FEDCURE: MITIGATING SAFETY EROSION IN FULLY BENIGN FEDERATED LLM FINE-TUNING THROUGH CROSS-CLIENT UPDATE CONSENSUS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning aligned large language models (LLMs) on benign data can compromise safety even without malicious intent, but this phenomenon has been studied primarily in centralized settings. We show that safety erosion also arises in federated LLM fine-tuning, even when all participating clients are benign and train exclusively on benign local data. Our analysis reveals a federated-specific signal: erosion-aligned update directions that are consistently shared across clients can become disproportionately represented in the aggregated update. Motivated by this observation, we propose **FedCURE**, a safety-aware federated fine-tuning framework centered on Erosion Consensus Filtering (ECF). ECF identifies erosion-aligned candidates within individual client updates and filters those with strong cross-client consensus before aggregation. FedCURE further employs offline reference adapters to stabilize local adaptation and correct residual safety degradation in the accumulated global model. Experiments across diverse LLM families and downstream datasets demonstrate that FedCURE substantially reduces safety erosion while retaining effective task performance and general utility.

1 INTRODUCTION

The rise of large language models (LLMs) as general-purpose foundation models has enabled powerful capabilities across diverse real-world applications (Li et al., 2024). Adapting these models to downstream requirements often involves fine-tuning on task-specific datasets (Ouyang et al., 2022; Chung et al., 2024). However, prior work has shown that even fine-tuning on benign data without any malicious intent does not necessarily preserve the model's safety properties (Qi et al., 2024; He et al., 2024; Guan et al., 2025; Kim et al., 2025). This highlights that benign training data and legitimate adaptation objectives alone do not ensure safety preservation (He et al., 2024; Kim et al., 2025).

Safety preservation during fine-tuning has received substantial attention in centralized settings. Prior work has shown that safety behavior can be brittle; changes along a small subset of parameter directions can weaken refusal behavior (Wei et al., 2024; Pan et al., 2025). While various safety-preserving methods have been proposed (Huang et al., 2024b; 2025b; Zhao et al., 2025b; Hsu et al., 2024; Li et al., 2025; Zhang et al., 2026; Huang et al., 2025a) and the geometry underlying safety degradation has been increasingly explored (Kim et al., 2025; Bach et al., 2026; Springer et al., 2026), safety-relevant subspaces often overlap with task-relevant representations (Ponkshe et al., 2026). This entanglement makes coarse parameter freezing or removal undesirable and motivates selective interventions during adaptation.

However, centralized fine-tuning is not always feasible when training data is distributed among users that cannot share their local data. Federated learning (FL) (McMahan et al., 2017), combined with parameter-efficient fine-tuning (PEFT), enables collaborative LLM adaptation without centralizing client data (Zhang et al., 2024; Wang et al., 2024; Cho et al., 2024; Fang et al., 2026). Among various research directions, safety has emerged as a key concern in FL, with existing work largely focusing on explicit threat models such as malicious clients (Ye et al., 2025; Tao et al., 2026), Byzantine behavior (Blanchard et al., 2017), or poisoned local data (Zhu et al., 2026). Yet, safety erosion in federated LLM fine-tuning in the absence of such adversarial factors, i.e., when all participating clients are benign and fine-tune exclusively on benign data, remains largely unexplored.

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.

Challenges. Our findings show that safety erosion can arise in federated fine-tuning even without any malicious clients or poisoned local data, as illustrated in Figure 1. Unlike centralized fine-tuning, federated fine-tuning repeatedly aggregates locally updated client models into a shared global model, which is then redistributed as the starting point for subsequent local training. Meanwhile, the server has limited visibility into the safety-related changes introduced during local adaptation. This poses two main challenges. First, safety-eroding components must be identified within legitimate client updates under limited server observability. As these changes arise from benign local adaptation, they do not necessarily appear as anomalous clients or outlying updates. Second, safety erosion must be prevented from propagating across rounds, as surviving changes become part of the next global initialization and may accumulate through repeated local adaptation and aggregation. These challenges motivate complementary client- and server-side interventions to mitigate safety erosion in FL. This leads to our central question:

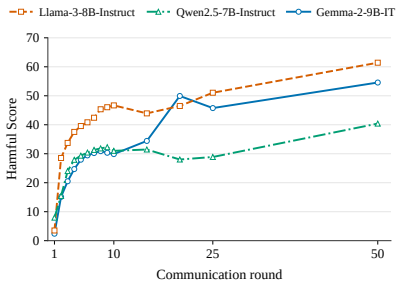


Figure 1: Harmful score for three different models on Dolly dataset. As communication rounds proceed, persistent safety erosion occurs during federated fine-tuning.

How can we characterize and mitigate safety erosion across repeated rounds of federated fine-tuning under fully benign conditions, while retaining task performance?

Contributions. To address these challenges, we propose **FedCURE**, a **Fed**erated **C**onsensus-guided **U**pdate **R**efinement framework against safety **E**rosion, designed to preserve safety during both client-side local update and server-side aggregation. Our key idea is to exploit *consensus update geometry*: We identify safety-erosion-related directions that are aligned with an erosion reference and consistently shared across clients, allowing us to selectively target erosion-related structure without broadly suppressing task-relevant updates. FedCURE combines this consensus-based filtering with offline reference adapters that stabilize local adaptation and the accumulated global model. Specifically, FedCURE comprises two complementary components:

- **Erosion Consensus Filtering (ECF).** ECF is a server-side filtering process applied to client marginal updates. It first identifies erosion-aligned candidate subspaces within each client update using an offline erosion reference. It then finds high-consensus directions shared across clients and selectively filters their contribution from the aggregated marginal update before it is incorporated into the global model.
- **Reference-Guided Safety Stabilization.** We employ two offline reference adapters to anchor federated adaptation: an Erosion Reference Adapter (ERA) that conditions local update generation while serving as the erosion reference for ECF, and a Reinforce Reference Adapter (RRA) that projects and merges safety corrections into the accumulated global model before redistribution.

Experiments across diverse LLM families and datasets demonstrate that FedCURE substantially reduces safety erosion while retaining effective task performance and general utility. Ablation studies further validate the contribution of each stage and the benefit of consensus-based selection. To the best of our knowledge, FedCURE is the first work to systematically characterize and mitigate safety erosion under *fully benign* federated LLM fine-tuning.

2 Related Works

Safety Preservation in Centralized Fine-Tuning. Safety algorithms in this setting intervene at different stages of adaptation; alignment-stage methods improve robustness to subsequent fine-tuning (Huang et al., 2024b; 2025b; Zhao et al., 2025b); training-time methods incorporate safety supervision or constrain updates through safety-relevant operators and geometric regularization (Bianchi et al., 2024; Zhou et al., 2024; Huang et al., 2024a; Hsu et al., 2024; Li et al., 2025; Yang et al., 2025; Zhang et al., 2026); and post-training methods restore alignment through projection, perturbations, pruning and subspace-fusion corrections (Wang et al., 2025; Huang et al., 2025a; Zhou et al., 2025; Bach et al., 2026). The reference-based view of safety-relevant parameters explored in prior work (Zhou et al., 2024; Ham et al., 2026) further motivates our use of offline reference adapters for local stabilization. However, in federated fine-tuning, limited server observability into clients' optimization trajectories restricts the direct transfer of centralized interventions, which fail to

consistently retain safety-utility trade-offs (Section 5.1), motivating approaches specifically tailored to federated settings.

Federated Fine-Tuning under Explicit Threats. Federated fine-tuning enables decentralized clients to collaboratively adapt LLMs without pooling their data (Zhang et al., 2024; Ye et al., 2024b), and recent benchmarks further characterize this setting under heterogeneous client data distribution (Ye et al., 2024a; Gao et al., 2026). However, existing studies of safety in federated LLM fine-tuning have primarily assumed explicit threats; Byzantine-robust aggregation targets unreliable or adversarial updates (Blanchard et al., 2017; Yin et al., 2018), while FedLLM-specific approaches consider malicious clients that intentionally compromise model safety (Ye et al., 2025; Tang et al., 2026), detect malicious updates before aggregation (Tao et al., 2026), or sanitize unsafe local training content (Zhu et al., 2026). In contrast, FedCURE studies a distinct benign setting in which all participating clients are non-malicious, yet safety can still degrade. To the best of our knowledge, FedCURE is the first work to systematically characterize and mitigate safety erosion in fully benign federated LLM fine-tuning, without assuming any malicious clients or poisoned local data.

Representational and Geometric Perspectives on Model Safety. Recent studies increasingly characterize model safety through structured model geometry. Prior work showed that small or low-rank modifications can alter alignment (Wei et al., 2024; Arditi et al., 2024; Pan et al., 2025), while subsequent work revealed richer structures involving higher-dimensional concepts, and safety-relevant representations entangled with other model functions (Wollschläger et al., 2025; Zhao et al., 2025a; Ponkshe et al., 2026). Fine-tuning can further reshape this geometry along optimization-dependent trajectories leading to safety erosion (Peng et al., 2024; Kim et al., 2025; Springer et al., 2026; Bach et al., 2026). These findings motivate geometry-aware safety preservation. In federated fine-tuning, this challenge is further complicated by heterogeneous client updates induced by non-IID data, which induces heterogeneous update geometries across clients. FedCURE addresses this by analyzing cross-client update geometry and leveraging consensus to target erosion-related structure consistently shared across heterogeneous client updates.

