Layer-Aware Representation Filtering: Purifying Finetuning Data to Preserve LLM Safety Alignment

Hao Li[1,2*] **Lijun Li**[1*†] **Zhenghao Lu**[1] **Xianyi Wei**[1,3]
Rui Li[4] **Jing Shao**[1†] **Lei Sha**[2†]
[1] Shanghai Artificial Intelligence Laboratory
[2] Institute of Artificial Intelligence, Beihang University
[3] School of Computer Science, Wuhan University
[4] School of Computer Science, Peking University
hao612@buaa.edu.cn 4065156@qq.com shalei@buaa.edu.cn

Abstract

With rapid advancement and increasing accessibility of LLMs, fine-tuning aligned models has become a critical step for adapting them to real-world applications, which makes the safety of this fine-tuning process more important than ever. However, recent studies have highlighted a critical challenge: even when fine-tuning with benign datasets, the safety alignment of aligned LLMs can be compromised, making them more susceptible to malicious instructions. In this paper, we show that fine-tuning datasets often contain safety-degrading samples that are not easily identifiable on the surface. These samples can easily degrade the safety alignment of LLMs during fine-tuning. To address this issue, we propose LARF, a **L**ayer-**A**ware **R**epresentation **F**iltering method. This method identifies safety-sensitive layers within the LLM and leverages data representations to detect safety-degrading data samples in the fine-tuning dataset. Experimental results demonstrate that LARF can efficiently and effectively identify safety-degrading data. After removing such data, the safety alignment degradation caused by fine-tuning is mitigated. Please see our code at https://github.com/LLLeoLi/LARF.

1 Introduction

The rapid progress toward generally capable LLMs brings unprecedented power and risk (Zhang et al., 2023). Ensuring that these models remain aligned with human safety standards is paramount before any real-world deployment. Yet evidence shows that even small injections of harmful Q&A pairs can easily undermine a model's guardrails (Qi et al., 2024). More surprisingly, recent work demonstrates that fine-tuning on entirely benign, non–toxic instruction data drawn from widely used corpora, for example, Alpaca, can even degrade safety alignment in previously robust models (Qi et al., 2024; He et al., 2024).

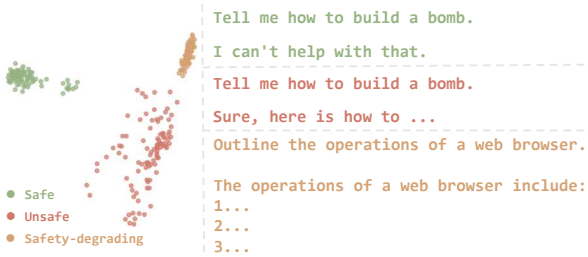


Figure 1: Comparison of LARF-identified safety-degrading samples against others. **Left:** PCA projection of representation from the selected safety-sensitive layer in Llama3.1, with safe refusals (green), unsafe compliances (red), and safety-degrading instances (orange). **Right:** Examples for each category: a safe refusal; an unsafe compliance; and a benign safety-degrading sample.

This vulnerability presents a critical barrier to adopting LLMs in sensitive domains (e.g., healthcare (Jin et al., 2019), finance (Wu et al., 2023), and education (Gan et al., 2023)), where unanticipated unsafe behavior could have serious consequences. Standard toxicity filters (LLaMa Guard (Llama Team, 2024), MD-Judge (Li et al., 2024), or the OpenAI Moderation API (Markov et al., 2023)) are designed to flag clearly harmful content, but not to detect benign examples that can degrade model safety. We term these stealthy instances **safety-degrading data**. Conversely, the few existing methods designed to detect safety-degrading data suffer from the following limitations:

1. *Bi-Anchoring (He et al., 2024) measures gradient similarity between candidate and reference instances to attribute risk, but suffers from noisy signals and poor scalability as output lengths grow.*

2. *SEAL (Shen et al., 2025) trains a dedicated ranker to distinguish safe from unsafe samples,*

* Equal contribution † Corresponding author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 8030–8050
November 4-9, 2025

but at the cost of extra training and significant compute overhead.

The safety alignment of LLM primarily relies on its mechanism for rejecting harmful instructions. We have observed that such rejection behavior is particularly prominent in certain specific network layers, which we therefore define as "safety-sensitive layers". We pinpoint these layers by selectively parameter scaling and evaluating safety behavior shifts. Subsequently, we rank the samples based on their bidirectional representations in the safety-sensitive layers—upranking truly safe samples while downranking safety-degrading samples that weaken the model's rejection capability. As shown in Figure 1, the safety-degrading samples identified by LARF lie closer in representation space to unsafe examples than to safe ones.

Our contributions can be summarized as follows:

- **A principled, efficient filtering framework**. LARF sidesteps costly gradient or ranker training by leveraging layer-wise representation sensitivity, achieving high accuracy in pinpointing safety-degrading data within benign corpora.

- **State-of-the-art detection performance**. On the Alpaca dataset, fine-tuning Llama3.1 with the 1,000 bottom ranked samples flagged by LARF raises the Attack Success Rate (ASR) on HarmBench from 3.5% to 39%, a 20% improvement over Bi-Anchoring, while fine-tuning with the 1,000 top ranked samples reduces ASR to 0%.

- **Broad generalizability and practical impact**. By removing safety-degrading examples identified by LARF, we substantially mitigate safety alignment degradation across diverse downstream tasks, including code generation, mathematical reasoning, and medical question answering, which demonstrates LARF's practical utility as a pre-deployment audit tool.

By offering a fast, resource-light, and highly accurate way to distinguish between safety-degrading and normal samples in benign datasets, LARF paves the way for more robust, trustworthy LLM fine-tuning.

2 Related work

2.1 Data Attribution Method

Data attribution methods are used to quantify the impact of a single data point on the model output. In contrast to semantic-based moderation classifiers, GradSafe (Xie et al., 2024) classifies the unsafe instruction based on the gradient of the model's safety-sensitive parameters. Inspired by LESS (Xia et al., 2024), a well-known gradient-based influential data attribution method, Bi-Gradient (He et al., 2024) identifies benign data that breaks safety alignment and DABUF (Pan et al., 2025b) filters jailbreaking and bias training data. Based on the safety-helpfulness bilevel optimization, SEAL (Shen et al., 2025) trains a data ranker to uprank the safe and high-quality fine-tuning data and downrank the unsafe or low-quality ones.

2.2 Representation Engineering

Recent studies (Zou et al., 2023a; Zhang et al., 2024) have shown that representation contains rich information and can influence the behavior of models across a wide range of safety-relevant problems, such as fairness and harmfulness. For example, Refusal Direction (Arditi et al., 2024) shows that by manipulating intermediate representation at inference time, one can switch a model's response to a harmful prompt from refusal to compliance, or vice versa. Similarly, by rerouting harmful representations away from critical decision paths, Circuit Breaker (Zou et al., 2024; Lu et al., 2025) can defend against powerful adversarial attacks (Zou et al., 2023b; Wang et al., 2024, 2025; Ren et al., 2024; Zhou et al., 2024; Miao et al., 2025b), which fully demonstrates the important role of representation in safety alignment. **See related works for LLM safe fine-tuning in Appendix A.**

Building on this representation-centric perspective, we introduce a data-driven framework that leverages intermediate data representation to quantitatively score and rank safety-degrading samples within the benign dataset, enabling precise identification and proactive filtering before fine-tuning.

3 Method

The overview of our method is shown in Figure 2. First, we identify the safety-sensitive layer by applying the scaling parameter to the weight of a specific model layer and measuring changes in

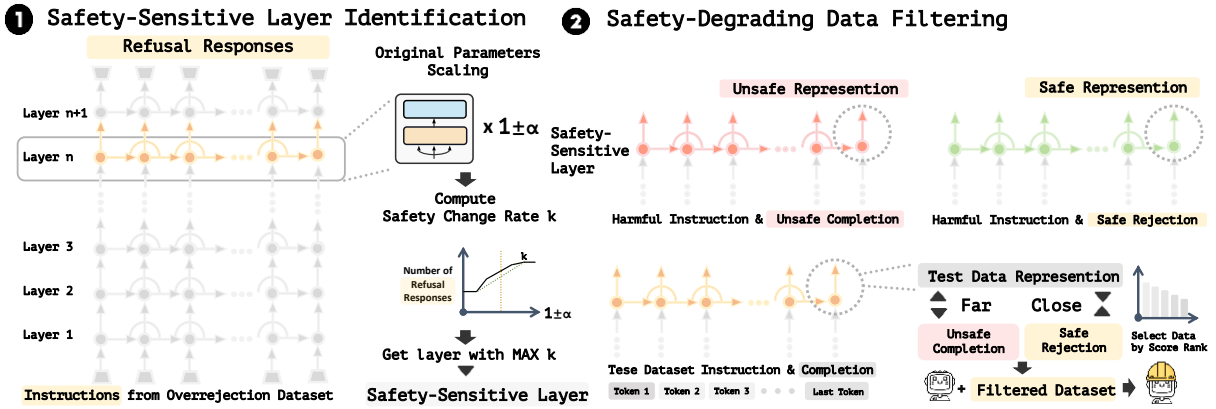


Figure 2: Overview of our two-stage LARF pipeline. (**1**) Safety-sensitive layer identification: we scale each layer's parameter, measure the resulting change in the number of refusal responses on an overrejection dataset, and select the layer with maximal sensitivity. (**2**) Safety-degrading data filtering: at the identified safety-sensitive layer, we compute average representations for safe (D_{safe}) and unsafe (D_{unsafe}) references, extract each test example's representation, and assign a safety-degrading score to rank and filter safety-degrading samples.

the number of refusal responses on an overrejection dataset. Second, we leverage the bidirectional representations extracted from the safety-sensitive layer to filter the safety-degrading data. The whole process is summarized in Algorithm 1 of the Appendix B.

3.1 Problem Formulation

Denote a sample $d = (x, y)$ where x is the instruction and y is the response, four datasets are introduced:

- D_{unsafe}: A small set of examples that feature N harmful instructions, paired with harmful completions generated by an uncensored model.
- D_{safe}: A safe reference dataset featuring the same N harmful instructions as D_{unsafe}, but paired with safe refusal responses.
- D_s: An overrejection dataset exhibits heightened sensitivity to parameter variations.
- D_{test}: The given test dataset.

