Jailbreak to Protect

Buffering and Reinforcing via Temporary Jailbreaking for Safe Fine-Tuning in LLMs

Seokil Ham, Jaehyuk Jang, Wonjun Lee, Changick Kim

Korea Advanced Institute of Science and Technology (KAIST)

KAIST

CILAB

Threat of Fine-tuning-as-a-Service

The Rise of FaaS

Fine-Tuning-as-a-Service (FaaS) allows users to personalize Large Language Models (LLMs) with their own datasets.

Vulnerability to Attacks

This opens the door to **harmful fine-tuning attacks**, where malicious user data catastrophically weakens the model's safety.

Compliance Risks

Even small amounts of harmful data can bypass existing safety mechanisms, turning safe models into compliance risks.

Analysis of Temporary Jailbreaking

Why it works: Gradient Saturation

Recent work[1] mitigates harmful fine-tuning attacks by intentionally inducing **temporary jailbreak**, but the underlying mechanism was previously unclear.

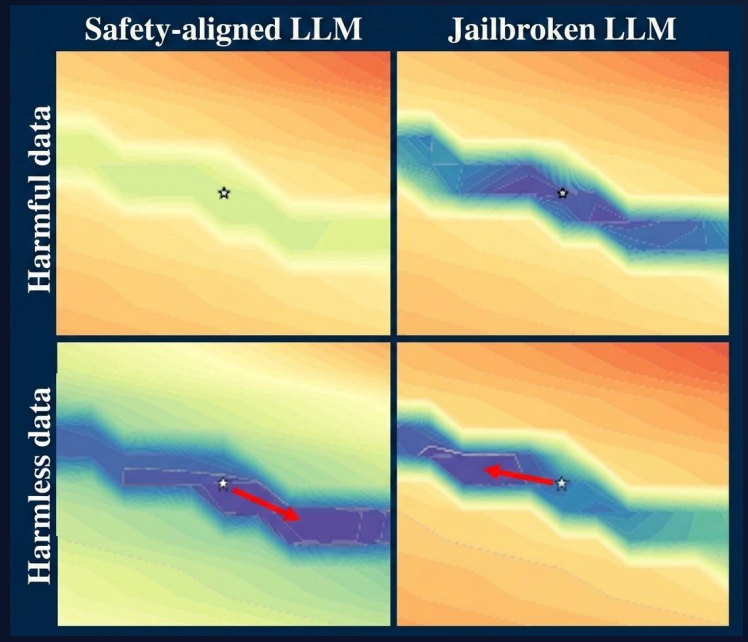
Our analysis via 2D Loss Landscapes and layer-wise metrics reveals that jailbreaking **neutralizes safety-degrading gradients** while preserving utility.

- Jailbroken LLM has already converged on harmful data, then they exhibit **near-zero safety loss gradients**.
- Safety-aligned LLM compromises safety regardless of data type.
- Both LLMs remain similar Utility-related gradient norms, ensuring task performance.

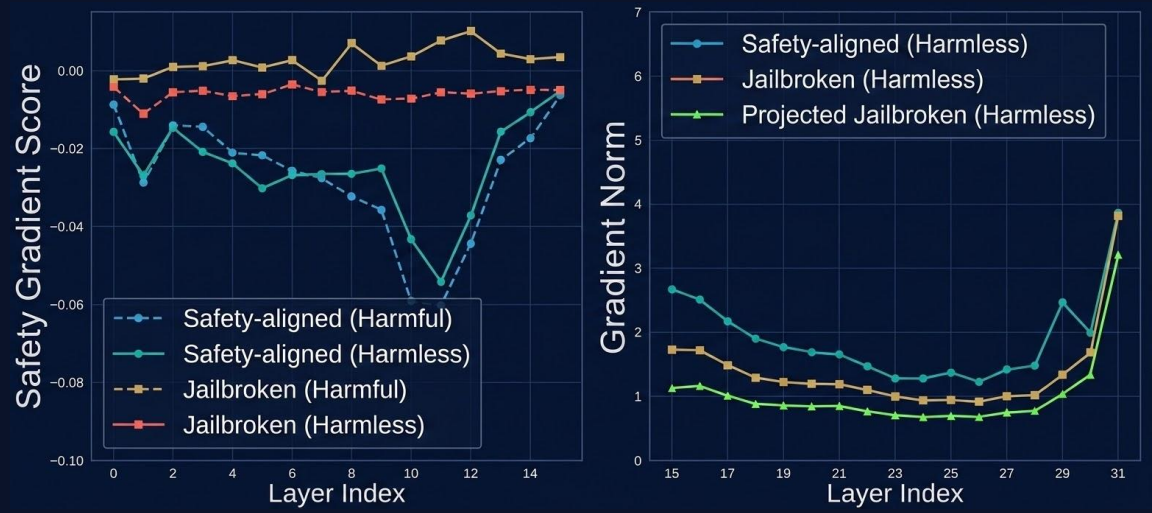
Safety Gradient Score (S)

l: layer index, v: safety-related vector, g: gradient, N: number of data

$$S^l = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{g}_i^l \cdot \mathbf{v}^l}{\|\mathbf{v}^l\|_2 + \epsilon},$$