3 Understanding Safety Erosion via Cross-Client Consensus

Problem Setup. We consider federated low-rank adaptation (LoRA) (Hu et al., 2022) fine-tuning where clients locally adapt a shared model W_0, without disclosing their raw data or intermediate gradients. At round t, participating clients $\mathcal{I}_t$ initialize rank-r LoRA adapters from the previous broadcast and obtain $\Delta W_{i,t} = sB_{i,t}A_{i,t}$, where $s = \alpha/r$ is the corresponding LoRA scaling factor. The server aggregates the LoRA factors as

$$B_t = \sum_{i\in\mathcal{I}_t} p_{i,t}B_{i,t}, \qquad A_t = \sum_{i\in\mathcal{I}_t} p_{i,t}A_{i,t}, \qquad \Delta W_t = sB_tA_t, \tag{1}$$

where $p_{i,t}$ denotes the sample proportion of client i. We assume that all clients and their local data are benign, with no malicious clients or poisoned local data.

We define $D_{i,t} := \Delta W_{i,t} - \Delta W_{t-1}$ as the marginal update of client i and $D_t := \Delta W_t - \Delta W_{t-1}$ as the committed server marginal. Since safety erosion emerges across communication rounds even under benign fine-tuning (Figure 1), we ask which components of heterogeneous client marginals are consistently committed to the evolving global update.

Cross-Client Consensus as a Commitment Signal. Cross-client consensus may help identify components that are strongly represented in the committed server marginal. We examine whether erosion-aligned components shared across clients are disproportionately represented after factor-wise aggregation. For a unit output direction u with nonzero weighted client-marginal energy, we define its *commitment score* as

$$\kappa_t(u) := \frac{\|u^\top D_t\|_2^2}{\sum_{i\in\mathcal{I}_t} p_{i,t}\|u^\top D_{i,t}\|_2^2}. \tag{2}$$

The score measures server-marginal energy relative to weighted client-marginal energy along the same direction and need not lie in $[0, 1]$. Under factor-wise aggregation, D_t need not equal $\bar{D}_t := \sum_{i\in\mathcal{I}_t} p_{i,t}D_{i,t}$; we therefore evaluate the score using the stored server marginal directly. As a reference, under linear averaging a component contributed only by client i yields $\kappa_t(u) = p_{i,t}$, whereas one shared identically across clients yields $\kappa_t(u) = 1$; larger scores thus indicate stronger commitment of u to the server marginal.

Erosion-Aligned Consensus Modes Concentrate Server Marginal Energy. We examine the retained FedIT (Zhang et al., 2024) checkpoints for Llama-3-8B-Instruct fine-tuned on Dolly at FL rounds 5 and 10. We use an independently trained erosion reference adapter W_E, obtained from 5K harmful prompts paired with compliant responses, to identify directions associated with safety erosion. For each client marginal, we partition its leading output directions according to the sign of their alignment with W_E. We denote positively and negatively aligned candidates by Q^+ and Q^-, respectively, and measure how consistently these directions are shared across clients through the spectrum of their weighted projector mean. Modes with spectral support $\lambda \geq 0.5$ are regarded as *consensus modes*; the formal construction is provided in Section 4.2.

Table 1: Shared modes of W_E-positive/negative candidates on Dolly. The commitment score is computed by summing projected server energy and weighted client energy separately over selected modes and modules, then taking their ratio.

Rnd	Cand.	Shared Modes (%)	Server Energy (%)	Client Energy (%)	Commitment Score
5	Q^+	0.76	17.5	4.7	0.67
	Q^-	0.09	0.1	0.1	0.22
10	Q^+	1.52	62.8	16.6	0.78
	Q^-	0.23	0.5	0.2	0.47

Table 1 reveals a clear asymmetry. At round 10, W_E-positive consensus modes (Q^+) constitute only 1.52% of the candidate modes, yet account for 62.8% of the committed server-marginal energy, compared with 16.6% of the weighted client-marginal energy. Their pooled commitment score is 0.78. In contrast, W_E-negative consensus modes (Q^-) account for only 0.5% of the server-marginal energy. Thus, in these checkpoints, the selected erosion-aligned consensus modes are disproportionately represented in the server marginal relative to their share of the weighted client-marginal energy.

Behavioral Validation of Shared Erosion-Aligned Modes. The analysis above shows that erosion-aligned consensus modes are disproportionately represented in the global update, but this geometric pattern alone does not establish their effect on model safety. To assess their behavioral relevance, we remove the shared W_E-positive component from retained checkpoints and compare it with energy-matched geometric controls. As shown in Figure 2, removing the shared W_E-positive component reduces the harmful score by approximately 8%pt at both rounds, while task performance and general utility remain largely preserved. In contrast, removing other directions with a comparable intervention magnitude (e.g., Q^- modes, random directions, and top-singular directions with matched energy) produces much smaller reductions in harmful score. This suggests that the safety improvement is specific to the shared erosion-aligned component, rather than simply due to removing a large update direction.

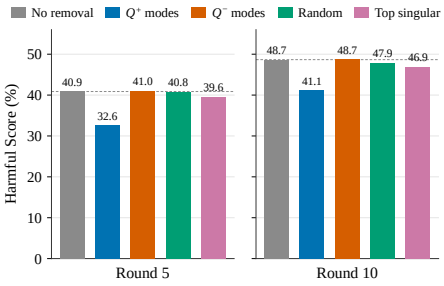


Figure 2: Harmful score after one-shot interventions at rounds 5 and 10.

Together, the geometric and behavioral evidence motivates our Erosion Consensus Filtering (ECF), which targets shared erosion-aligned components that are consistently shared across clients.

4 FedCURE Algorithm

Overview of Approach. Motivated by our findings in Section 3, we propose **FedCURE**, a federated fine-tuning framework centered on *Erosion Consensus Filtering* (ECF) and supported by two frozen reference adapters. FedCURE operates across both client-side and server-side stages of federated adaptation. On the client-side, the *Erosion Reference Adapter* (ERA) conditions local training to stabilize update generation. On the server-side, ERA further provides the erosion reference for ECF, which identifies erosion-aligned directions consistently shared across clients and selectively filters their contribution from the aggregated update. Finally, the server applies the *Reinforce Reference Adapter* (RRA) to correct remaining safety degradation in the accumulated global adapter before redistribution to clients. Figure 3 shows the overview of FedCURE. We next describe the detailed procedure of FedCURE, beginning with a prestage that initializes the reference adapters.

Prestage. Before federated fine-tuning begins, the server constructs ERA and RRA through supervised fine-tuning (SFT) on the frozen base model W_0, using the same LoRA parameterization and target modules as the client adapters. ERA is trained on harmful prompts paired with compliant responses

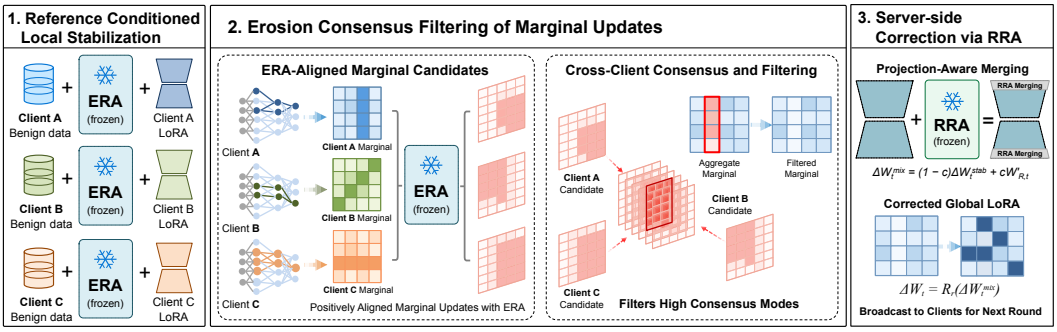


Figure 3: Overview of **FedCURE** framework. ERA conditions client-side local adaptation, ECF filters shared erosion-aligned components from the aggregate marginal update, and RRA corrects safety degradation in the cumulative global adapter.

to provide the erosion-oriented reference, whereas RRA is trained on harmful prompt-refusal pairs together with utility datasets to support safety restoration. The effective weight updates of ERA and RRA are $W_E = s_E B_E A_E$ and $W_R = s_R B_R A_R$, respectively, where s_E and s_R denote the corresponding LoRA scaling factors. ERA is retained on the server and distributed to clients, while RRA remains solely on the server.