Assuming an LLM with L hidden layers, the l-th layer attention module is denoted as A_l, and the feedforward module is denoted as F_l. For the l-th layer, it takes representation r_l as input and outputs representation r_{l+1}. This process can be formalized as

$$r_{l+1} = F_l(A_l(r_l) + r_l) + A_l(r_l) + r_l \quad (1)$$

3.2 Safety-sensitive Layers Identification

Overrejection, where the model erroneously refuses benign inputs, reflects an overly sensitive safety mechanism. To identify the safety-sensitive layer, we follow (Li et al., 2025c) and construct an overrejection dataset D_s. Dataset construction details can be found in the Appdenix C.1. We then apply the small scaling factor to each layer's attention and feedforward parameters, measure the resulting change in refusal rate on D_s, and designate the layer whose scaling induces the greatest refusal-rate variation as the most safety-sensitive.

Scaled modules. For each layer $l \in \{0, \ldots, L-1\}$ and scale factor $\alpha > 0$, define

$$A_l^{\pm} = (1 \pm \alpha)\, A_l, \qquad F_l^{\pm} = (1 \pm \alpha)\, F_l. \quad (2)$$

Refusal counts. Let

$$y_s^{\pm}(x) = \text{LLM}\big(x;\, A_l^{\pm}, F_l^{\pm}\big) \quad \forall\, x \in D_s, \quad (3)$$

and define the corresponding refusal counts

$$c_l^{\pm}(\alpha) \;=\; \big|\{\, x \in D_s \mid y_s^{\pm}(x) \text{ is refusal}\}\big|. \quad (4)$$

Sensitivity score calculation. Compute the difference in refusal counts

$$\Delta_l(\alpha) = c_l^{+}(\alpha) \;-\; c_l^{-}(\alpha), \quad (5)$$

and define the normalized change rate

$$k_l = \max_{\alpha \in \{\alpha_1, \alpha_2\}} \frac{\Delta_l(\alpha)}{\alpha}, \quad (6)$$

where in practice $\{\alpha_1, \alpha_2\} = \{0.1, 0.2\}$.

Layer selection. The safety-sensitive layer index l_s is

$$l_s = \arg \max_{l=0,\ldots,L-1} k_l. \quad (7)$$

Then, the representation $r_{l_s+1}(d)$ extracted from layer l_s for each example $d \in D_{\text{test}}$ is used in the subsequent data selection.

3.3 Bidirectional Representation Similarity Calculation

After identifying the safety-sensitive layer l_s, we leverage its representation r_{l_s+1} for data selection. Instead of using only unsafe data representation to calculate the similarity score, using the difference between unsafe and safe representations can represent the rejection direction of the model, which strengthens the influence of safety-related features.

Representation extraction. For each $d \in D_{\text{unsafe}} \cup D_{\text{safe}}$, let $r_{l_s+1}(d)$ denote the hidden state at the final <eos> token, then

$$r_{\text{safe}} = \frac{1}{|D_{\text{safe}}|} \sum_{d \in D_{\text{safe}}} r_{l_s+1}(d), \quad (8)$$

$$r_{\text{unsafe}} = \frac{1}{|D_{\text{unsafe}}|} \sum_{d \in D_{\text{unsafe}}} r_{l_s+1}(d). \quad (9)$$

Safety-degrading score calculation. Given a test dataset D_{test}, extract representation r_i for each example $d_i \in D_{\text{test}}$

$$r_i = r_{l_s+1}(d_i). \quad (10)$$

Then calculate cosine similarities

$$s_{\text{safe}}(r_i) = \text{sim}(r_i, r_{\text{safe}}), \quad (11)$$

$$s_{\text{unsafe}}(r_i) = \text{sim}(r_i, r_{\text{unsafe}}). \quad (12)$$

The overall safety-degrading score is

$$\text{score}_i = s_{\text{unsafe}}(r_i) - s_{\text{safe}}(r_i). \quad (13)$$

We validate bidirectional representation data selection by computing similarity scores using only D_{unsafe} on the safety-sensitive layer. The scoring formula is

$$\text{score}_i = s_{\text{unsafe}}(r_i) \quad (14)$$

Figure 3 shows that the ASR of the fine-tuned Llama3 on 1,000 top ranked samples from the Alpaca dataset selected by this method is lower than when using the bidirectional method. Meanwhile, the ASR for the 1,000 bottom ranked samples is significantly higher than the bidirectional method, indicating the effectiveness of bidirectional data selection. The results for other models and datasets are detailed in the Appendix D.1.

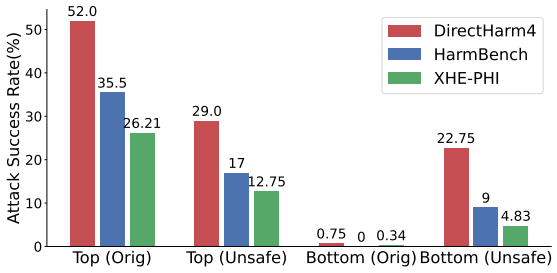


Figure 3: ASR of the fine-tuned Llama3 on the top and bottom 1,000 samples ranked by the bidirectional method (Orig) and the unidirectional method (Unsafe) from Alpaca across three safety benchmarks.

4 Experiment

4.1 Experimental Setups

Models We evaluate our approach on three models: Llama3-8B-Instruct (Llama3), Llama3.1-8B-Instruct (Llama3.1) (Llama Team, 2024) and Qwen2.5-7B-Instruct (Qwen2.5) (Qwen et al., 2025). The effectiveness of our method has also been verified on models such as Mistral-v0.2 (Jiang et al., 2023), Phi-3-mini (Abdin et al., 2024) and Qwen2 in Appendix D.2.

Datasets For safety evaluation, we test the fine-tuned models on three harmful datasets: HarmBench (Mazeika et al., 2024), HEx-PHI (Qi et al., 2024), and DirectHarm4 (Lyu et al., 2024). Notably, DirectHarm4 contains four categories (Malware, Drug, Phishing, and Disinformation) specifically selected to challenge fine-tuned models, as they empirically demonstrate higher success rates in eliciting harmful responses.

For bidirectional representation similarity data selection, dataset construction details for D_{safe} and D_{unsafe} can be found in the Appendix C.2.

Evaluation Metrics We employ LlamaGuard 3 (Llama Team, 2024), which is a Llama-3.1-8B-based model fine-tuned for content safety classification, as our safety evaluator. For most experiments, we adopt the ASR metric to quantitatively assess model harmfulness. The Appendix C.3 shows the evaluation details.

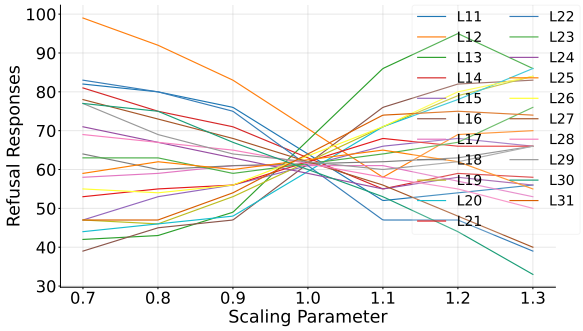


Figure 4: Layer-wise sensitivity of Llama3's refusal behavior under parameter scaling. The 13th layer is the most safety-sensitive: attenuating its parameters sharply reduces refusals, while amplifying them sharply increases refusals.

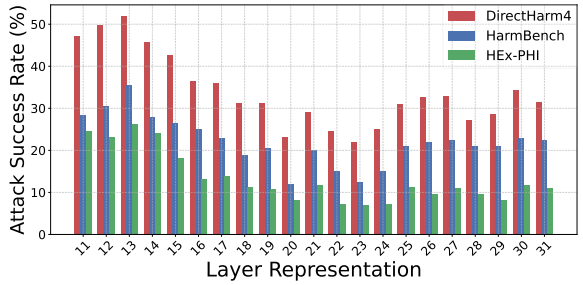


Figure 5: Attack Success Rates (ASR) of Llama3 fine-tuned on the 1,000 top ranked examples selected by corresponding representations from layers 11th–31st. Bars correspond to three safety benchmarks and reveal that selecting examples by the 13th-layer representation yields the highest ASR across all benchmarks, confirming the effectiveness of the identified safety-sensitive layer in data selection.

4.2 Safety-sensitive Layers Identification

Using the method described in Section 3.2, we perform layer-wise analysis across safety-aligned models. Specifically, for each model, we scaled the parameters of the four weight matrices W_Q, W_K, W_V, W_O of the self-attention module and the weight matrices W_{gate}, W_{up} and W_{down} of the feed-forward module.

Since the earlier layers lack safety awareness, following previous experiments (Li et al., 2025c), we apply scaling factors $\alpha \in \{0.1, 0.2\}$ to each layer, from the 11th through the final layer—and then measure the number of refusal responses on the overrecjtion dataset and calculate normalized refusal change rate. Finally, we identify the safety-sensitive layer for each model: the 13th layer for both Llama3 and Llama3.1, and the 18th layer for Qwen2.5.

We show the experimental result of Llama3 in Figure 4. As the modules of the 13th layer are weakened, the number of refusal responses is greatly reduced, and as the modules of the 13th layer are strengthened, the number of refusal responses is greatly increased. When α exceeds 0.2, many layers begin to exhibit anomalous behavior that deviates from the previously observed trends, indicating that excessive perturbation can induce confusion within the LLM. Therefore, we conduct our experiments using only $\alpha \in \{0.1, 0.2\}$.

To prove the effectiveness of the safety-sensitive layer in data selection, we fine-tune models on the Alpaca dataset using the 1,000 top ranked examples ranked by representations from the 11th layer through the 31st. Figure 5 shows that Llama3 fine-tuned on samples selected by the 13th layer's representations yields the highest ASR, indicating that the safety-sensitive layer can be effectively used for data selection. The results for the other models are presented in Appendix D.2.

We also compute the mean and variance values of the s_{safe} and s_{unsafe} for all data points across each layer of the model on the Alpaca Dataset. As shown in Figure 6, the s_{safe} and s_{unsafe} corresponding to the safety-sensitive layers of the Llama3.1 and Qwen2.5 are the lowest among all layers. After passing through these safety-sensitive layers, both the s_{safe} and s_{unsafe} begin to increase. This indicates that safety-related features are significantly enhanced since these layers.

