Jailbreak to Protect: Buffering and Reinforcing via Temporary Jailbreaking for Safe Fine-Tuning in Large Language Models

Seokil Ham, Jaehyuk Jang, Wonjun Lee, Changick Kim

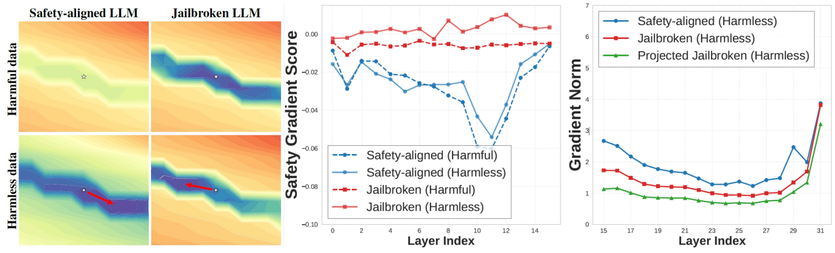
Korea Advanced Institute of Science and Technology (KAIST) / Email: {gkatjrdlf, jhyuk, dpenguin, changick}@kaist.ac.kr

Problem Setting: Fine-tuning-as-a-Service

- Recently, AI service providers such as OpenAI and Google have introduced **Fine-tuning-as-a-Service (FaaS)**, which enables users to customize LLMs with user-provided datasets.
- However, fine-tuning can weaken safety alignment, even in safety-aligned LLMs.
- This safety degradation becomes more severe when user data contains harmful samples.
- We refer to this threat as **Harmful Fine-tuning Attacks**.
- A recent work [1] activates harmful-behavior modules during fine-tuning to suppress undesired behavior learning, but **its underlying mechanism remains unclear.**

Our goal is to reveal the mechanism through gradient-level analysis and propose a Buffer-and-Reinforce fine-tuning framework that buffers harmful updates and reinforces safety while preserving user-task utility.

How jailbreaking mitigates harmful fine-tuning?



- **Safety-aligned LLM**: Llama3-8B-Instruct
- **Jailbroken LLM**: Llama3-8B-Instruct fine-tuned on LAT harmful data [2].
- **Harmful data**: BeaverTails dataset [3] (e.g., abuse, violence, hate speech, and crime).
- **Harmless data**: GSM8K dataset [4] (grade-school math problems)

- **Safety Gradient Score**

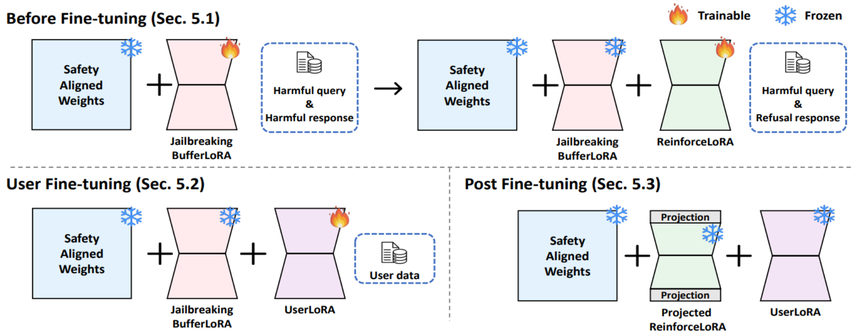
$$S^l = \frac{1}{N}\sum_{i=1}^{N}\frac{g_i^l \cdot v^l}{\|v^l\|_2 + \epsilon}$$

  - l: layer index
  - g: gradient
  - v: safety vector (from safety-trained LoRA weights)
  - ϵ: epsilon $(= 10^{-8})$

- Safety-aligned LLM has room to learn harmful data, allowing harmful fine-tuning.
- Jailbroken LLM has converged on harmful data, saturating safety-degrading gradients.
- However, jailbroken LLM retains comparable gradients on harmless data.

Therefore, temporary jailbreaking buffers harmful updates by saturating safety-degrading gradients while preserving user-task learning.

Buffer-and-Reinforce Fine-tuning Framework



Our framework consists of three LoRA modules.
- **BufferLoRA:** Temporarily jailbreaks the LLM to buffer harmful updates during user fine-tuning.
- **UserLoRA**: Learns user-specific task knowledge from user data.
- **ReinforceLoRA**: Reinforces final model safety through QR-based merging with UserLoRA.