4.1 Reference-Conditioned Local Stabilization

FedCURE first stabilizes local adaptation to mitigate safety erosion, by conditioning client training on the frozen ERA W_E. At each round, the server broadcasts the current global LoRA modules (B_{t-1}, A_{t-1}) to participating clients. Each client trains the received LoRA modules on its local data, while both W_0 and W_E remain fixed. Let ϕ denote the trainable client LoRA parameters, and let $\Delta W(\phi) = \{sB_\ell A_\ell\}_\ell$ denote their effective weight updates across layers. Each client i minimizes

$$\mathcal{L}_i^{\mathcal{C}}(\phi) = \frac{1}{\bar{T}_i}\,\mathbb{E}_{(x,y)\sim\mathcal{D}_i^{\text{task}}}\left[-\log p_{W_0+W_E+\Delta W(\phi)}(y \mid x)\right], \tag{3}$$

where $\bar{T}_i = \mathbb{E}_{\mathcal{D}_i^{\text{task}}}[|y|]$ is the average response length over client i's local data $\mathcal{D}_i^{\text{task}}$.

Although ERA (W_E) is frozen, it remains active during local optimization. Consequently, client gradients are evaluated at the ERA-conditioned model state, allowing ERA to guide local adaptation without introducing an additional objective. After local fine-tuning, client i uploads only its updated LoRA modules $(B_{i,t}, A_{i,t})$ to the server.

4.2 Erosion Consensus Filtering of Marginal Updates

After receiving client LoRA modules, the server obtains each client i's marginal $D_{i,t} = \Delta W_{i,t} - \Delta W_{t-1}$. ECF then identifies erosion-related components within these marginals, selects those consistently supported across clients, and filters their contribution from the aggregate marginal D_t.

ERA-Aligned Marginal Candidates. A key point is that each client marginal contains both task-related and erosion-related changes. Therefore, ECF first restricts its search to directions positively aligned with the erosion reference W_E. For each layer, let $V_{\Delta i,t} \in \mathbb{R}^{d_{\text{out}}\times k_{i,t}}$ contain up to r leading left singular vectors of $D_{i,t}$. Then, the server constructs

$$\begin{aligned} K_{i,t} &= \text{sym}\big((V_{\Delta i,t})^\top D_{i,t} W_E^\top V_{\Delta i,t}\big), \\ Q_{i,t}^+ &= V_{\Delta i,t} O_{i,t}^+ \quad \text{where} \quad K_{i,t} O_{i,t}^+ = O_{i,t}^+ \Lambda_{i,t}^+ \;\; (\Lambda_{i,t}^+ \succ 0), \end{aligned} \tag{4}$$

where $\text{sym}(X) = (X + X^\top)/2$. Selecting the positive eigenspace ensures that every non-zero direction in $\text{col}(Q_{i,t}^+)$ has positive signed alignment with W_E. Thus, $Q_{i,t}^+$ defines the erosion-aligned candidate space used for subsequent cross-client consensus estimation and filtering.

Cross-Client Consensus and Filtering. ECF uses cross-client agreement to select erosion-aligned candidate directions that are consistently supported across clients. To quantify the consensus, the server forms the weighted projector mean of the ERA-aligned candidate projectors:

$$C_t^\Delta = \sum_{i\in\mathcal{I}_t} p_{i,t} Q_{i,t}^+ (Q_{i,t}^+)^\top. \tag{5}$$

Since C_t^{Δ} is a weighted average of orthogonal projectors, it is symmetric positive semidefinite and satisfies $0 \preceq C_t^{\Delta} \preceq I$, implying that all its eigenvalues lie in $[0, 1]$. For an eigenvector satisfying $C_t^{\Delta} u_{j,t} = \lambda_{j,t} u_{j,t}$ and $|u_{j,t}|_2 = 1$, we have

$$\lambda_{j,t} = u_{j,t}^{\top} C_t^{\Delta} u_{j,t} = \sum_{i \in \mathcal{I}_t} p_{i,t} \|(Q_{i,t}^{+})^{\top} u_{j,t}\|_2^2.$$

Thus, $\lambda_{j,t} \in [0, 1]$ measures the weighted cross-client support for direction $u_{j,t}$, which captures marginal components positively aligned with W_E. We set the consensus threshold to $\tau = 0.5$ and let $Q_{H,t}$ collect as columns the orthonormal eigenvectors $u_{j,t}$ satisfying $\lambda_{j,t} \geq \tau$. The server constructs the corresponding orthogonal projector and filters the aggregate marginal:

$$P_{H,t} = Q_{H,t} Q_{H,t}^{\top}, \qquad \widetilde{D}_t = (I - P_{H,t}) D_t. \tag{6}$$

As observed in Section 3, erosion-aligned consensus modes become disproportionately prominent after aggregation, accounting for a much larger share of the server-marginal energy than of the weighted client-marginal energy. ECF combines ERA alignment with cross-client support to filter such shared contributions from each round's aggregate marginal, aiming to limit their repeated accumulation in the global state.

The filtered marginal is then added to the previously broadcast cumulative adapter, and the result is retracted to rank at most r:

$$\widetilde{\Delta W}_t = \Delta W_{t-1} + \widetilde{D}_t, \qquad \Delta W_t^{\mathrm{stab}} = \mathcal{R}_r(\widetilde{\Delta W}_t), \tag{7}$$

where $\mathcal{R}_r$ denotes SVD-based retraction to rank at most r. Before rank retraction, ECF preserves the previous cumulative update along the selected directions while leaving the orthogonal component of the aggregate marginal unchanged.

4.3 Server-side Correction: Cumulative Correction via RRA Merging

ECF limits the incorporation of erosion-related components from the current round, but it does not directly correct safety degradation already accumulated in the global adapter. To complement ECF, FedCURE applies the frozen RRA W_R as a post-aggregation correction to restore safety while limiting interference with aggregated task information. The server reduces the overlap of W_R with the dominant subspace of the stabilized adapter $\Delta W_t^{\mathrm{stab}}$. Let $P_{T,t} = Q_{T,t} Q_{T,t}^{\top}$, where $Q_{T,t}$ collects the left singular vectors of $\Delta W_t^{\mathrm{stab}}$ whose singular values exceed $\tau_b \sigma_{\max}(\Delta W_t^{\mathrm{stab}})$ for $\tau_b \in [0, 1]$. The reference update is then projected and attenuated as:

$$W'_{R,t} = (I - \alpha_b P_{T,t}) W_R, \qquad 0 \leq \alpha_b \leq 1, \tag{8}$$

where α_b controls the attenuation along this task subspace. Then the server blends the projected reference with the stabilized cumulative adapter and applies rank-r truncated SVD:

$$\Delta W_t^{\mathrm{mix}} = (1 - c) \Delta W_t^{\mathrm{stab}} + c W'_{R,t}, \qquad \Delta W_t = \mathcal{R}_r(\Delta W_t^{\mathrm{mix}}), \qquad 0 \leq c \leq 1, \tag{9}$$

where $c \in [0, 1]$ controls the correction strength. The resulting LoRA modules (B_t, A_t), satisfying $\Delta W_t = s B_t A_t$, are broadcast to clients for round $t + 1$. Appendix E analyzes the correction and its interaction with ECF.

4.4 Overall FedCURE Procedure

Together, the complementary components described above constitute FedCURE. Before training begins, the server constructs the frozen ERA and RRA once in the prestage. At round t, the server broadcasts the current global LoRA modules together with ERA to participating clients. Each client performs ERA-conditioned local adaptation on its data and uploads only its updated LoRA modules. The server then applies ECF, using ERA to identify erosion-aligned candidates and cross-client consensus. Then, the RRA provides a complementary correction for residual safety degradation. The resulting LoRA modules are then broadcast as the initialization for the next round. In this way, ERA stabilizes local update generation, ECF selectively controls shared erosion-related changes during aggregation, and RRA corrects residual degradation in the accumulated global state.