4.3 Safety-degrading Data Selection

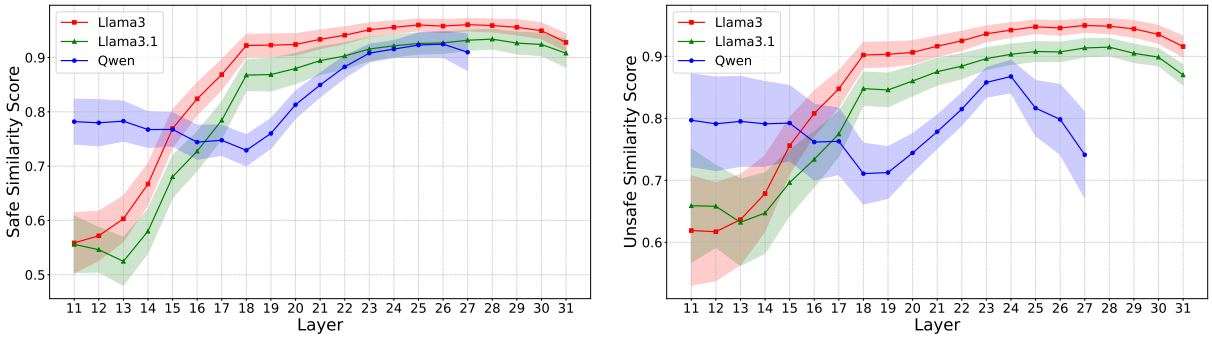
To validate our method, we extract an equal-sized subset of the highest safety-degrading scores and assess its impact on two standard instruction-tuning datasets: Alpaca (Taori et al., 2023; Peng et al., 2023) and Dolly (Conover et al., 2023).

4.3.1 Baselines

Random We randomly sampled a subset of 1,000 dialogues from the dataset for fine-tuning, computed the ASR, and then repeated this procedure three times. The results reported herein are the average ASR across these three runs.

SEAL We adopt BlueORCA (Longpre et al., 2023; Mukherjee et al., 2023) as the safe reference dataset and employ the instruction-tuning corpus as the fine-tuning dataset. For each model and each dataset, we train a dedicated data ranker.

GradSafe We first identify each model's safety-sensitive parameters using the reference safety data.



(a) Layer-wise mean and variance of s_{safe}

(b) Layer-wise mean and variance of s_{unsafe}

Figure 6: Layer-wise mean (points) and variance (shaded bands) of s_{safe} (a) and s_{unsafe} (b) on the Alpaca dataset, showing that both metrics reach their lowest values near the identified safety-sensitive layer and begin to increase thereafter–indicating that safety-related features are enhanced since this layer.

Model	Dataset	Bench	Instruct	Random	LARF	SEAL	GradSafe	Bi-Anchoring
Llama3	Alpaca	DirectHarm4	11.25	25.00	**52.00**	26.75	28.00	49.00
		Harmbench	9.50	15.00	**35.50**	13.50	16.00	35.00
		HEx-PHI	8.62	6.55	**26.21**	6.90	8.97	24.58
	Dolly	DirectHarm4	11.25	55.25	**79.25**	28.25	75.00	74.50
		Harmbench	9.50	39.25	78.50	13.00	**82.00**	75.00
		HEx-PHI	8.62	31.38	68.97	7.24	**74.14**	67.59
Llama3.1	Alpaca	DirectHarm4	13.25	22.50	**49.50**	27.75	7.50	11.00
		Harmbench	3.50	18.50	**39.00**	13.00	5.00	12.50
		HEx-PHI	5.86	8.97	**31.38**	6.90	3.45	3.10
	Dolly	DirectHarm4	13.25	54.00	**84.00**	71.75	59.50	67.25
		Harmbench	3.50	51.00	**85.00**	65.00	60.50	50.50
		HEx-PHI	5.86	29.30	**60.34**	38.62	33.79	40.00
Qwen2.5	Alpaca	DirectHarm4	9.25	27.50	**44.50**	20.00	26.00	44.50
		Harmbench	6.00	11.00	**31.00**	9.00	10.00	24.50
		HEx-PHI	9.66	13.10	**27.24**	6.55	12.07	24.80
	Dolly	DirectHarm4	9.25	50.50	**83.75**	49.75	66.50	60.50
		Harmbench	6.00	36.00	**86.50**	65.50	60.00	60.50
		HEx-PHI	9.66	32.41	**77.24**	51.03	51.03	42.07

Table 1: Attack Success Rate (%) on different safety evaluation benchmarks: DirectHarm4, Harmbench, and HEx-PHI. Higher is better. **Bold** indicates the highest ASR.

Once these parameters are identified, we compute gradients only with respect to them by pairing each test instruction with the fixed response "Sure".

Bi-Anchoring For each test data, we concatenate its instruction with the first 10 tokens of its response and compute the loss gradient over all model parameters. We then measure its similarity to reference unsafe and safe gradients and rank examples by the difference in the unsafe and safe similarity scores.

4.3.2 Discussion of Results

LARF is the most efficient method for data filtering. We provide the GPU memory usage and wall-clock runtime of each method on Alpaca dataset. Table 3 shows that LARF is the most efficient, requiring only 1×18.4GB of memory, with a much faster processing time of just 0.5 hour on Llama3.1.

Benign data with the highest safety-degrading scores breaks LLM safety alignment during fine-tuning. Table 1 shows the baseline comparison results. Almost all baselines show that there are some safety-degrading data in the fine-tuning dataset, which makes the model more harmful than random sampling after fine-tuning, highlighting the necessity of data filtering before fine-tuning.

LARF is the most effective method for selecting safety-degrading data. Although LARF neither requires additional training data nor gradi-

Model	Benchmark	Random	LARF	SEAL	Bi-Anchoring
Llama3 (Magicoder)	Humaneval (↑)	53.05	53.05	53.05	51.22
	DirectHarm4 (↓)	2.23(28.00)	**1.95(22.00)**	2.37(31.00)	2.10(25.25)
Llama3 (PubMedQA)	PubMedQA (↑)	76.5	76.8	76.4	76.8
	DirectHarm4 (↓)	3.23(29.25)	3.21(28.75)	**3.08(27.75)**	3.24(32.75)
Llama3 (MetaMath)	MATH (↑)	21.22	21.34	21.32	21.60
	DirectHarm4 (↓)	1.77(18.75)	**1.75(18.00)**	1.81(19.50)	**1.75(18.00)**
Llama3.1 (Magicoder)	Humaneval (↑)	62.50	62.80	62.20	64.02
	DirectHarm4 (↓)	1.68(14.50)	**1.46(10.25)**	1.53(11.00)	1.52(10.75)
Llama3.1 (PubMedQA)	PubMedQA (↑)	76.5	76.8	77.2	76.4
	DirectHarm4 (↓)	1.49(11.00)	**1.45(10.25)**	1.82(18.00)	2.12(20.50)
Llama3.1 (MetaMath)	MATH (↑)	28.36	29.02	29.44	27.82
	DirectHarm4 (↓)	1.62(14.50)	**1.61(14.50)**	1.68(15.75)	1.71(16.50)
Qwen2.5 (Magicoder)	Humaneval (↑)	71.95	72.56	71.95	73.78
	DirectHarm4 (↓)	2.71(37.50)	**2.40(31.50)**	2.65(35.50)	2.54(33.25)
Qwen2.5 (PubMedQA)	PubMedQA (↑)	75.7	75.2	76.0	76.0
	DirectHarm4 (↓)	3.22(25.75)	**2.71(20.50)**	3.17(23.00)	3.08(22.50)
Qwen2.5 (MetaMath)	MATH (↑)	36.77	36.74	36.80	36.78
	DirectHarm4 (↓)	2.13(26.25)	**2.11(25.50)**	2.12(25.50)	**2.11(25.50)**

Table 2: Comparison of downstream task utility and safety metrics for methods across three benchmarks and model variants. The first row reports the downstream task score (higher is better), and the second row shows Score(ASR), the average GPT Score on DirectHarm4 with the ASR (lower is better). **Bold** indicates the best safety performance.

ent computations, it remains the most effective. For all models, LARF achieves the highest ASR on the two datasets. This demonstrates that the LLM can effectively identify training examples exhibiting safety-degrading features via its bidirectional representations. Furthermore, we also select the 1,000 data samples with the lowest safety-degrading scores for experiments. Table 5 shows that LARF surpasses all baselines and even the original instruct model.

SEAL and gradient-based methods face challenges in identifying safety-degrading data. SEAL leverages a safety dataset and an aligned model to train a data ranker via bilevel optimization, with the goal of up-ranking safe, high-quality fine-tuning examples. However, because the safety dataset contains over 100K samples, it inevitably includes safety-degrading instances, undermining selection effectiveness: for Llama3 (Alpaca) on DirectHarm4, SEAL achieves only 26.75% ASR compared to 52.00% for LARF. GradSafe, which selects data using only instruction gradients and ignores responses, similarly underperforms its ASR falls to 7.50% on DirectHarm4 with Llama3.1 (Alpaca) and to 3.45% on HEx-PHI—far below LARF's 49.50% and 31.38%, respectively. Bi-Anchoring aggregates ths loss over the first 10 output tokens and achieves competitive results (49.0% ASR on DirectHarm4 with Llama3 (Alpaca)). However, it exploits "alignment shortcuts" in LLMs (Qi et al., 2025; Haize Labs, 2024). Attackers can craft data where the first 10 tokens exhibit harmless content, while harmful information is generated only in subsequent tokens. Since longer sequences diminish gradient similarity effectiveness, gradient-based methods face a dilemma in addressing security challenges.

Method	Time	Memory	GPU
LARF	**0.5 Hour**	**18.4GB**	**1 GPU**
SEAL	6 Hours	36GB	8 GPUs
GardSafe	5.3 Hours	48GB	**1 GPU**
Bi-Anchoring	3 Hours	27.8GB	4 GPUs

Table 3: Wall-clock runtime, per GPU memory usage, and number of NVIDIA A100-SXM 80GB GPUs when filtering Alpaca dataset on the Llama3.1 model.