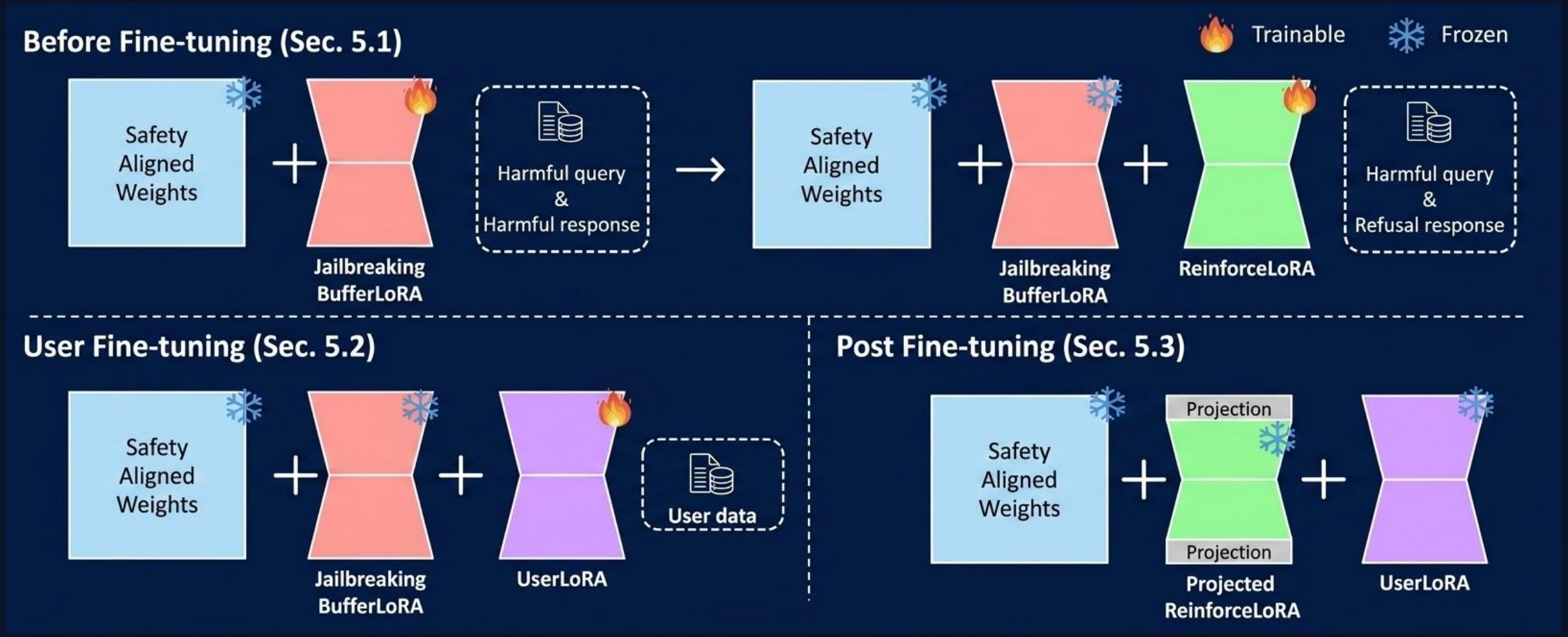
Comparison of 2D Loss Landscape



Comparison of Safety Gradient Score and Gradient Norm

[1] Zhou, Xin, et al. "Making harmful behaviors unlearnable for large language models." Findings of the Association for Computational Linguistics: ACL 2024. 2024.

Buffer-and-Reinforce Framework



Buffer-and-Reinforce Framework

Phase 1: Before Fine-Tuning

FaaS Provider pretrains BufferLoRA and ReinforceLoRA.

- Trains **BufferLoRA** on harmful pairs to **create a jailbreak state**.

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

- Trains **ReinforceLoRA** to **refuse harmful prompts** while in this jailbreak state.

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

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

Phase 2: User Fine-Tuning

When a user uploads data for customization:

- **BufferLoRA** is attached but **frozen** to saturate harmful gradients.
- **UserLoRA** is attached and **trained**, protected from learning harmful knowledge.
- BufferLoRA is detached after user fine-tuning.

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

Buffer-and-Reinforce Framework

Phase 3: Post Fine-Tuning

Integrating safety knowledge **without compromising** user-task performance.