Table 2: Evaluation on Meta-Llama-3-8B-Instruct across Commonsense and Dolly. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Dolly							Commonsense						
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	48.67	0.00	39.83	0.00	58.93	0.00	0.00	25.10	0.00	85.39	0.00	49.27	0.00	0.00
Multi-Krum	44.89	−3.78	39.19	−0.64	60.29	+1.36	+4.50	22.24	−2.86	84.94	−0.45	48.18	−1.09	+1.32
Trimmed Mean	50.35	+1.68	39.19	−0.64	59.69	+0.76	−1.56	24.67	−0.43	85.58	+0.19	47.19	−2.08	−1.46
SaLoRA	34.80	−13.87	39.04	−0.79	60.85	+1.92	+15.00	13.80	−11.30	86.01	+0.62	45.45	−3.82	+8.10
SafeLoRA	45.67	−3.00	39.63	−0.20	60.31	+1.38	+4.18	11.74	−13.36	84.94	−0.45	49.59	+0.32	+13.23
GuardSpace	11.19	−37.48	37.88	−1.95	59.48	+0.55	+36.08	37.95	+12.85	35.85	−49.54	25.77	−23.50	−85.89
Booster	44.47	−4.20	38.94	−0.89	59.98	+1.05	+4.36	18.90	−6.20	84.65	−0.74	52.33	+3.06	+8.52
FedLLM-Attack	11.90	−36.77	31.31	−8.52	56.35	−2.58	+25.67	17.43	−7.67	76.06	−9.33	56.16	+6.89	+5.23
FedCURE (ours)	2.03	−46.64	38.62	−1.21	68.21	+9.28	**+54.71**	1.58	−23.52	84.36	−1.03	67.54	+18.27	**+40.76**

Table 3: Evaluation on the Dolly dataset across Gemma-2-9B-it and Qwen2.5-7B-Instruct. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Gemma-2-9B-it							Qwen2.5-7B-Instruct						
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	30.53	0.00	40.13	0.00	67.34	0.00	0.00	31.98	0.00	38.26	0.00	68.28	0.00	0.00
Multi-Krum	29.57	−0.96	40.00	−0.13	68.50	+1.16	+1.99	26.69	−5.29	38.24	−0.02	67.69	−0.59	+4.68
Trimmed Mean	33.89	+3.36	40.99	+0.86	68.17	+0.83	−1.67	31.25	−0.73	38.02	−0.24	67.77	−0.51	−0.02
SaLoRA	18.96	−11.57	40.26	+0.13	69.15	+1.81	+13.51	17.01	−14.97	37.41	−0.85	67.75	−0.53	+13.59
SafeLoRA	32.07	+1.54	40.24	+0.11	67.94	+0.60	−0.83	30.48	−1.50	39.50	+1.24	70.21	+1.93	+4.67
GuardSpace	1.91	−28.62	38.74	−1.39	70.56	+3.22	+30.45	12.97	−19.01	36.60	−1.66	64.33	−3.95	+13.40
Booster	30.83	+0.30	40.83	+0.70	66.68	−0.66	−0.26	31.16	−0.82	37.90	−0.36	67.53	−0.75	−0.29
FedLLM-Attack	8.98	−21.55	33.33	−6.80	59.24	−8.10	+6.65	9.13	−22.85	29.50	−8.76	63.26	−5.02	+9.07
FedCURE (ours)	1.84	−28.69	39.62	−0.51	75.20	+7.86	**+36.04**	3.23	−28.75	33.36	−4.90	75.94	+7.66	**+31.51**

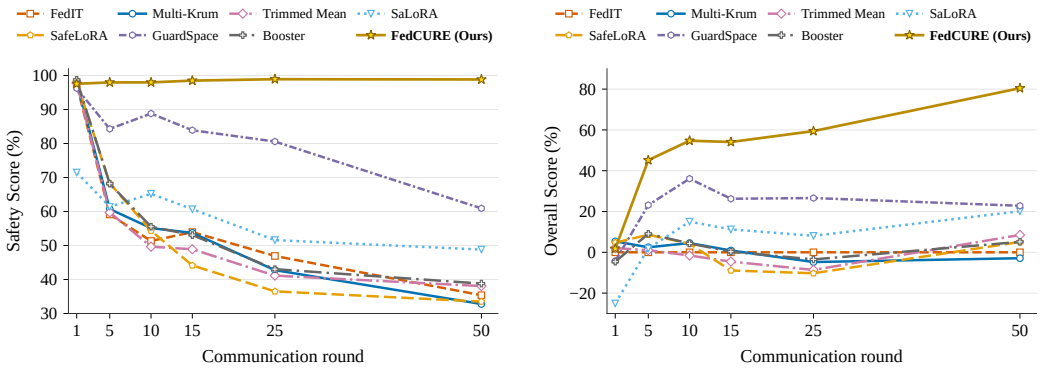
5 EXPERIMENTS

Our experiments address four main questions: (i) how FedCURE compares with federated and safety-preserving baselines across datasets and model families; (ii) whether its safety gains persist over extended federated fine-tuning; (iii) whether consensus-based selection in ECF is more effective than generic attenuation of the aggregated update; and (iv) how ERA and RRA complement ECF.

Models and Datasets. We evaluate three instruction-tuned model families: Meta-Llama-3-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-it with LoRA (Hu et al., 2022) applied to all linear decoder layers. For federated fine-tuning, we use Dolly-15K (Conover et al., 2023), a suite of eight commonsense reasoning tasks (Hu et al., 2023), and WizardLM (Xu et al., 2024). ERA is constructed using harmful prompt-compliant response pairs from Circuit Breakers (Zou et al., 2024), whereas RRA is constructed on harmful prompt-refusal response pairs from Circuit Breakers and instruction-response pairs from the Alpaca dataset. We use eight fully participating clients with dataset-specific partitions, where the results with more clients are provided in Appendix G. Full training and data-partition details are provided in Appendix A.

Baselines. We compare FedCURE against standard federated fine-tuning (FedIT) (Zhang et al., 2024), Byzantine-robust aggregation (Multi-Krum (Blanchard et al., 2017) and Trimmed Mean (Yin et al., 2018)), centralized safety-preserving methods(SaLoRA (Li et al., 2025), SafeLoRA (Hsu et al., 2024), GuardSpace (Zhang et al., 2026), Booster (Huang et al., 2025b)), and the post-hoc defense from FedLLM-Attack (Ye et al., 2025). Implementation details are provided in Appendix B.

Metrics. We evaluate safety using harmful score (HS), aggregated over multiple harmful prompt benchmarks (Zou et al., 2023; Röttger et al., 2024; Souly et al., 2024; Mazeika et al., 2024) using LLM classifiers (Han et al., 2024; Mazeika et al., 2024). Task performance is measured by Rouge-L F1 for Dolly, accuracy for commonsense reasoning, and MT-Bench for WizardLM. General utility is measured by the average performance on MMLU (Hendrycks et al., 2021) and IFEval (Zhou et al., 2023). All methods are evaluated by applying the resulting global adapter to W_0. Detailed evaluation protocols are provided in Appendix A.



(a) Safety score progression

(b) Overall score progression

Figure 4: Safety and overall performance on the Dolly dataset with Meta-Llama-3-8B-Instruct across communication rounds: (a) safety score (100 − Harmful Score), illustrating persistent safety erosion during federated fine-tuning; (b) overall score relative to FedIT (Δ_{Overall}), capturing the cumulative trade-off. Unlike baselines, FedCURE consistently preserves safety while sustaining superior overall trade-offs throughout training.

Other Results. To assess robustness beyond the main setting, we conduct additional experiments under different degrees of data heterogeneity, larger number of participating clients, and a larger-scale model. We also study the effect of ERA-alignment in consensus. The corresponding results are reported in Appendix G. Computational cost results are provided in Appendix C.

5.1 Main Experimental Results

Safety and Task Performance. We first evaluate whether FedCURE consistently preserves safety while maintaining task performance and general utility across diverse downstream adaptation tasks. In Table 2, FedCURE substantially suppresses safety erosion across different datasets, while achieving competitive task performance and general utility. Consequently, FedCURE yields the highest overall score among all evaluated baselines across these heterogeneous benchmarks. Crucially, this advantage is not confined to a specific architecture; on both Gemma-2-9B-it and Qwen2.5-7B-Instruct in Table 3, FedCURE also achieves the lowest harmful score, while consistently achieving the highest Δ_{Overall}. These empirical findings confirm that FedCURE effectively prevents safety erosion across both diverse task domains and distinct model families without compromising the functional capabilities acquired through benign federated training. Further experiments with diverse datasets, model architectures, parameter scale, and data distributions are in Appendix G.

Trajectories across Communication Rounds. We next examine whether safety gains of FedCURE persist over extended federated fine-tuning. Figure 4 shows that FedCURE sustains high safety scores (100 − HS) into the later communication rounds while maintaining the highest Δ_{Overall} throughout the evaluated training horizon. In contrast, the evaluated baselines generally exhibit increasing harmful scores as communication proceeds, failing to sustain comparable safety protection. These trajectories suggest that centralized safety-preserving interventions offer limited protection when changes introduced by local fine-tuning are repeatedly aggregated and inherited as the initialization for subsequent rounds. Meanwhile, Byzantine-robust aggregation primarily targets anomalous or adversarial client updates rather than erosion-related components shared across otherwise benign updates. Together, these results highlight safety erosion in fully benign federated fine-tuning as a challenge requiring protection across repeated local adaptation and aggregation, and show that FedCURE's safety gains are sustained rather than confined to the early stages of training.