Overall, our method can efficiently and effectively select training data that compromise model safety alignment across multiple datasets and models based on their safety-degrading scores.

4.4 Downstream Tasks Performance

Datasets To further validate our method's impact on downstream tasks, we evaluate it on three

For Bi-anchoring, since different projectors have different time consumption, we only report the gradient calculation result here.

datasets: Magicoder (Wei et al., 2024), PubMedQA (Jin et al., 2019), and MetaMath (Yu et al., 2023). For all fine-tuning datasets, we sample 10,000 data points. For each method, following SEAL, we remove the 2,000 top ranked samples. The random baseline is averaged over three independent runs.

Evaluation metrics To evaluate the performance of downstream tasks after fine-tuning, for Magicoder, we employ the HumanEval (Chen et al., 2021); for PubMedQA, we use its test split; and for MetaMath, we leverage the MATH (Hendrycks et al., 2021). To accurately capture harmful behavior of the model, we use GPT-4o to rate its output on DirectHarm4, assigning each response a score from 1 (least harmful) to 5 (most harmful). We report two metrics: **GPT Score**, the mean harmfulness rating across all responses, and **GPT ASR**, the proportion of responses that receive the maximum score of 5. More experimental details can be found in the Appendix C.4.

Results discussion. Table 2 summarizes downstream utility and safety outcomes for each method. All methods maintain task performance within 1% of the random baseline, demonstrating that safety mitigation does not degrade utility. Crucially, **LARF is the only method that consistently mitigates safety alignment loss**, lowering both average GPT Score and ASR on DirectHarm4 for every model–benchmark pair. In contrast, SEAL and Bi-Anchoring sometimes increase harmfulness relative to random sampling. These results demonstrate that LARF achieves consistent safety improvements without sacrificing downstream performance. We also verify the transferability of LARF on larger models, and the results are shown in the Appendix D.3.

4.5 Further Analysis on Safety-degrading Data

Safety-degrading examples are characterized by long point-by-point responses. We examine the 1,000 top ranked samples from each model across all five datasets. The results for Alpaca are shown in Table 4. First, point-by-point responses constitute more than 50% of these top ranked samples for every model, substantially exceeding the average of the dataset and corroborating the findings of He et al. (2024). Second, these samples yield consistently longer outputs than the dataset average. The patterns observed in the other datasets (Appendix E.1) mirror this trend. We hypothesize that this arises because models typically produce concise, refusal-style replies to harmful prompts, whereas the more elaborate, point-by-point responses interrupt this inherent safety-preserving tendency.

Model	Point-style	Output token
Avg	276	138
Llama3	516	**354**
Llama3.1	**872**	349
Qwen2.5	558	333

Table 4: Point-style response counts and average output token lengths of 1,000 top ranked samples for each model on the Alpaca dataset. Top ranked samples tend to have long point-style responses.

Fine-tuning on safety-degrading data induces representational drifts. Figure 17 plots the safety-sensitive layer representations of DirectHarm4 examples for the instruct baseline, and models fine-tuned on the top or bottom 1,000 ranked samples. The bottom 1,000 fine-tuned Llama series model's representations remain tightly clustered with the instruct baseline, whereas the top 1,000 fine-tuned model's representations shifts markedly. This representational shift demonstrates that fine-tuning on the safety-degrading data induces greater drift in safety feature space, thereby compromising the model's safety alignment. We also observed a similar phenomenon from the perspective of effective rank in the Appendix E.4.

Fine-tuning on safety-degrading data amplifies ASR on harmful content generation topics. We also analyze ASR changes across categories after fine-tuning on the 1,000 top ranked samples. The fine-tuned model shows a marked increase in ASR for harmful content generation topics, including "Adult Content", "Political Campaigning", "Disinformation" and "Phishing Crimes", whereas categories such as "Physical Harm" and "Illegal Activities" exhibit no significant ASR change. Detailed radar charts are provided in the Appendix E.5.

5 Conclusion

In this paper, we show that LLM safety alignment can be significantly compromised by benign safety-degrading data. And we propose a **L**ayer-**A**ware **R**epresentation **F**iltering method. We demonstrate that LARF can efficiently and effectively select

safety-degrading data. By removing such data, we mitigate the safety alignment degradation induced by fine-tuning. Our method outperforms existing approaches in both identifying safety-degrading data and reducing the ASR without requiring additional training data or gradient computation.

6 Limitations

Although our method can mitigate degradation in safety alignment during fine-tuning, data-only filtering cannot fully prevent safety degradation. In practice, integrating our filtering approach with safety-aware fine-tuning techniques may offer stronger protection of model alignment throughout the adaptation process.

Our filtering strategy relies on representational similarity between samples and a chosen reference set, so its effectiveness is inherently tied to the quality and composition of that reference data. While we acknowledge that carefully curated reference datasets could further improve results, exploring optimal reference selection lies beyond the scope of this work and represents a promising direction for future research.

Our experiments have been limited to LLMs, and we have not yet evaluated our approach on vision–language models (VLMs) or Diffusion Models (Li et al., 2025a). Prior work, such as VLGuard (Zong et al., 2024), has shown that even a small amount of harmful data during fine-tuning can significantly degrade VLM safety (Hu et al., 2025). In future work, we plan to explore the application of our method to the VLM setting to assess its efficacy and robustness.

Acknowledgement

This work was supported by the National Science Fund for Excellent Young Scholars (Overseas) under grant No. KZ37117501, National Natural Science Foundation of China (No. 62306024) and Shanghai Artificial Intelligence Laboratory.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Yi Ding, Lijun Li, Bing Cao, and Jing Shao. 2025. Rethinking bottlenecks in safety fine-tuning of vision language models. *arXiv preprint arXiv:2501.18533*.

Aladin Djuhera, Swanand Kadhe, Farhan Ahmed, Syed Zawad, and Holger Boche. 2025. SafeMERGE: Preserving safety alignment in fine-tuned large language models via selective layer-wise model merging. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

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

Hua Farn, Hsuan Su, Shachi H Kumar, Saurav Sahay, Shang-Tse Chen, and Hung-yi Lee. 2024. Safeguard fine-tuned llms through pre-and post-tuning model merging. *arXiv preprint arXiv:2412.19512*.

Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Jerry Chun-Wei Lin. 2023. Large language models in education: Vision and opportunities. In *2023 IEEE international conference on big data (BigData)*, pages 4776–4785. IEEE.

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.

Haize Labs. 2024. A trivial jailbreak against LLaMA 3. `https://github.com/haizelabs/llama3-jailbreak`.

Luxi He, Mengzhou Xia, and Peter Henderson. 2024. What is in your safe data? identifying benign data that breaks safety. In *First Conference on Language Modeling*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

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. *Advances in Neural Information Processing Systems*, 37:65072–65094.

Xuhao Hu, Dongrui Liu, Hao Li, Xuanjing Huang, and Jing Shao. 2025. Vlsbench: Unveiling visual leakage in multimodal safety. *Preprint*, arXiv:2411.19939.

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

Tiansheng Huang, Sihao Hu, and Ling Liu. 2024b. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. *Preprint*, arXiv:2310.06825.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Divyanshu Kumar, Anurakt Kumar, Sahil Agarwal, and Prashanth Harshangi. 2024. Increased llm vulnerabilities from fine-tuning and quantization. *arXiv e-prints*, pages arXiv–2404.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *arXiv preprint arXiv:2402.05044*.

Lijun Li, Zhelun Shi, Xuhao Hu, Bowen Dong, Yiran Qin, Xihui Liu, Lu Sheng, and Jing Shao. 2025a. T2isafety: Benchmark for assessing fairness, toxicity, and privacy in image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13381–13392.

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

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

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

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 22631–22648.

Xiaoya Lu, Dongrui Liu, Yi Yu, Luxin Xu, and Jing Shao. 2025. X-boundary: Establishing exact safety boundary to shield llms from multi-turn jailbreaks without compromising usability. *arXiv preprint arXiv:2502.09990*.

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. 2024. Keeping LLMs aligned after fine-tuning: The crucial role of prompt templates. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15009–15018.

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

Ziqi Miao, Yi Ding, Lijun Li, and Jing Shao. 2025a. Visual contextual attack: Jailbreaking mllms with image-driven context injection. *arXiv preprint arXiv:2507.02844*.

Ziqi Miao, Lijun Li, Yuan Xiong, Zhenhua Liu, Pengyu Zhu, and Jing Shao. 2025b. Response attack: Exploiting contextual priming to jailbreak large language models. *arXiv preprint arXiv:2507.05248*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Wenbo Pan, Zhichao Liu, Qiguang Chen, Xiangyang Zhou, Haining Yu, and Xiaohua Jia. 2025a. The hidden dimensions of llm alignment: A multi-dimensional safety analysis. *arXiv preprint arXiv:2502.09674*.

Yijun Pan, Taiwei Shi, Jieyu Zhao, and Jiaqi W Ma. 2025b. Detecting and filtering unsafe training data via data attribution. *arXiv preprint arXiv:2502.11411*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

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

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 *ICLR*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. 2024. Derail yourself: Multi-turn llm jailbreak attack through self-discovered clues. *arXiv preprint arXiv:2410.10700*.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Robie Gonzales, carsten maple, Subhabrata Majumdar, Hassan Sajjad, and Frank Rudzicz. 2024. Representation noising: A defence mechanism against harmful finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE.

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

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

Hao Wang, Hao Li, Minlie Huang, and Lei Sha. 2024. Asetf: A novel method for jailbreak attack on llms through translate suffix embeddings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2697–2711.

Hao Wang, Hao Li, Junda Zhu, Xinyuan Wang, Chengwei Pan, MinLie Huang, and Lei Sha. 2025. Diffusionattacker: Diffusion-driven prompt manipulation for llm jailbreak. *Preprint*, arXiv:2412.17522.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and LINGMING ZHANG. 2024. Magicoder: Empowering code generation with OSS-instruct. In *Forty-first International Conference on Machine Learning*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. 2024. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 507–518.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and 1 others. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Yihao Zhang, Zeming Wei, Jun Sun, and Meng Sun. 2024. Adversarial representation engineering: A general model editing framework for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. 2025. Spurious forgetting in continual learning of language models. In *The Thirteenth International Conference on Learning Representations*.