Step 1. Before Fine-tuning

FaaS Provider pretrains BufferLoRA and ReinforceLoRA.
- **BufferLoRA:** trained on harmful query and answer pairs to induce jailbreak state.

$$\mathcal{L}_B(\theta_B) = -\mathbb{E}_{(x,y)\sim D_H}\left[\sum_{t=1}^{|y|}\log P(y_t|x, y_{<t}; \theta, \theta_B)\right]$$

- **ReinforceLoRA**: trained on harmful query and refusal answer pairs and benign data under the jailbreak state that BufferLoRA is attached to learn strong safety

$$\mathcal{L}_R(\theta_R) = -\mathbb{E}_{(x,y)\sim D_S \cup D_B}\left[\sum_{t=1}^{|y|}\log P(y_t|x, y_{<t}; \theta, \theta_B, \theta_R)\right]$$

- These modules are trained once and reused globally for all users, minimizing overhead.

Step 2. User Fine-tuning

Fine-tune the LLM on user-provided data for model customization.
- **BufferLoRA:** attached but frozen to saturate safety-degrading gradients.
- **UserLoRA**: attached and trained to learn task knowledge while harmful updates are buffered.

$$\mathcal{L}_U(\theta_U) = -\mathbb{E}_{(x,y)\sim D_U}\left[\sum_{t=1}^{|y|}\log P(y_t|x, y_{<t}; \theta, \theta_B, \theta_U)\right]$$

- BufferLoRA is detached after user fine-tuning to restore the safety-aligned base model.

Step 3. Post Fine-tuning

Integrate safety knowledge without compromising user-task performance.
- **Naively merging ReinforceLoRA with UserLoRA degrade downstream utility.**
- We use **QR decomposition** to identify the UserLoRA task subspace.

$$\hat{B}_U = Q_B R$$

- **ReinforceLoRA is projected onto the orthogonal complement of this subspace.**

$$\widetilde{W}_R = (I - \alpha Q_B Q_B^{\top}) W_R$$

$$W_{final} = W_{base} + \frac{1}{2}(W_U + \widetilde{W}_R)$$

- **QR-based merging** reinforces safety while preserving learned user-task utility.

Experiment Results

- Buffer-and-Reinforce consistently achieves low Harmful Score while maintaining high Fine-tuning Accuracy across varying harmful ratios and user data sizes.

Methods	Harmful Score (↓)					Fine-tuning Accuracy (↑)				
	$p=0.0$	$p=0.1$	$p=0.3$	$p=0.5$	$p=1.0$	$p=0.0$	$p=0.1$	$p=0.3$	$p=0.5$	$p=1.0$
SFT	$33.0_{\pm 1.0}$	$75.2_{\pm 0.5}$	$79.1_{\pm 0.1}$	$80.7_{\pm 0.5}$	$81.0_{\pm 0.2}$	$70.6_{\pm 0.8}$	$69.0_{\pm 0.9}$	$67.4_{\pm 0.4}$	$67.3_{\pm 1.7}$	-
LDIFS (Mukhoti et al., 2024)	$16.6_{\pm 0.5}$	$16.4_{\pm 0.9}$	$16.5_{\pm 0.4}$	$16.9_{\pm 1.2}$	$17.6_{\pm 0.3}$	$75.4_{\pm 0.3}$	$74.1_{\pm 1.1}$	$73.0_{\pm 1.3}$	$73.5_{\pm 0.6}$	-
SafeInstruct (Bianchi et al., 2024)	$\mathbf{7.9}_{\pm 1.1}$	$19.6_{\pm 1.6}$	$50.5_{\pm 3.5}$	$66.3_{\pm 1.8}$	$74.0_{\pm 0.8}$	$70.4_{\pm 1.3}$	$69.4_{\pm 0.5}$	$67.8_{\pm 0.3}$	$67.2_{\pm 0.2}$	-
Lisa (Huang et al., 2024a)	$14.7_{\pm 0.5}$	$29.9_{\pm 2.7}$	$49.9_{\pm 7.7}$	$64.0_{\pm 6.0}$	$73.5_{\pm 2.4}$	$70.0_{\pm 0.7}$	$69.5_{\pm 0.4}$	$67.6_{\pm 1.8}$	$66.9_{\pm 3.4}$	-
Security Vector (Zhou et al., 2024)	$24.3_{\pm 0.3}$	$22.1_{\pm 0.7}$	$22.8_{\pm 0.7}$	$25.3_{\pm 0.9}$	$23.9_{\pm 1.3}$	$72.3_{\pm 0.8}$	$71.3_{\pm 0.8}$	$69.3_{\pm 0.4}$	$68.7_{\pm 0.9}$	-
AsFT (Yang et al., 2026)	$21.1_{\pm 0.6}$	$26.7_{\pm 0.7}$	$29.6_{\pm 0.9}$	$32.2_{\pm 0.8}$	$34.1_{\pm 0.2}$	$67.0_{\pm 0.2}$	$68.4_{\pm 0.1}$	$68.9_{\pm 1.7}$	$69.3_{\pm 0.8}$	-
SafeLoRA (Hsu et al., 2024)	$20.0_{\pm 0.2}$	$26.6_{\pm 0.8}$	$31.7_{\pm 0.8}$	$41.6_{\pm 1.5}$	$53.2_{\pm 0.8}$	$75.1_{\pm 0.7}$	$73.3_{\pm 0.5}$	$73.4_{\pm 0.4}$	$73.1_{\pm 0.4}$	-
Antidote (Huang et al., 2025a)	$22.2_{\pm 3.4}$	$27.2_{\pm 1.7}$	$45.4_{\pm 19.8}$	$49.8_{\pm 20.1}$	$58.1_{\pm 14.2}$	$\mathbf{76.1}_{\pm 0.7}$	$75.0_{\pm 0.3}$	$74.2_{\pm 1.6}$	$74.7_{\pm 2.2}$	-
Panacea (Wang et al., 2026)	$19.9_{\pm 2.4}$	$36.2_{\pm 14.4}$	$47.4_{\pm 18.1}$	$52.5_{\pm 22.3}$	$64.2_{\pm 12.6}$	$68.4_{\pm 2.3}$	$67.1_{\pm 2.7}$	$65.2_{\pm 2.8}$	$67.3_{\pm 3.4}$	-
Buffer-and-Reinforce (Ours)	$8.7_{\pm 1.0}$	$\mathbf{8.1}_{\pm 0.3}$	$\mathbf{8.7}_{\pm 0.3}$	$\mathbf{8.2}_{\pm 0.3}$	$\mathbf{8.8}_{\pm 1.2}$	$76.0_{\pm 1.3}$	$\mathbf{76.6}_{\pm 1.3}$	$\mathbf{75.8}_{\pm 0.8}$	$\mathbf{75.2}_{\pm 0.4}$	-