5.2 Further Analyses

Complementary Roles of FedCURE Components. Table 4 highlights the synergy between ERA, ECF, and RRA. ERA and RRA provide safety benefits even without ECF, while combining all three further improves safety with competitive task performance and general utility. ERA conditions the local updates subsequently processed by ECF, whereas RRA complements consensus-guided filtering through global safety restoration. Together, these results support integrating local

Table 4: Component ablations on Meta-Llama-3-8B-Instruct fine-tuned on Commonsense benchmarks. Deltas are relative to FedIT; $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

Method	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	25.10	0.00	85.39	0.00	49.27	0.00	0.00
ECF-only	9.12	−15.98	85.51	+0.12	54.18	+4.91	+21.01
ERA-only	6.41	−18.69	84.99	−0.40	53.32	+4.05	+22.34
RRA-only	3.23	−21.87	84.71	−0.68	62.44	+13.17	+34.37
ERA–RRA	3.38	−21.72	84.79	−0.60	64.56	+15.29	+36.41
ECF–RRA	1.69	−23.41	84.46	−0.93	67.08	+17.81	+40.29
FedCURE	1.58	−23.52	84.36	−1.03	67.54	+18.27	**+40.76**

stabilization, selective update filtering, and global restoration to balance safety, task performance, and utility. Additional details are provided in Appendix G.1.

Analysis of ECF Design. ECF builds on two core design choices: restricting client candidates to positively ERA-aligned directions and filtering modes with strong cross-client consensus. ERA alignment conditions the search toward potentially erosion-related geometry, while consensus identifies directions shared across clients. To evaluate these core idea, we compare alternative filtering policies on Dolly dataset (Figure 5), without client-side ERA stabilization and RRA correction. Random targets random directions in the marginal update, Uniform attenuates the entire marginal, and Negative-all removes the union of ERA-negative client candidate spaces. Low-consensus instead uses the same ERA-aligned candidate construction as ECF but filters weakly shared modes, whereas ECF targets strongly shared modes. ECF achieves the lowest harmful score among these policies. Further, targeted ablations show that consensus without ERA alignment degrades task performance, suggesting that the relevant signal is cross-client consensus conditioned on erosion-related geometry. Additional details are provided in Appendix G.2.

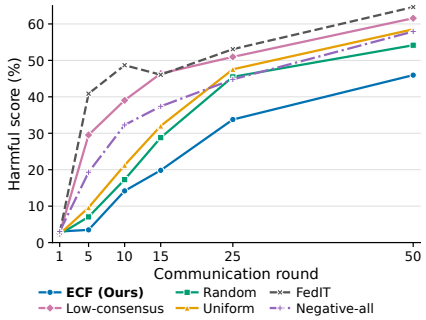


Figure 5: Ablation of subspace selection strategies in ECF on Meta-Llama-3-8B-Instruct which fine-tuned on Dolly dataset. Only filtering directions with high cross-client consensus within the ERA-aligned subspace effectively suppresses safety erosion across rounds.

Over-Refusal. Table 5 reports over-refusal for FedCURE and representative baselines to estimate whether lower harmful scores come at the expense of increased refusal even on harmless prompts. Compared with other baselines, FedCURE substantially reduces harmful scores while incurring a modest change in over-refusal. These results indicate that safety gains of FedCURE are not accompanied by a systematic increase in over-refusal; detailed results are provided in Appendix G.5

Table 5: Over-refusal (%; lower is better) of Meta-Llama-3-8B-Instruct, measured on 250 harmless XSTest prompts.

Method	Commonsense	Dolly	WizardLM	Average
SaLoRA	42.40	1.20	26.40	23.33
Booster	10.40	4.80	22.40	12.53
FedLLM-Attack	14.40	16.80	20.00	17.06
FedCURE	5.60	4.40	17.20	**9.06**

Robustness to the Erosion Reference Data. We further evaluate the sensitivity of FedCURE to the reference dataset by replacing the Circuit Breakers data with the LAT dataset (Sheshadri et al., 2025). As shown in Table 6, FedCURE with LAT-based reference adapters yields results largely consistent with the original configuration, suppressing safety erosion while preserving task competence. This indicates that ECF does not rely on specific harmful samples, but on the subspace along which W_0 is degraded in terms of safety. Detailed experimental configurations and extended results are provided in Appendix G.4.

Table 6: Ablations with LAT reference adapters for Meta-Llama-3-8B-Instruct on Dolly dataset.

Method	HS ↓	Task ↑	Utility ↑
FedIT	48.67	39.83	58.93
FedCURE (Circuit-Based)	2.03	38.62	68.21
FedCURE (LAT-based)	2.33	36.70	69.77

Computational Cost. FedCURE constructs ERA and RRA only once prior to federated fine-tuning, while introducing on overall per-round training delay of approximately 20% relative to standard FedIT on an NVIDIA B200 GPU. This computation overhead is to eliminate safety erosion while preserving task performance and general utility. We provide a detailed runtime analysis in Appendix C.

6 Conclusion and Discussion

In this work, we identified safety erosion as a previously overlooked challenge in fully benign federated LLM fine-tuning, and showed that it can emerge even without malicious clients or poisoned local data. We proposed FedCURE, a framework centered on ECF that exploits shared structure in client marginal updates. ECF identifies erosion-aligned directions with strong cross-client consensus and selectively filters their contribution from the aggregated update. FedCURE further combines this filtering with ERA-based local stabilization and RRA-based global correction. Experiments across different models and datasets show that FedCURE substantially improves safety while maintaining competitive task performance and general utility. Future work may extend this perspective to heterogeneous LoRA ranks, asynchronous updates, and more diverse domain-specific settings.

AI USE STATEMENT

In this work, we used generative AI tools to assist with reviewing and refining research experimental design, implementing and debugging research code, interpreting experimental results, and deriving and verifying mathematical claims and theoretical proofs. We also used generative AI tools for literature discovery and analysis, brainstorming, and drafting and editing portions of the manuscript. All AI-assisted suggestions, code, analyses, and written content were independently reviewed and verified by the authors. We take full responsibility for the final content of this work, including all technical claims, experimental results, references, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our method is to mitigate safety erosion during fully benign federated fine-tuning of large language models while preserving task utility. Harmful supervision is used offline to construct fixed reference adapters, including harmful prompt-compliant response pairs for ERA and harmful prompt-refusal pairs for RRA; clients' federated task-training data remain benign. We evaluate safety erosion and mitigation using established public safety benchmarks. We acknowledge the potential dual-use risks associated with constructing erosion-oriented references and analyzing safety failures. FedCURE is intended as a defensive framework, and improvements on the evaluated benchmarks should not be interpreted as guarantees of safety in deployment.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility of our federated learning simulations and safety evaluations, we provide detailed descriptions of our experimental setup and framework implementation. Section 5 and Appendix A specify the model configurations, LoRA hyperparameters, optimization settings, federated training protocol, data partitioning, and evaluation procedures. We also document the construction and use of the offline Erosion Reference Adapter (ERA) and Reinforce Reference Adapter (RRA), together with the server-side Erosion Consensus Filtering (ECF) procedure and its low-rank implementation.

All downstream training datasets and safety evaluation benchmarks used in our experiments are publicly available, and their preprocessing and evaluation protocols are described in Section 5 and Appendix A. Appendix B details the implementation and federated adaptation of all comparison methods, while Appendix C reports runtime and computational overhead. Appendices D–F provide additional geometric analysis, properties of the RRA correction, and implementation details for the low-rank server-side operations. Appendix G reports extended ablations and robustness evaluations across random seeds, reference datasets, model families, client cohort sizes, data heterogeneity, and model scales. We also provide the implementation code as supplementary material to support reproduction of the proposed method and experiments.

REFERENCES

Andrei Alexandru, Antonia Calvi, Henry Broomfield, Jackson Golden, Kyle Dai, Mathias Leys, Maurice Burger, Max Bartolo, Roman Engeler, Sashank Pisupati, Toby Drane, and Young Sun Park. Atla selene mini: A general purpose evaluation model, 2025. URL https://arxiv.org/abs/2501.17195.

Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=pH3XAQME6c.

Thong Bach, Thanh Nguyen-Tang, Dung Nguyen, Thao Minh Le, and Truyen Tran. Curvature-aware safety restoration in LLMs fine-tuning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=FSUehLhGyl.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, 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. URL https://openreview.net/forum?id=gT5hALch9z.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, and Gauri Joshi. Heterogeneous LoRA for federated fine-tuning of on-device foundation models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12903–12913, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.717. URL https://aclanthology.org/2024.emnlp-main.717/.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL http://jmlr.org/papers/v25/23-0870.html.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Wenzhi Fang, Dong-Jun Han, Liangqi Yuan, Seyyedali Hosseinalipour, and Christopher Brinton. Federated sketching loRA: A flexible framework for heterogeneous collaborative fine-tuning of LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=qFKFSVvff5.

Yan Gao, Massimo Roberto Scamarcia, Javier Fernandez-Marques, Mohammad Naseri, Chong Shen Ng, Dimitris Stripelis, Zexi Li, Tao Shen, Jiamu Bai, Daoyuan Chen, Zikai Zhang, Rui Hu, InSeo Song, Lee KangYoon, Hong Jia, Ting Dang, Junyan Wang, Zheyuan Liu, Daniel Janes Beutel, Lingjuan Lyu, and Nicholas D. Lane. Flowertune: A cross-domain benchmark for federated fine-tuning of large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=l8Nb6ecZjW.