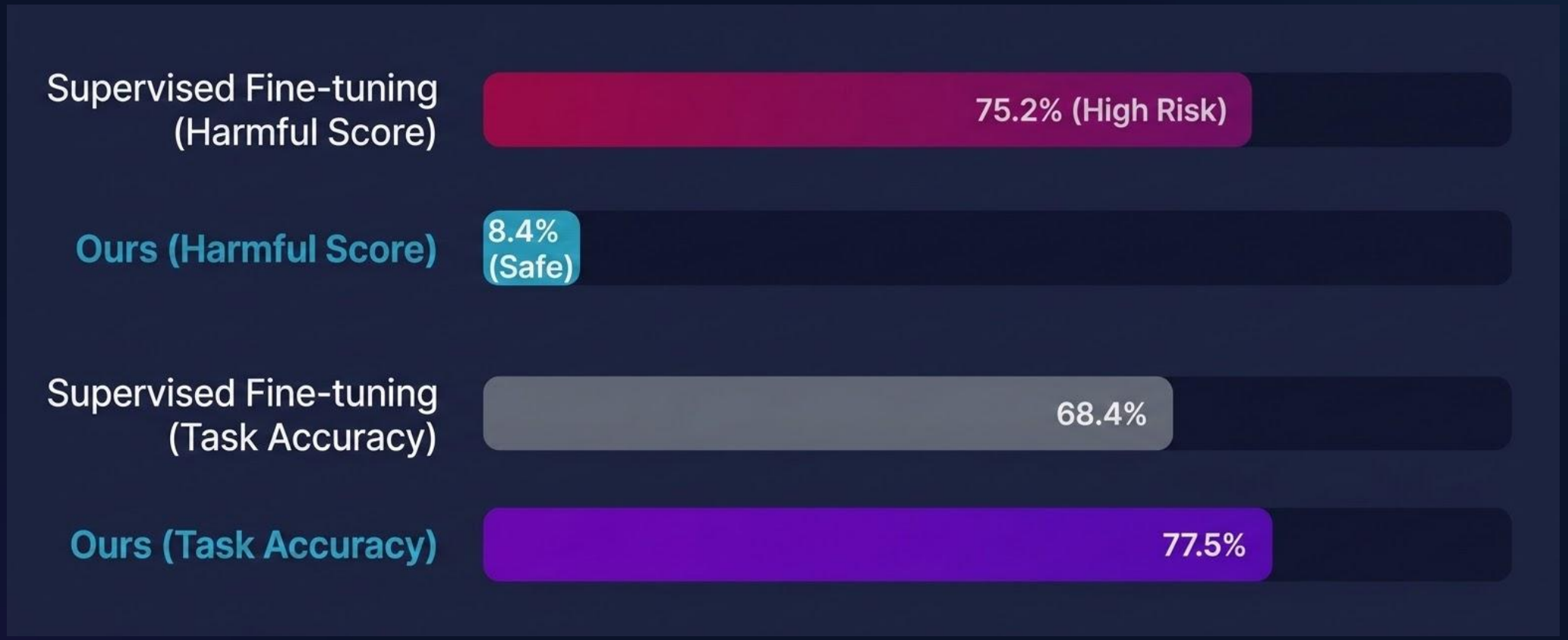
Preventing Task Interference

- Naively merging ReinforceLoRA with UserLoRA destroys downstream utility.
- We use **QR decomposition** to identify the UserLoRA task subspace and then project ReinforceLoRA onto its **orthogonal complement**.
- This **QR-based Merging** reinforces safety while preserving learned utility.

$$Gv_i = \lambda_i v_i, \ \text{s.t.} \ \lambda_i > \tau \cdot \max_j \lambda_j,$$
$$V_{eff} = [v_1, \ldots, v_k] \in \mathbb{R}^{r \times k},$$

$$\hat{B}_U = Q_B R,$$
$$\bar{W}_R = (I - \alpha Q_B Q_B^T) W_R,$$
$$W_{\text{final}} = W_{\text{base}} + \frac{1}{2}(W_U + \bar{W}_R)$$

Safety vs. Utility



*Evaluated on the GSM8K benchmark with a 10% harmful data contamination ratio. The proposed framework aggressively suppresses harmful outputs while actually **improving** overall task accuracy compared to standard SFT.*

Conclusion

Analysis of Temporary Jailbreaking:

Gradient-level analysis proves that temporary jailbreaking naturally saturates safety-degrading gradients, effectively **neutralizing harmful updates** while fully preserving benign, task-relevant learning pathways.

Buffer-and-Reinforce Framework:

Proposed a novel framework deploying **BufferLoRA** to absorb harmful updates during fine-tuning, and integrating **ReinforceLoRA** via QR-based orthogonal merging to **restore safety** post-adaptation.

Safety & Utility Balance:

Achieved **state-of-the-art safety and task utility** across diverse models and tasks. The framework operates with zero additional safety data during user fine-tuning and incurs **minimal computational overhead**.