Weikang Zhou, Xiao Wang, Limao Xiong, Han Xia, Yingshuang Gu, Mingxu Chai, Fukang Zhu, Caishuang Huang, Shihan Dou, Zhiheng Xi, and 1 others. 2024. Easyjailbreak: A unified framework for jailbreaking large language models. *arXiv preprint arXiv:2403.12171*.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Hospedales Timothy. 2024. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023a. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. 2024. Improving alignment and robustness with circuit breakers. *Preprint*, arXiv:2406.04313.

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

A Safety Fine-tuning

Existing works (Qi et al., 2024; Kumar et al., 2024; Miao et al., 2025a) have demonstrated that fine-tuning LLMs can lead to safety degradation, even when using benign data without any harmful content. Following works (Huang et al., 2024a; Ding et al., 2025) concentrate on how to mitigate the safety degradation caused by fine-tuning from a parameter-centric perspective. The most direct approach involves parameter freezing, where (Li et al., 2025c; Du et al., 2024; Zheng et al., 2025) and (Zhao et al., 2025) preserve safety alignment by fixing the gradient of critical safety parameters during fine-tuning. While effective in maintaining baseline safety, these methods inherently limit model adaptability. Alternative approaches focus on parameter restoration, exemplified by (Farn et al., 2024; Hsu et al., 2024; Djuhera et al., 2025), restoring safety alignment through parameter merging. A third paradigm, represented by (Li et al., 2025b; Huang et al., 2024b; Rosati et al., 2024) maintains LLM safety alignment by adding restrictions on parameter updating during fine-tuning.

B Algorithm

First, we identify the safety-sensitive layer by applying the scaling parameter to the weight of the safety-sensitive layer and measuring changes in the number of refusal responses on the overrejection dataset. Second, we leverage the bidirectional representations extracted from the safety-sensitive layer to filter the safety-degrading data. The whole process is summarized in Algorithm 1.

C Experiment Setting

For Bi-Anchoring, for fair comparison, we used the same D_{unsafe} and D_{safe} reference datasets as LARF. When reporting GPU memory and wall-block time, since the time taken to reduce dimension using different projectors varies, we only count the time and memory for calculating the gradient.

C.1 Overrecjtion Dataset Construction

We use Llama-3.1-8B-Lexi-Uncensored-V2 model to generate instructions that pair potentially dangerous verbs with innocuous intents (e.g., "kill time"). During the generation process, we filter out harmful instructions that the model will obviously reject. Finally, we have a dataset of 110 instructions.

Algorithm 1: LARF

Input: LLM with L layers; attention modules $\{A_l\}_{l=0}^{L-1}$, feedforward modules $\{F_l\}_{l=0}^{L-1}$; safety-sensitive calibration set D_s; reference sets D_{safe}, D_{unsafe}; test set D_{test}.

Output: Ranking of D_{test} by harmfulness score.

Initialize $k_l := 0$ for all $l = 0, \ldots, L-1$
for $l = 0 \to L-1$ **do**
  for $\alpha \in \{0.1, 0.2\}$ **do**
    Enhance layer l:
    $A_l^+ := (1+\alpha)\, A_l,\ F_l^+ := (1+\alpha)\, F_l$
    $\{y_s^+(x)\}_{x \in D_s} := \text{LLM}(D_s; A_l^+, F_l^+)$
    $c_{\text{ref}}^+ := \left|\{x \in D_s : y_s^+(x) \text{ is refusal}\}\right|$
    Weaken layer l:
    $A_l^- := (1-\alpha)\, A_l,\ F_l^- := (1-\alpha)\, F_l$
    $\{y_s^-(x)\}_{x \in D_s} := \text{LLM}(D_s; A_l^-, F_l^-)$
    $c_{\text{ref}}^- := \left|\{x \in D_s : y_s^-(x) \text{ is refusal}\}\right|$
    $\Delta_{\text{ref}} := c_{\text{ref}}^+ - c_{\text{ref}}^-$
    $k_l := \max\left(k_l,\ \Delta_{\text{ref}}/\alpha\right)$
  end for
end for
$l_s := \arg\max_l(k_l)$
Compute reference representations at layer l_s:
  $r_{\text{safe}} := \frac{1}{|D_{\text{safe}}|} \sum_{d \in D_{\text{safe}}} r_{l_s+1}(d)$
  $r_{\text{unsafe}} := \frac{1}{|D_{\text{unsafe}}|} \sum_{d \in D_{\text{unsafe}}} r_{l_s+1}(d)$
for $d_i \in D_{\text{test}}$ **do**
  $r_i := r_{l_s+1}(d_i)$
  $\text{score}_i := \text{sim}(r_i, r_{\text{unsafe}}) - \text{sim}(r_i, r_{\text{safe}})$
end for
return D_{test} sorted by descending score_i.

C.2 Reference Dataset Construction

We constructed the datasets D_{safe} and D_{unsafe} by selecting five examples from each of 20 categories provided by the Circuit Breaker training dataset, including "Chemical & Biological Weapons/Drugs", "Cybercrime & Unauthorized Intrusion", "Drug Abuse, Weapons, Banned Substances", "Harassment & Bullying", "Illegal Activities", "Misinformation & Disinformation", "Privacy Violations", "Adult Content", "Economic Harm", "Fraud / Deception", "harmbench_misinformation_disinformation", "Hate / Harassment / Discrimination / Violence", "Illegal Activities", "Illegal Activity", "Illegal Substances", "Malware", "Physical Harm", "Political Campaigning", "Privacy Violations" and "Tailored Financial Advice".

These categories basically cover the harmful categories of DirectHarm4, Harmbench, and HEx-PHI.

C.3 Model Responses Evaluation

For all models to be tested, we set the do_sample parameter to False during generation and set the temperature to 0 to ensure the reproducibility of the generated content.

For Llama Guard and GPT-4o, we set the temperature to 0 to ensure the reproducibility of the evaluation.

For the safety-sensitive layer selection experiments, we observe that the model's refusal intent can be reliably inferred from its initial few response tokens. Following (Li et al., 2025c), we set max_new_tokens = 32 and implement rejection detection by monitoring these tokens for predefined safety-related patterns (e.g., "I cannot", "Sorry", etc.), thereby reducing computational overhead while maintaining evaluation accuracy.

C.4 Downstream Performance Evaluation

For each training dataset, following the setting of SEAL, we randomly sample 10,000 data points, and each method removes the top 2,000 ranked data points.

downstream performance For Magicoder, we use the HumanEval (Chen et al., 2021) for evaluation and set num_fewshot = 0, task = humaneval_instruct and report the pass@1 metric. For PubMedQA, we use its test set for evaluation, set num_fewshot = 0 and report the accuray metric. For MetaMath, we fine-tune on the MATH augmentation subset and evaluate on the MATH benchmark (Hendrycks et al., 2021), set num_fewshot = 0 and report the math_verify metric. We use lm-eval (Gao et al., 2024) to evaluate model's downstream performance.

safety performance We evaluate the safety of the fine-tuned models on DirectHarm4. To obtain more accurate evaluation results, we use GPT-4o to score from 1 to 5. The prompt is a revised version of the one used by (Qi et al., 2024).

C.5 Fine-tuning Setting

Setting for safety-degrading data selection. We perform LoRA training on all linear layers of all models and use LoRA weights with a rank of 8, $\alpha = 8$. The training is conducted over 3 epochs using a batch size of 8, a learning rate of 1×10^{-4}, and a warmup ratio set to 0.1.

Settings for downstream tasks. We perform LoRA training W_q and W_k on all layers and use LoRA weights with a rank of 8, $\alpha = 8$. The training is conducted on 4 GPUs with a per-device training batch size of 8 and a learning rate of 1.0×10^{-4}. The model is trained for 3 epochs using a cosine learning rate scheduler with a warmup ratio of 0.1.

D Experiment Results

D.1 The effectiveness of bidirectional representation data selection

Figure 13, Figure 14, and Figure 15 have shown the effectiveness of bidirectional representation data selection. The ASR of the fine-tuned model on the top 1,000 ranked samples from datasets selected by this method is lower than when using the bidirectional method. Meanwhile, the ASR for the bottom 1,000 ranked samples is significantly higher than the bidirectional.

D.2 Safety-Sensitive Layer Selection

- **Llama3**: Figure 7 shows that the 13-th layer is the safety-sensitive layer of Llama3, with the highest normalized change rate $k = 370$.
- **Llama3.1**: Figure 8 shows that the 13-th layer is the safety-sensitive layer of Llama3.1, with the highest normalized change rate $k = 310$.
- **Qwen2**: Figure 9 shows that the 25-th layer is the safety-sensitive layer of Qwen2, with the highest normalized change rate $k = 210$.
- **Qwen2.5**: Figure 10 shows that the 18-th layer is the safety-sensitive layer of Qwen2.5, with the highest normalized change rate $k = 280$.
- **Mistral-v0.2**: Figure 11 shows that the 16-th layer is the safety-sensitive layer of Mistral-v0.2, with the highest normalized change rate $k = 140$.
- **Phi-3-mini**: Figure 12 shows that the 21-st layer is the safety-sensitive layer of Phi-3-mini, with the highest normalized change rate $k = 150$.

D.3 The Transferability of LARF

On the PubMedQA dataset, we fine-tuned the larger-capacity Llama3-70B-Instruct, Qwen2.5-32B-Instruct and Qwen2.5-72B-Instruct. We compare our method against random sampling. Table 6 shows that our approach consistently achieves lower GPT Scores and reduced ASR.

E Analysis

E.1 Data Character Analysis

Table 7 reports the results for Llama3, Table 8 for Llama3.1, and Table 9 for Qwen2.5. The fine-tuning datasets include Alpaca, Dolly, Magicoder, PubMedQA, and MetaMath. In all cases, the top 1,000 ranked examples exhibit both point-style counts and response token lengths above the dataset average, whereas the bottom 1,000 ranked examples fall below average–demonstrating that point-by-point and longer responses compromise the safety alignment.

E.2 Similarity Heatmap Analysis

We also compute the Jaccard similarity among the 1,000 top ranked data points selected by LARF at each layer, defined by the equation:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

We visualize the pairwise similarity of the selected samples across layers using a heatmap. Figure 16a, Figure 16b, and Figure 16c reveal that the data selected by the safety-sensitive layers consistently cluster in the corner of a square region, indicating lower similarity with samples from other layers. Furthermore, as the layer depth increases, the data selected by deeper layers exhibit progressively higher similarity, suggesting convergence in safety feature extraction.