Zihan Guan, Mengxuan Hu, Ronghang Zhu, Sheng Li, and Anil Vullikanti. Benign samples matter! fine-tuning on outlier benign samples severely breaks safety. In *Forty-second International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/guan25c.html.

Seokil Ham, Jaehyuk Jang, Wonjun Lee, and Changick Kim. Jailbreak to protect: Buffering and reinforcing via temporary jailbreaking for safe fine-tuning in large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=O0JXgBBo3k.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=Ich4tv4202.

Luxi He, Mengzhou Xia, and Peter Henderson. What is in your safe data? identifying benign data that breaks safety. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Hi8jKh4HE9.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/77baa7c2a3a675823e89131698fd6e19-Abstract-Conference.html.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.319/.

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 *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=RPChapuXlC.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=lpXDZKiAnt.

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*, 2025a. URL https://openreview.net/pdf?id=Arepl4R86m.

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. URL https://openreview.net/forum?id=tTPHgb0EtV.

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 *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=g0QovXbFw3.

Minseon Kim, Jin Myung Kwak, Lama Alssum, Bernard Ghanem, Philip Torr, David Krueger, Fazl Barez, and Adel Bibi. Rethinking safety in LLM fine-tuning: An optimization perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=ZnOoEA2nDn.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, Yiguan Lin, Bin Xu, Bowen Ren, Chong Feng, Yang Gao, and Heyan Huang. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.599/.

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*, 2025. URL https://openreview.net/forum?id=GOoVzE9nSj.

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. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=f3TUipYU3U.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Wenbo Pan, Zhichao Liu, Qiguang Chen, Xiangyang Zhou, Yu Haining, and Xiaohua Jia. The hidden dimensions of LLM alignment: A multi-dimensional analysis of orthogonal safety directions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=wGFEzfhFae.

ShengYun Peng, Pin-Yu Chen, Matthew Daniel Hull, and Duen Horng Chau. Navigating the safety landscape: Measuring risks in finetuning large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GZnsqBwHAG.

Kaustubh Ponkshe, Shaan Shah, Raghav Singhal, and Praneeth Vepakomma. Safety subspaces are not linearly distinct: A fine-tuning case study. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Fj6LakRHcT.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hTEGyKf0dZ.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024.

Zhanming Shen, Tianqi Xu, Hao Wang, Jian Li, and Miao Pan. pFedGPT: Hierarchically optimizing LoRA aggregation weights for personalized federated GPT models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4766–4778. Association for Computational Linguistics, November 2025. URL https://aclanthology.org/2025.emnlp-main.239/.

Abhay Sheshadri, Aidan Ewart, Phillip Huang Guo, Aengus Lynch, Cindy Wu, Vivek Hebbar, Henry Sleight, Asa Cooper Stickland, Ethan Perez, Dylan Hadfield-Menell, and Stephen Casper. Latent adversarial training improves robustness to persistent harmful behaviors in LLMs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=6LxMeRlkWl.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongREJECT for empty jailbreaks. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=KZLE5BaaOH.

Max Springer, Chung Peng Lee, Blossom Metevier, Jane Castleman, Bohdan Turbal, Hayoung Jung, Zeyu Shen, and Aleksandra Korolova. The geometry of alignment collapse: When fine-tuning breaks safety, 2026. URL https://arxiv.org/abs/2602.15799.

Zichen Tang, Zhenheng Tang, Gaoning Pan, Buhua Liu, Xin He, Kunfeng Lai, Xiaowen Chu, and Bo Li. Ghost in the cloud: Your geo-distributed large language models training is easily manipulated. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FwnmQnVc7g.

Mingxiang Tao, Yu Tian, Wenxuan Tu, Yue Yang, Xue Yang, and Xiangyan Tang. Safe-FedLLM: Delving into the safety of federated large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24405–24420, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1120. URL https://aclanthology.org/2026.acl-long.1120/.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=sMtiGB2YZT.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. FLoRA: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=TcCorXxNJQ.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 52588–52610, 2024. URL https://proceedings.mlr.press/v235/wei24f.html.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=80IwJqlXs8.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CfXh93NDgH.

Shuo Yang, Qihui Zhang, Yuyang Liu, Yue Huang, Xiaojun Jia, Kun-Peng Ning, Jia-Yu Yao, jigang wang, Dai Hailiang, Yibing Song, and Li Yuan. AsFT: Anchoring safety during LLM fine-tuning within narrow safety basin. In *NeurIPS 2025 Workshop: Reliable ML from Unreliable Data*, 2025. URL https://openreview.net/forum?id=96UEsasTfE.

Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Yaxin Du, Yang Liu, Yanfeng Wang, and Siheng Chen. FedLLM-bench: Realistic benchmarks for federated learning of large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL https://openreview.net/forum?id=djGx0hucok.

Rui Ye, WenHao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. OpenfedLLM: Training large language models on decentralized private data via federated learning. In *ICLR 2024 Workshop: How Far Are We From AGI*, 2024b. URL https://openreview.net/forum?id=fOuN2GdtlY.

Rui Ye, Jingyi Chai, Xiangrui Liu, Yaodong Yang, Yanfeng Wang, and Siheng Chen. Emerging safety attack and defense in federated instruction tuning of large language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sYNWqQYJhz.

Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5650–5659, 2018. URL https://proceedings.mlr.press/v80/yin18a.html.

Bingjie Zhang, Yibo Yang, Zhe Ren, Dandan Guo, Jindong Gu, Philip Torr, and Bernard Ghanem. A guardrail for safety preservation: When safety-sensitive subspace meets harmful-resistant null-space. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/pdf?id=887vde4ZAW.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6915–6919. IEEE, 2024.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. LLMs encode harmfulness and refusal separately. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/cd18539787d90e1d682d557c2c71b534-Abstract-Conference.html.

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=yR47RmND1m.

Guanghao Zhou, Panjia Qiu, Cen Chen, Hongyu Li, Jason Chu, Xin Zhang, and Jun Zhou. LSSF: Safety alignment for large language models through low-rank safety subspace fusion. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30621–30638, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1479. URL https://aclanthology.org/2025.acl-long.1479/.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Xin Zhou, Yi Lu, Ruotian Ma, Yujian Wei, Tao Gui, Qi Zhang, and Xuanjing Huang. Making harmful behaviors unlearnable for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10258–10273. Association for Computational Linguistics, August 2024. URL https://aclanthology.org/2024.findings-acl.611/.

Shunan Zhu, Jiawei Chen, Yonghao Yu, and Hideya Ochiai. FedDetox: Robust federated SLM alignment via on-device data sanitization. *arXiv preprint arXiv:2604.06833*, 2026. URL https://arxiv.org/abs/2604.06833.

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

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=IbIB8SBKFV.

A Experimental Setup

All experiments are conducted on a server equipped with a single NVIDIA B200 GPU with 180GB of memory, AMD EPYC 9365 36-core processor × 2 (72 cores total), and 2.2TB of system memory. We use bf16 mixed-precision training with a maximum sequence length of 1024 tokens. For training, we use gradient accumulation to obtain an effective batch size of 128.

Model and Datasets. We evaluate Meta-Llama-3-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-it. For downstream fine-tuning, we use instruction-tuning data from Dolly-15K (Conover et al., 2023) and WizardLM (Xu et al., 2024), together with a commonsense reasoning benchmark (Hu et al., 2023) comprising ARC-Challenge, ARC-Easy, BoolQ, HellaSwag, OpenBookQA, PIQA, Social-IQA, and WinoGrande. These datasets cover instruction following, conversational assistance, and commonsense reasoning. Following prior federated LLM fine-tuning studies (Zhang et al., 2024; Ye et al., 2024b;a), we simulate non-iid clients via Dirichlet partitioning with concentration $\alpha = 0.5$ by default. To reduce computational cost, we subsample Dolly and WizardLM while preserving their original task/category proportions, consistent with prior federated language model fine-tuning studies using task-wise subsampling (Ye et al., 2025; Shen et al., 2025). For Dolly and the commonsense benchmark, client proportions for each instruction category or task are drawn from dirichlet distribution, whereas WizardLM uses balanced client partitions. We consider benign federated fine-tuning without malicious clients or injected harmful training examples.

