce RTX 3090 GPUs.

Methods	Epochs	Clock Time (min)			GPU Memory (GB)	
		Offline	Fine-tuning/Epoch	Total	Peak/GPU	Sum GPUs
SFT	20	–	2.68	53.60	12.80	40.67
Lisa	20	–	2.38	47.60	12.83	40.86
SBR	20	0.01	2.33	46.61	11.82	40.43
AsFT	10	–	7.84	78.40	17.86	61.27
AIRP (Ours)	20	3.41	5.14	106.21	20.72	76.98

AIRP differs from SBR in how safety-relevant representation information is defined and preserved. Rather than treating a complete hidden state at a single layer as the representation target, AIRP uses the Base-to-Aligned representation change to define an alignment-relative reference direction at each Transformer layer. Downstream degradation is then characterized by Base-directed movement along these directions. This formulation preserves the representation changes introduced by alignment without forcing the current model to reproduce complete hidden states, while also exposing how alignment degradation varies across layers. The resulting layer-wise variation is used to adapt the strength of representation protection during fine-tuning.

The two methods also assign different roles to behavioral anchors. In SBR, safety anchors are prompts whose cached final hidden states serve as representation targets. In AIRP, the anchor set $\mathcal{D}_{\text{anchor}}$ contains harmful prompt–refusal response pairs that provide direct supervision for maintaining refusal behavior. This behavioral supervision complements the alignment-relative representation constraint by preserving the output behavior associated with the safety-aligned state.

Overall, AIRP formulates harmful fine-tuning as degradation of alignment-induced representation changes rather than as an isolated parameter-drift or representation-drift problem. The layer-wise reference directions provide a shared alignment-relative coordinate system for controlling downstream adaptation: they characterize how downstream examples relate to the representation geometry introduced by alignment, define the Base-directed changes that should be constrained during fine-tuning, and expose where such degradation concentrates across Transformer layers. Refusal supervision further preserves the behavioral outcome associated with this geometry. AIRP therefore provides a unified alignment-preservation framework that connects sample-level adaptation, layer-wise representation change, and refusal behavior.

G Computational Overhead

Table 11 compares AIRP with representative fine-tuning-stage defenses, with SFT included as the standard fine-tuning reference. We focus mainly on In-FT methods because AIRP introduces its additional computation during downstream fine-tuning.

For AIRP, offline preparation constructs the layer-wise reference directions from the reference set and precomputes direction-dependent quantities for the downstream examples, including directional consistency, sample weights, and reference displacements. This stage takes 3.41 minutes. SBR also includes offline anchor acquisition, but its default configuration uses only eight safety-anchor queries, resulting in an offline cost of 0.01 minutes. During downstream fine-tuning, AIRP requires 5.14 minutes per epoch. Its per-epoch training time is higher than SFT, Lisa, and SBR, but lower than the 7.84 minutes required by AsFT. With 20 training epochs, AIRP takes 106.21 minutes in total, compared with 78.40 minutes for AsFT with 10 epochs.

AIRP also introduces additional memory overhead due to the frozen Aligned Model and layer-wise representation computations. Its peak memory usage is 20.72 GB per GPU, with a summed peak of 76.98 GB across the four devices. The per-device peak remains within the 24 GB memory capacity of an NVIDIA GeForce RTX 3090. Overall, AIRP incurs additional computation and memory overhead for representation-level protection during downstream fine-tuning.

H LIMITATIONS AND FUTURE EXTENSIONS

AIRP currently has two main limitations. First, AIRP requires paired Base and Aligned checkpoints and harmful reference prompts to construct the layer-wise reference directions. It also requires harmful prompt–refusal response pairs for anchor supervision. Second, our current evaluation focuses on open-weight models with LoRA-based customization, and its generality beyond these settings remains to be further studied.

Future work could explore more flexible ways to construct transferable reference directions with reduced dependence on paired model states. It would also be useful to extend the current layer-wise reference direction to richer low-rank or category-conditioned representations. Finally, AIRP could be evaluated under full-parameter fine-tuning, other parameter-efficient adaptation methods, and larger model families.