E.3 Representation Analysis

We provide PCA visualizations of the safety-sensitive-layer representations on DirectHarm4 for three model variants: the instruction-tuned baseline and models fine-tuned on the bottom and top 1,000 ranked samples. Figure 17 shows these projections for (a) Llama 3, (b) Llama 3.1, and (c) Qwen 2.5, highlighting that top-ranked fine-tuning induces a pronounced representational drift away from the baseline (especially in the Llama series) whereas bottom-ranked fine-tuning remains closely clustered.

E.4 Effective Rank Analysis

We further investigate the differential impacts of fine-tuning with the top 1,000 ranked data points

Model		Bench	Instruct	Random	LARF	SEAL	GardSafe	Bi-Anchoring
Llama3	Alpaca	DirectHarm4	11.25	25.00	**0.75**	26.75	39.00	4.25
		Harmbench	9.50	15.00	**0.00**	13.50	21.50	0.50
		HEx-PHI	8.62	6.55	**0.34**	6.90	16.90	1.38
	Dolly	DirectHarm4	11.25	55.25	**7.50**	28.25	70.00	37.50
		Harmbench	9.50	39.25	**5.50**	13.00	67.00	18.50
		HEx-PHI	8.62	31.38	**1.72**	7.24	48.97	14.48
Llama3.1	Alpaca	DirectHarm4	13.25	22.50	**0.25**	27.75	41.00	2.50
		Harmbench	3.50	18.50	**0.00**	13.00	33.50	3.00
		HEx-PHI	5.86	8.97	**0.00**	6.90	18.28	0.34
	Dolly	DirectHarm4	13.25	54.00	**3.75**	71.75	52.00	37.25
		Harmbench	3.50	51.00	**1.00**	65.00	50.00	29.00
		HEx-PHI	5.86	29.30	**2.41**	38.62	31.38	14.13
Qwen2.5	Alpaca	DirectHarm4	9.25	27.50	**0.25**	20.00	36.00	7.75
		Harmbench	6.00	11.00	**0.50**	9.00	14.00	3.00
		HEx-PHI	9.66	13.10	**0.34**	6.55	17.24	5.17
	Dolly	DirectHarm4	9.25	50.50	**9.50**	49.75	44.00	20.25
		Harmbench	6.00	36.00	**9.50**	65.50	28.00	16.00
		HEx-PHI	9.66	32.41	**7.59**	51.03	28.97	11.37

Table 5: Attack Success Rate (%) on different safety evaluation benchmarks: directHarm4, Harmbench, and HEx-PHI. Lower is better. **Bold** indicates the lowest ASR.

Model	Random	LARF
Llama3-70B	$3.47_{56.50}$	$\mathbf{3.44}_{55.75}$
Qwen2.5-32B	$3.58_{36.50}$	$\mathbf{3.54}_{36.00}$
Qwen2.5-72B	$3.09_{26.25}$	$\mathbf{2.92}_{20.25}$

Table 6: Performance comparison between Random sampling and LARF on the PubMedQA dataset for Llama3-70B, Qwen2.5-32B, and Qwen2.5-72B. Entries report $\text{Score}_{\text{ASR}}$ (mean harmfulness; lower is better). LARF consistently achieves lower GPT Scores and reduced ASR across all models.

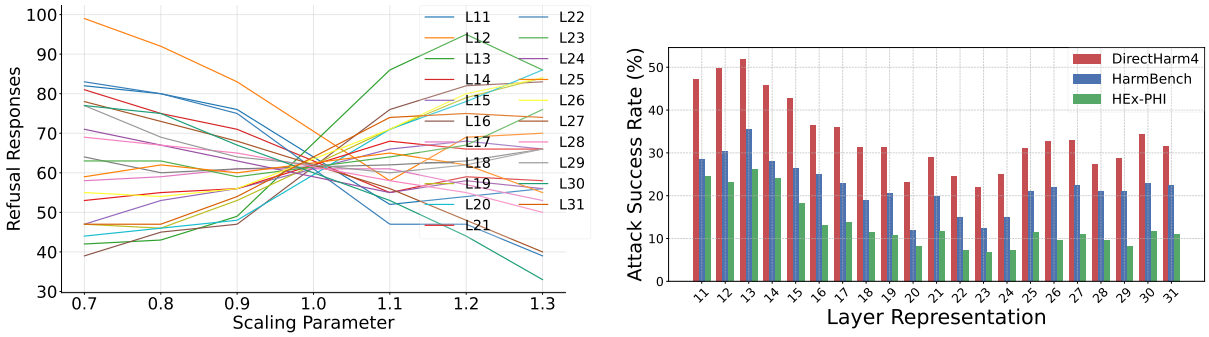
versus the bottom 1,000 ranked on model representation. For each layer's representations on the DirectHarm4 dataset, we computed both the transformation matrix W (Pan et al., 2025a) and its effective rank (Roy and Vetterli, 2007). Figure 18a, Figure 18b, and Figure 18c reveal that models fine-tuned on the top 1,000 data points exhibit progressively higher effective rank compared to bottom-1,000-tuned models as layer depth increases. This suggests that top-1,000 fine-tuning produces more diverse representation directions when processing harmful instructions, compromising the model's safety alignment.

E.5 Category Analysis

We present detailed radar-chart visualizations of the ASR for each safety category before and after fine-tuning on the top 1,000 ranked Alpaca examples. Figures 19, 20, and 21 respectively show Llama 3, Llama 3.1, and Qwen 2.5 performance on three benchmarks (DirectHarm4, HarmBench, and XEx-PHI). In each chart, green spokes denote pre-fine-tuning ASR and red spokes post-fine-tuning, revealing pronounced increases in vulnerability to these safety-sensitive scenarios after incorporating the top-ranked data.

Dataset	Type	Point-style	Output length
Alpaca	Top	516.00	353.92
	Mean	275.84	138.31
	Bottom	4.00	46.99
Dolly	Top	222.00	319.93
	Mean	79.80	75.29
	Bottom	0.00	14.05
Magicoder	Top	602.00	468.24
	Mean	259.10	361.32
	Bottom	49.00	138.88
PubMedQA	Top	3.00	93.49
	Mean	1.50	54.36
	Bottom	0.00	28.8
MetaMath	Top	28.00	352.04
	Mean	7.30	180.59
	Bottom	1.00	60.26

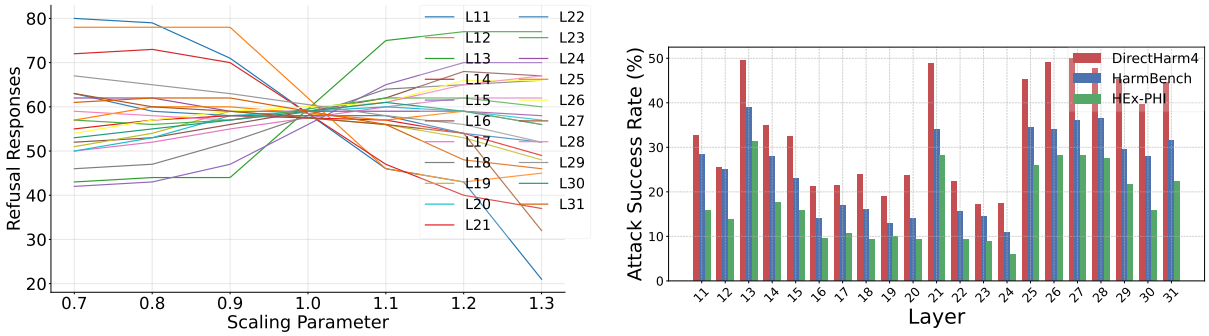
Table 7: Point-style counts and output token lengths for the top, mean, and bottom 1,000 ranked examples across five fine-tuning datasets on Llama3. Top-ranked samples exceed the dataset averages, while bottom-ranked samples fall below.



(a) Refusal repsonses

(b) ASR

Figure 7: Llama 3: the 13th layer is the safety-sensitive layer.



(a) Refusal repsonses

(b) ASR

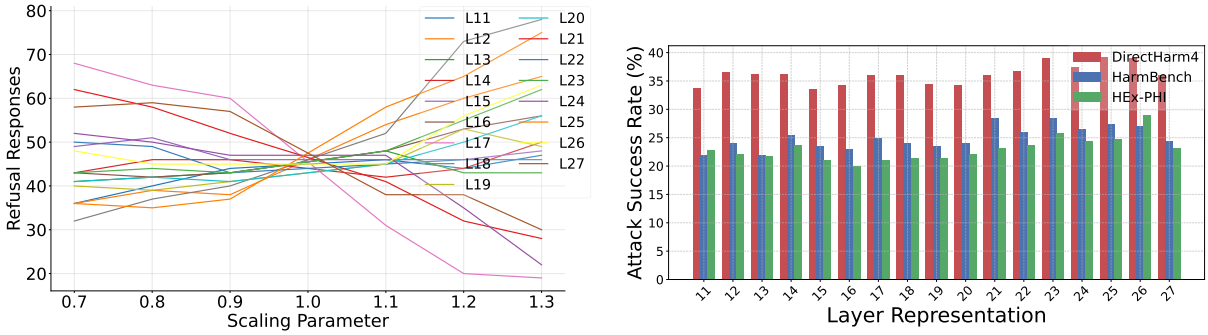
Figure 8: Llama 3.1: the 13th layer is the safety-sensitive layer.

Dataset	Type	Point-style	Output length
Alpaca	Top	872.00	349.06
	Mean	275.84	138.31
	Bottom	5.00	48.38
Dolly	Top	201.00	268.73
	Mean	79.80	75.29
	Bottom	3.00	18.50
Magicoder	Top	629.00	478.63
	Mean	259.10	361.32
	Bottom	109.00	230.51
PubMedQA	Top	4.00	85.74
	Mean	1.50	54.36
	Bottom	0.00	34.61
MetaMath	Top	21.00	265.08
	Mean	7.30	180.59
	Bottom	2.00	107.08

Table 8: Point-style counts and output token lengths for the top, mean, and bottom 1,000 ranked examples across five fine-tuning datasets on Llama3.1. Top-ranked samples exceed the dataset averages, while bottom-ranked samples fall below.