Federated Fine-Tuning and Optimization. We use synchronous federated training with eight fully participating clients and one local epoch per communication round, training for 10 rounds by default and extending up to 50 rounds. Clients optimize LoRA adapters applied to all linear decoder layers while keeping the backbone frozen. We use rank $r = 16$ and LoRA scaling parameter $\alpha_{\text{LoRA}} = 32$. Local training minimizes a response-only supervised fine-tuning objective with AdamW at a learning rate of 10^{-4}. We select the learning rate through centralized fine-tuning under the same LoRA configuration, evaluating candidates in $\{5 \times 10^{-6}, 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 10^{-4}\}$ for one epoch each. The candidate achieving the lowest validation loss, 10^{-4}, is used for federated training with a constant learning-rate schedule and no warm-up. FedIT provides the default aggregation rule, with method-specific interventions described separately. The resulting global adapter initializes local training in the next round. For ERA and RRA, ERA is trained by supervised fine-tuning on 5K harmful prompt-compliant response pairs, and RRA on 2.5K harmful prompt-refusal pairs combined with 2.5K examples from the Alpaca dataset. Both reference adapters are trained for 3 epochs using the same optimizer and learning rate (10^{-4} with a cosine schedule) as client local training. For FedCURE, we set the consensus threshold 0.5, the task-subspace attenuation $\alpha_b = 0.1$, and the merge scale $c = 0.1$; these FedCURE hyperparameters are shared across all models and datasets.

Metrics. We evaluate safety using harmful score (HS) on AdvBench (Zou et al., 2023), StrongREJECT (Souly et al., 2024), XSTest-harmful (Röttger et al., 2024), and the standard and contextual subsets of HarmBench (Mazeika et al., 2024). For each of the two judges, WildGuard (Han et al., 2024) and the HarmBench (Mazeika et al., 2024) classifier, harmful-response rates are aggregated across these subsets using the number of valid judgments as weights; harmful score is the equal-weighted average of the two resulting rates. Over-refusal is measured by WildGuard's refusal rate on XSTest-harmless (Röttger et al., 2024). Task performance is measured using ROUGE-L F1 on held-out Dolly examples excluding creative writing and brainstorming; macro-averaged accuracy across the eight commonsense tasks; and MT-Bench for WizardLM. MT-Bench uses Selene-1-Mini (Alexandru et al., 2025) as the judge and averages scores across both conversation turns. General utility is assessed using MMLU micro-accuracy pooled over examples, MMLU (Hendrycks et al., 2021) macro-accuracy averaged over subjects, and IFEval (Zhou et al., 2023) strict instruction-level accuracy. All metrics are reported as percentages except MT-Bench, which uses a 1–10 scale.

B Baseline Implementations

We compare FedCURE with various baselines. These baselines are implemented based on the original papers following their algorithmic designs and public reference implementations, using the reported configurations as the basis for method-specific settings. Unless otherwise specified, downstream federated task training follows the protocol and optimization settings in Appendix A. Below, we

describe federated adaptations of each method and identify the auxiliary datasets used for safety construction, alignment, or restoration.

Multi Krum (Blanchard et al., 2017). We implement the original Multi-Krum selection rule and apply it to client LoRA updates at the server. After local training, each update is computed relative to the broadcast LoRA modules and scored by the sum of squared Euclidean distances to its $n - f - 2$ nearest neighbors, where n denotes participating clients and f the assumed the budget. The server uniformly averages the m lowest-scoring updates and adds the result to the global adapter for redistribution. We use $(f, m) = (2, 6)$ for 8 clients and $(8, 24)$ for 32 clients.

Trimmed Mean (Yin et al., 2018). We implement the original coordinate-wise trimming rule and apply it to client LoRA updates after local training. For every coordinate, the server sorts the client updates, removes the largest and smallest b values, and uniformly averages the remainder. The resulting update is added to the global adapter and broadcast for the next round. We set $b = 1$ for 8 clients and $b = 4$ for 32 clients, maintaining a trimming proportion of $1/8$ at each tail. Local task training follows the common federated protocol, and the aggregation requires no auxiliary safety data.

SaLoRA (Li et al., 2025). We implement fixed safety projection and task-aware initialization of SaLoRA. The safety subspace is estimated from harmful prompts paired with safe refusal responses in Circuit Breakers (Zou et al., 2024). To establish a common parameterization across clients, the server constructs the task-aware initialization once from pooled client training samples. A fixed residual adjustment preserves the original model at initialization. Clients optimize their LoRA modules on local data under the shared projection onto the orthogonal complement of the safety subspace. The server forms the global adapter through sample-weighted averaging of the trained LoRA modules and redistributes it after each round.

SafeLoRA (Hsu et al., 2024). We implement the alignment operator and layer-selection criterion of SafeLoRA, extending their application to individual client adapters before every federated aggregation. For each model family, we use the original instruction-tuned checkpoint as the aligned reference and its corresponding pretrained base checkpoint before instruction tuning, as the unaligned reference. Layer-wise alignment operators are constructed from the weight differences between these fixed checkpoints. After local training, the server independently selects layers for each client when the cosine similarity between the LoRA update and its projected counterpart is at most 0.5. It projects the selected output factors of the full uploaded adapter and combines the corrected adapters through sample-weighted averaging of their LoRA parameters.

GuardSpace (Zhang et al., 2026). We implement both components of GuardSpace: safety-irrelevant initialization and a fixed input projector during local training. We use 520 harmful prompts from BeaverTails (Ji et al., 2023) for initialization and 520 prompt-disjoint harmful prompts from Circuit Breakers (Zou et al., 2024) for projector construction. The initialization uses weight components weakly associated with harmful-prompt activations, while the projector restricts adaptation to the approximate null space of the reference input covariance. A fixed residual adjustment preserves the original model at initialization. All clients share the initialization and projector, and the server combines their LoRA modules through sample-weighted averaging after each round. The reference geometry remains fixed throughout federate fine-tuning.

Booster (Huang et al., 2025b). We implement the alignment objective of Booster and apply it once before federated fine-tuning. Circuit Breakers (Zou et al., 2024) provides harmful prompts paired with safe refusal responses for alignment, while BeaverTails (Ji et al., 2023) supplies harmful-completion examples for simulated harmful adaptation. The alignment stage minimizes the safe-response loss while penalizing the decrease in harmful-response loss induced by a simulated harmful update. We merge the resulting alignment adapter into the backbone and distribute this model as the common initialization for federated fine-tuning. Clients then train fresh task LoRA adapters on their local benign data under the common task-training protocol, and the server combines their factors through sample-weighted averaging.

FedLLM-Attack (Ye et al., 2025). We use the Level-2 server-side defense of FedLLM-Attack as a post-hoc safety-restoration baseline. After standard federated task training, the server continues training the global LoRA adapter using the released Mistral-generated defense data. Following the released defense configuration, we use 1,000 examples: 500 harmful instructions paired with harmless responses and 500 ordinary instructions paired with helpful responses. The server performs 500

optimizer steps while keeping the backbone frozen and retaining the task adapter's rank and target modules. The restored checkpoint is then evaluated as the defended global model.

C Runtime and Computational Overhead

We analyze the computational overhead of FedCURE. All measurements use Meta-Llama-3-8B-Instruct on Dolly with eight clients on a single NVIDIA B200 GPU, with each job running alone. Clients are trained sequentially, and per-round costs average across communication rounds.

Reference Adapter Construction. FedCURE requires an additional prestage that constructs ERA and RRA by supervised fine-tuning on the frozen base model. ERA and RRA construction take 11 minutes 53 seconds and 6 minutes 55 seconds, respectively, or 18.9 minutes in total including adapter loading. This cost is incurred only once, and both adapters remain fixed throughout federated training.

Table 7: Measured costs on Meta-Llama-3-8B-Instruct with Dolly and eight clients using a single NVIDIA B200 GPU. Server time includes aggregation, method-specific server operations, and adapter export.

	One-time	Per round (s)		
Method	(min)	Client	Server	Total
FedIT	–	407.7	6.6	414.3
SafeLoRA	<0.1	406.2	18.8	424.9
SaLoRA[‡]	1.6	377.1	6.9	384.1
GuardSpace[‡]	8.4	468.3	8.9	477.2
Booster[†]	53.4	407.7	6.6	414.3
FedLLM-Attack[†]	27.5	407.7	6.6	414.3
FedCURE	18.9	484.1	15.8	499.9

Per-round training cost. As shown in Table 7, FedCURE requires 499.9 seconds per round, compared with 414.3 seconds for FedIT. Most of this increase comes from client training (+76.4 seconds, 18.7%), because every local forward and backward pass includes the frozen ERA branch. ECF and RRA merging add 9.2 seconds of server time, as they operate on rank-16 adapter factors without LLM forward or backward passes.

Comparison with Baselines. One-time cost of FedCURE is 35% of Booster's alignment and 69% of FedLLM-Attack's post-hoc restoration, while it exceeds the preparation of GuardSpace (8.4 minutes) and SaLoRA (1.6 minutes). SafeLoRA requires no separate preparation but adds 12.2 seconds of server-side projection per round, and GuardSpace adds 60.6 seconds of client-side projection per round.

Summary. Overall, FedCURE introduces a one-time reference adapter construction and a moderate per-round increase dominated by ERA-conditioned local training. Since the construction cost is incurred only once, its relative cost decreases over longer training. Although the per-round increase recurs throughout training, it remains feasible on a single GPU and is the cost of substantially safer federated fine-tuning with competitive task performance.