Table 1. Comparison of safety and utility under varying **harmful data ratios (p)** in user data. Lower Harmful Score and higher Fine-tuning Accuracy indicate better performance.

Methods	Harmful Score (↓)						Fine-tuning Accuracy (↑)					
	n=500	n=1000	n=1500	n=2000	n=2500	Average	n=500	n=1000	n=1500	n=2000	n=2500	Average
SFT	61.8	75.2	77.0	79.3	80.0	74.7	67.2	68.4	70.3	71.0	71.4	69.7
LDIFS (Mukhoti et al., 2024)	19.1	17.2	18.4	16.5	18.7	18.0	68.6	73.8	71.2	66.7	69.1	69.9
SafeInstruct (Bianchi et al., 2024)	29.0	21.5	19.0	19.4	19.0	21.6	68.9	69.8	70.0	70.6	70.4	69.9
Lisa (Huang et al., 2024a)	25.5	26.8	26.0	24.7	23.1	25.2	68.2	69.1	69.7	70.1	70.0	69.4
Security Vector (Zhou et al., 2024)	20.5	21.7	23.6	21.0	23.2	22.0	71.2	72.2	71.5	71.5	70.2	71.3
AsFT (Yang et al., 2026)	23.7	26.0	28.8	29.7	33.2	28.3	70.2	68.3	67.8	70.3	68.2	69.0
SafeLoRA (Hsu et al., 2024)	21.7	26.0	25.9	24.8	27.6	25.2	73.8	72.8	74.5	74.9	74.2	74.0
Antidote (Huang et al., 2025a)	27.4	25.3	45.6	50.4	56.7	41.1	73.6	74.7	74.7	75.9	75.7	74.9
Panacea (Wang et al., 2026)	35.7	41.0	81.9	88.3	62.9	62.0	49.7	65.5	55.1	63.2	62.2	59.1
Buffer-and-Reinforce (Ours)	**8.5**	**8.4**	**8.2**	**8.7**	**9.1**	**8.6**	**75.1**	**77.5**	**77.7**	**76.3**	**76.7**	**76.7**

Table 2. Comparison of safety and utility across varying **user data sizes (n)**, ranging from 500 to 2,500. Lower Harmful Score and higher Fine-tuning Accuracy indicate better performance.

References

[1] Zhou, X., Lu, Y., Ma, R., Wei, Y., Gui, T., Zhang, Q., & Huang, X. J. (2024, August). Making harmful behaviors unlearnable for large language models. In Findings of the Association for Computational Linguistics: ACL 2024 (pp. 10258-10273).
[2] Sheshadri, A., Ewart, A., Guo, P. H., Lynch, A., Wu, C., Hebbar, V., ... & Casper, S. Latent Adversarial Training Improves Robustness to Persistent Harmful Behaviors in LLMs. Transactions on Machine Learning Research.
[3] Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., ... & Yang, Y. (2023). Beavertails: Towards improved safety alignment of llm via a human-preference dataset. Advances in Neural Information Processing Systems, 36, 24678-24704.
[4] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., ... & Schulman, J. (2021). Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.