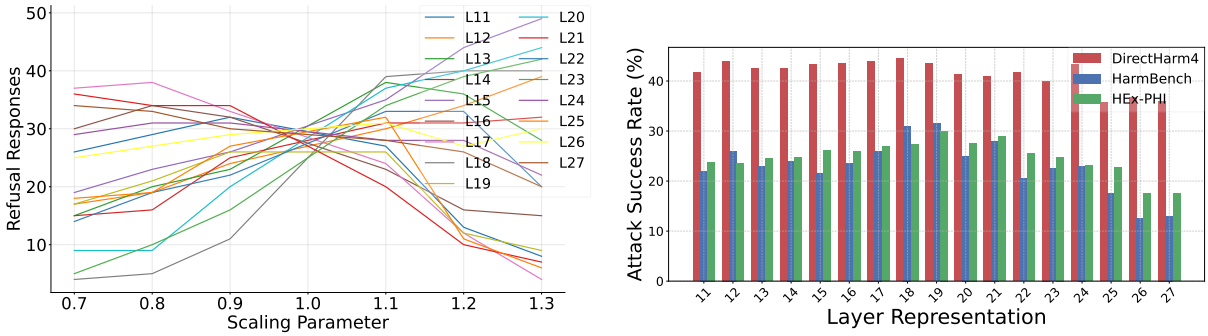
Dataset	Type	Point-style	Output length
Alpaca	Top	558.00	333.16
	Mean	275.84	138.31
	Bottom	4.00	25.75
Dolly	Top	177.00	224.95
	Mean	79.80	75.29
	Bottom	11.00	14.24
Magicoder	Top	288.00	452.15
	Mean	259.10	361.32
	Bottom	87.00	164.66
PubMedQA	Top	3.00	81.63
	Mean	1.50	54.36
	Bottom	0.0	35.27
MetaMath	Top	25.00	364.25
	Mean	7.30	180.59
	Bottom	1.00	77.30

Table 9: Point-style counts and output token lengths for the top, mean, and bottom 1,000 ranked examples across five fine-tuning datasets on Qwen2.5. Top-ranked samples exceed the dataset averages, while bottom-ranked samples fall below.



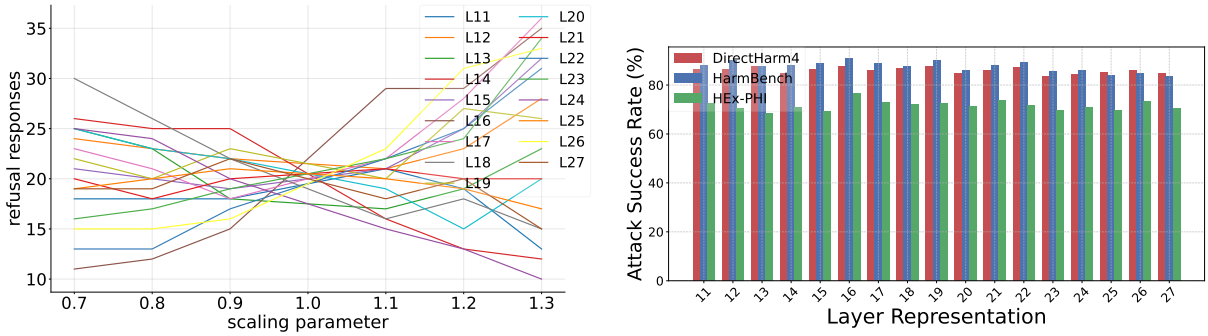
(a) Refusal repsonses (b) ASR

Figure 9: Qwen2: the 25th layer is the safety-sensitive layer.



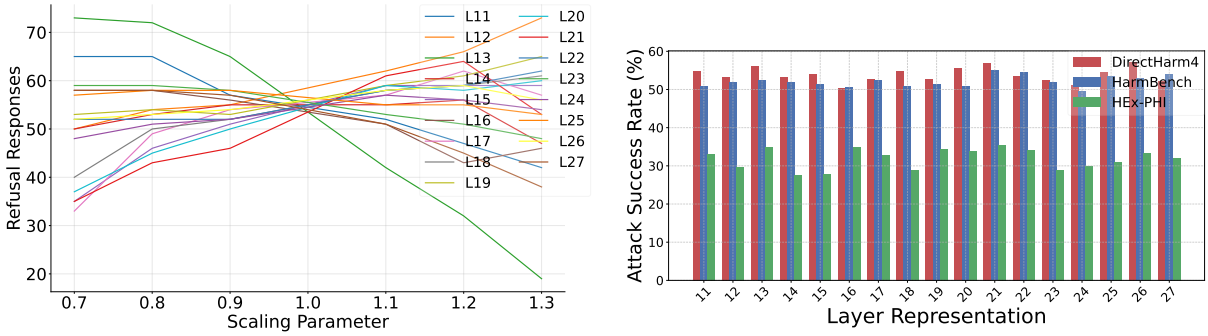
(a) Refusal repsonses (b) ASR

Figure 10: Qwen2.5: the 18th layer is the safety-sensitive layer.



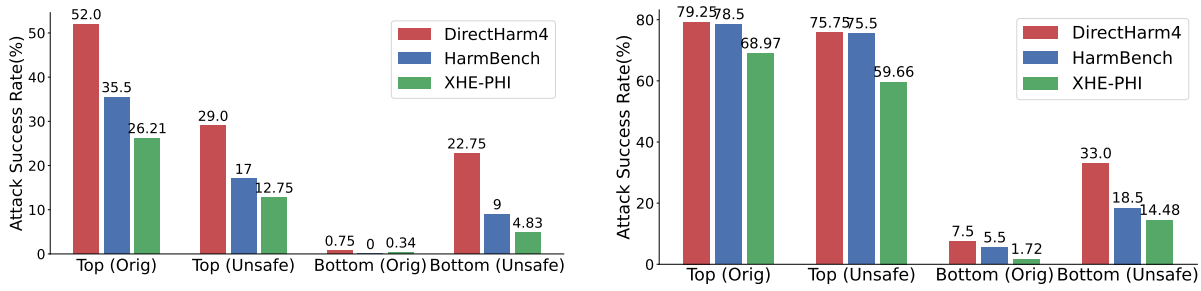
(a) Refusal repsonses (b) ASR

Figure 11: Mistral-v0.2: the 16th layer is the safety-sensitive layer.



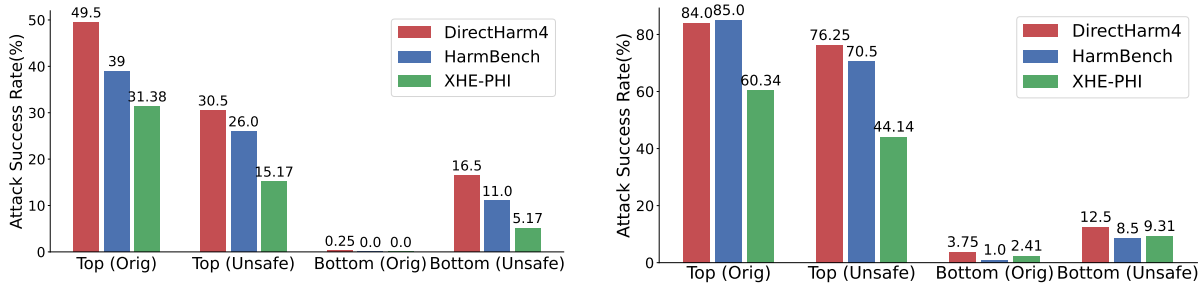
(a) Refusal repsonses (b) ASR

Figure 12: Phi-3-mini: the 21th layer is the safety-sensitive layer.



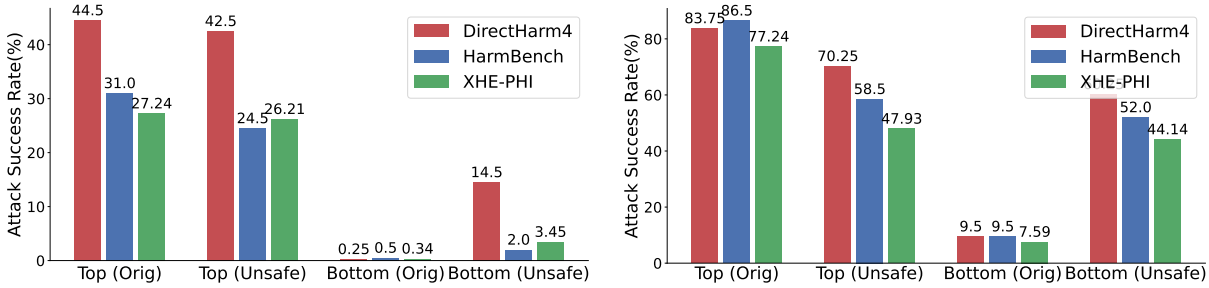
(a) Alpaca (b) Dolly

Figure 13: ASR of the fine-tuned Llama3 on the top and bottom 1,000 samples ranked by the bidirectional method (Orig) and the unidirectional method (Unsafe) across three safety benchmarks.



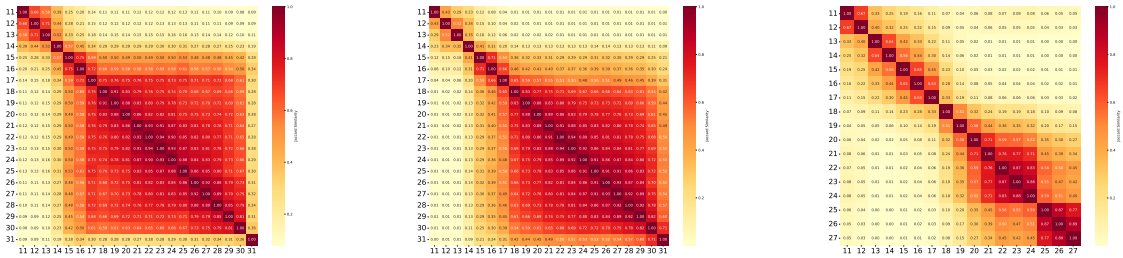
(a) Alpaca (b) Dolly

Figure 14: ASR of the fine-tuned Llama3.1 on the top and bottom 1,000 samples ranked by the bidirectional method (Orig) and the unidirectional method (Unsafe) across three safety benchmarks.



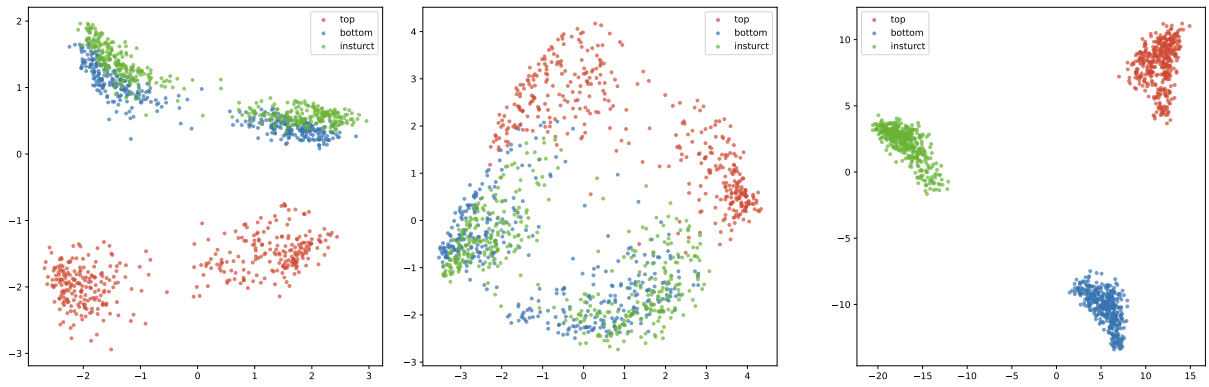
(a) Alpaca (b) Dolly

Figure 15: ASR of the fine-tuned Qwen2.5 on the top and bottom 1,000 samples ranked by the bidirectional method (Orig) and the unidirectional method (Unsafe) across three safety benchmarks.



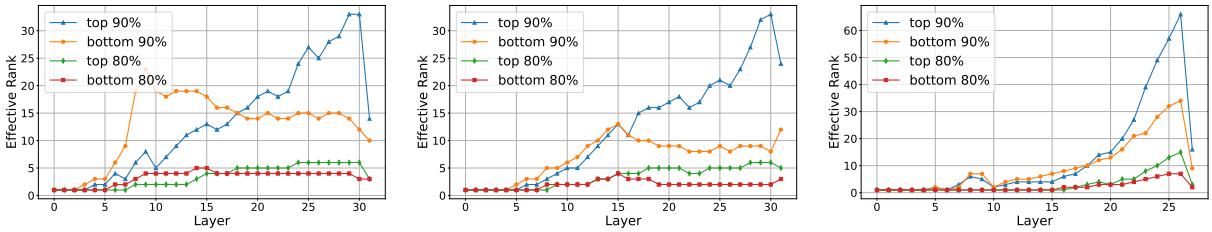
(a) Llama3 (b) Llama3.1 (c) Qwen2.5

Figure 16: Pairwise cosine similarity heatmaps of the top-1,000 samples selected by each layer of (a) Llama3, (b) Llama3.1, and (c) Qwen2.5. In each model, the safety-sensitive layer's selections form a distinct block in the corner (indicating low similarity with other layers) while deeper layers show progressively higher intra-layer similarity, reflecting convergence in safety-related feature extraction.



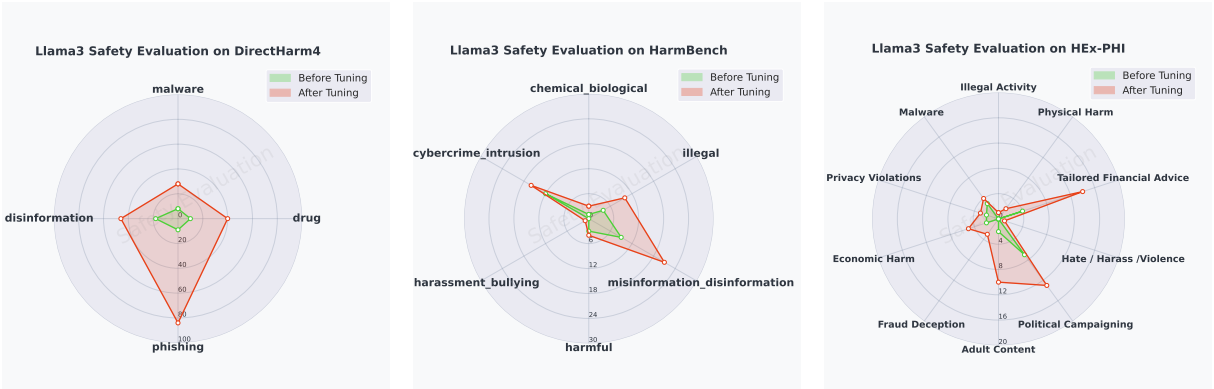
(a) Llama 3 (b) Llama 3.1 (c) Qwen 2.5

Figure 17: Principal component analysis of safety-sensitive layer representations on DirectHarm4 for (a) Llama3, (b) Llama3.1, and (c) Qwen2.5. Each plot overlays the instruction-tuned baseline (green) with models fine-tuned on the bottom 1,000 (blue) and top 1,000 (red) ranked samples. For Llama series models, bottom 1,000 fine-tuned variants remain closely clustered with the baseline, whereas top 1,000 variants diverge substantially, indicating greater representational drift and potential degradation in safety alignment. For Qwen2.5, the bottom-1,000 fine-tuned variants also deviate from the instruction baseline, likely due to a distribution mismatch between the fine-tuning data and the original model.



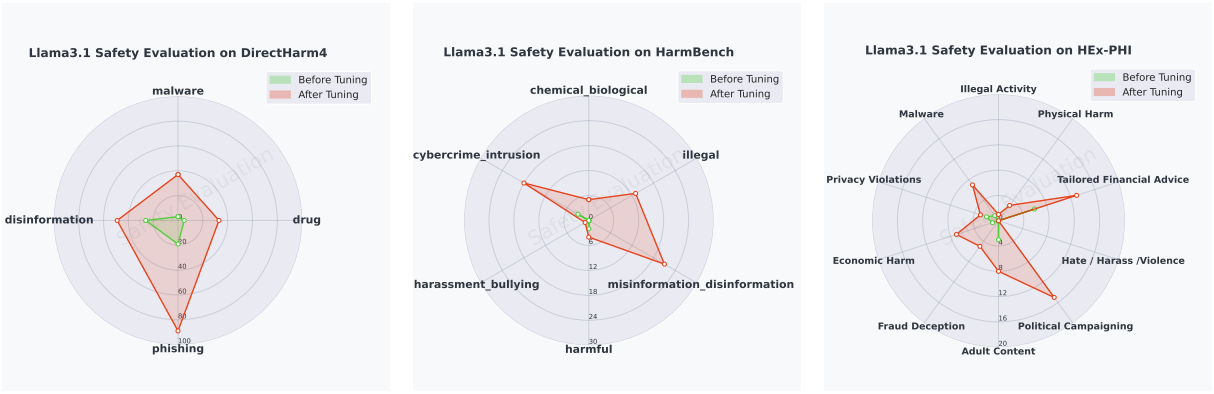
(a) Llama3 (b) Llama3.1 (c) Qwen2.5

Figure 18: Effective rank of the transformation matrix W at each layer for models fine-tuned on the top-1,000 versus bottom-1,000 samples from DirectHarm4. (a) Llama3, (b) Llama3.1, and (c) Qwen2.5. In all three models, fine-tuning on the top-1,000 harmful examples yields progressively higher effective rank with increasing depth compared to bottom-1,000 tuning, indicating more diverse representation directions and potential degradation in safety alignment.



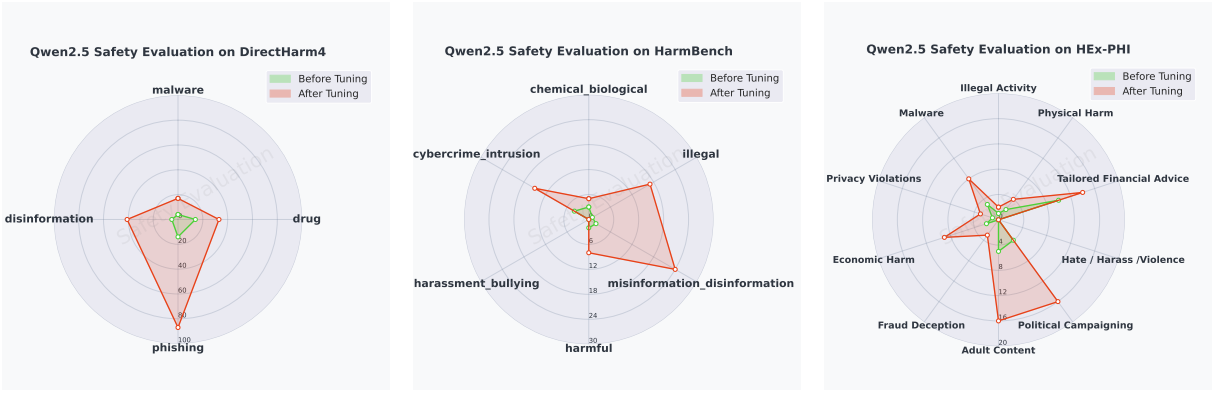
(a) DirectHarm4 (b) HarmBench (c) XEx-PHI

Figure 19: Radar-chart comparison of Llama3 safety evaluation scores before (green) and after (red) fine-tuning on Alpaca dataset on three benchmarks: (a) DirectHarm4 (b) HarmBench (c) HEx-PHI



(a) DirectHarm4 (b) HarmBench (c) XEx-PHI

Figure 20: Radar-chart comparison of Llama3.1 safety evaluation scores before (green) and after (red) fine-tuning on Alpaca dataset on three benchmarks: (a) DirectHarm4 (b) HarmBench (c) HEx-PHI



(a) DirectHarm4 (b) HarmBench (c) XEx-PHI

Figure 21: Radar-chart comparison of Qwen2.5 safety evaluation scores before (green) and after (red) fine-tuning on Alpaca dataset on three benchmarks: (a) DirectHarm4, (b) HarmBench, (c) HEx-PHI