D Geometric Interpretation and Empirical Evidence

We examine how consensus among client marginal spaces positively aligned with the Erosion Reference Adapter (ERA) overlaps the previous broadcast factor space. We use the candidates defined in equation 4, with marginal support and positive reference contribution established in Section 4.2. This geometry motivates suppressing further marginal accumulation along the selected directions. We then assess the behavioral relevance of this selection empirically.

Broadcast Space Localization. Let $P_{\mathcal{S}}$ denote the orthogonal projector onto $\mathcal{S}$. Define the previous broadcast factor space and the outside components of the ERA-aligned candidate spaces as

$$\mathcal{B}_t = \mathrm{col}(B_{t-1}), \qquad \mathcal{E}^{+}_{i,t} = \mathrm{col}\big((I - P_{\mathcal{B}_t})Q^{+}_{i,t}\big) \subseteq \mathcal{B}_t^{\perp}.$$

These residual spaces are obtained by projecting the already selected candidates outside $\mathcal{B}_t$. The superscript $+$ records their origin in the ERA-aligned candidate spaces; no further sign selection is applied to the residuals. Define their maximum agreement as

$$C_{\perp,t}^{+} = \sum_i p_{i,t} P_{\mathcal{E}_{i,t}^{+}}, \qquad \nu_t^{+} = \|C_{\perp,t}^{+}\|_2 \in [0,1].$$

The scalar ν_t^{+} is the spectral ceiling of agreement among the outside components of the selected candidates. Decomposing $Q_{i,t}^{+}$ into its projections onto $\mathcal{B}_t$ and $\mathcal{B}_t^{\perp}$ gives $\mathrm{col}(Q_{i,t}^{+}) \subseteq \mathcal{B}_t \oplus \mathcal{E}_{i,t}^{+}$. Orthogonal-projector ordering and averaging therefore yield

$$\begin{aligned} 0 \preceq C_t^{\Delta} &= \sum_i p_{i,t} Q_{i,t}^{+} (Q_{i,t}^{+})^{\top} \\ &\preceq P_{\mathcal{B}_t} + C_{\perp,t}^{+} \preceq P_{\mathcal{B}_t} + \nu_t^{+}(I - P_{\mathcal{B}_t}). \end{aligned} \tag{10}$$

Proposition 1 (Cumulative-space localization for ERA-aligned candidates)**.** *Assume exact singular subspaces and projectors in the construction above, with orthonormal $Q_{i,t}^{+}$ and $\sum_{i\in\mathcal{I}_t} p_{i,t} = 1$. If $\nu_t^{+} < \tau$, then every unit vector $u \in \mathrm{ran}(P_{H,t})$ satisfies*

$$\|P_{\mathcal{B}_t} u\|_2^2 \geq \frac{\tau - \nu_t^{+}}{\tau(1 - \nu_t^{+})}.$$

Equivalently,

$$\left\|(I - P_{\mathcal{B}_t}) Q_{H,t}\right\|_2^2 \leq \delta_t^{+} := \frac{\nu_t^{+}(1-\tau)}{\tau(1-\nu_t^{+})} < 1, \tag{11}$$

and $\mathrm{rank}(P_{H,t}) \leq \dim(\mathcal{B}_t) \leq r$.

Proof. Write $P = P_{\mathcal{B}_t}$ and $\eta = \nu_t^{+}$. For any unit $u \in \mathrm{ran}(P_{H,t})$, spectral selection and equation 10 give

$$\tau u u^{\top} \preceq \tau P_{H,t} \preceq C_t^{\Delta} \preceq P + \eta(I - P).$$

If $\eta = 0$, then $0 \preceq C_t^{\Delta} \preceq P$ implies $\mathrm{ran}(C_t^{\Delta}) \subseteq \mathcal{B}_t$. Since $\tau > 0$, the selected space is contained in $\mathrm{ran}(C_t^{\Delta})$, so both bounds follow. Otherwise, $M = P + \eta(I - P)$ is positive definite. Conjugating $\tau u u^{\top} \preceq M$ by $M^{-1/2}$ gives $\tau u^{\top} M^{-1} u \leq 1$. With $a = \|Pu\|_2^2$ and $M^{-1} = P + \eta^{-1}(I - P)$, this becomes

$$\tau\left(a + \frac{1-a}{\eta}\right) \leq 1, \qquad a \geq \frac{\tau - \eta}{\tau(1-\eta)}.$$

Taking the worst unit vector in the selected space yields equation 11. The assumption $\eta < \tau$ gives $1 - \delta_t^{+} = (\tau - \eta)/[\tau(1-\eta)] > 0$. Hence the restriction of P to this space is injective, giving the rank bound. □

When the outside components of the ERA-aligned candidates cannot reach τ on their own, Proposition 1 shows that the selected high-consensus modes must substantially overlap the previous broadcast factor space. The bound applies to every unit vector in the selected space, including mixtures of its eigenvectors. It therefore also describes the energy of the removed marginal component

$$D_t^{\mathrm{rem}} = D_t - \widetilde{D}_t = P_{H,t} D_t.$$

Applying the localization bound to each column gives

$$\|P_{\mathcal{B}_t} D_t^{\mathrm{rem}}\|_F^2 \geq (1 - \delta_t^{+}) \|D_t^{\mathrm{rem}}\|_F^2. \tag{12}$$

Thus, whenever $\nu_t^{+} < \tau$, at least a fraction $1 - \delta_t^{+}$ of the removed component's energy occupies the previous broadcast factor space. The intervention is selected through ERA-aligned client marginal spaces, and the bound quantifies how much of its removed energy lies along directions already represented by the broadcast factors. The reference space for this statement is $\mathrm{col}(B_{t-1})$; it also equals the cumulative product's output space when $\mathrm{col}(B_{t-1}A_{t-1}) = \mathrm{col}(B_{t-1})$, for example when A_{t-1} has full row rank.

The squared-norm decrease caused by filtering is

$$\|D_t\|_F^2 - \|\widetilde{D}_t\|_F^2 = \|D_t^{\mathrm{rem}}\|_F^2,$$

and equation 7 gives

$$P_{H,t}\widetilde{\Delta W}_t = P_{H,t}\Delta W_{t-1}. \tag{13}$$

ECF therefore suppresses additional marginal changes along the selected directions while preserving their previous cumulative values before rank retraction. Combined with equation 12, this gives a geometric rationale for limiting further accumulation along directions selected from ERA-aligned marginal spaces that overlap the broadcast state. Repeating this intervention at aggregation is intended to curb cumulative safety erosion. The proposition establishes a round-wise geometric localization property; the safety effect of repeatedly applying the intervention is assessed empirically.

E Properties of RRA Correction

Fix one module and round, and consider the RRA correction in Section 4.3. We use its orthogonal task projector $P_{T,t}$, attenuated reference $W'_{R,t}$, and convex target ΔW_t^{mix}, with $0 \le \alpha_b, c \le 1$. Appendix F gives their low-rank realization; here we establish norm bounds and the effect on the selected marginal component.

Norm bounds. By equation 8, $W'_{R,t} = (1-\alpha_b)P_{T,t}W_R + (I - P_{T,t})W_R$. The two terms are orthogonal in the Frobenius inner product, so

$$\begin{aligned}\|W'_{R,t}\|_F^2 &= (1-\alpha_b)^2\|P_{T,t}W_R\|_F^2 + \|(I-P_{T,t})W_R\|_F^2 \\ &= \|W_R\|_F^2 - \alpha_b(2-\alpha_b)\|P_{T,t}W_R\|_F^2 \le \|W_R\|_F^2.\end{aligned}$$

Rank-r truncated SVD discards singular-value energy. Applying the triangle inequality to equation 9 therefore gives

$$\begin{aligned}\|\Delta W_t\|_F &\le \|\Delta W_t^{\text{mix}}\|_F \\ &\le (1-c)\|\Delta W_t^{\text{stab}}\|_F + c\|W'_{R,t}\|_F \\ &\le \max\{\|\Delta W_t^{\text{stab}}\|_F, \|W_R\|_F\}.\end{aligned}$$

This bounds the correction's norm relative to its two inputs. It does not imply a decrease relative to ΔW_t^{stab} alone or across rounds, nor does it establish behavioral safety or task-utility preservation.

Effect on marginal stabilization. Let $E_t = \Delta W_t^{\text{stab}} - \widetilde{\Delta W}_t$ be the first compression residual from equation 15, and let $E_t^{\text{mix}} = \Delta W_t - \Delta W_t^{\text{mix}}$ be the final compression residual. Substituting these definitions into equation 9 and subtracting ΔW_{t-1} yields

$$\begin{aligned}\Delta W_t - \Delta W_{t-1} = (1-c)\big((I-P_{H,t})D_t + E_t\big) \\ + c(W'_{R,t} - \Delta W_{t-1}) + E_t^{\text{mix}}.\end{aligned}$$

Since $P_{H,t}(I - P_{H,t}) = 0$, projecting gives the exact identity

$$\begin{aligned}P_{H,t}(\Delta W_t - \Delta W_{t-1}) = cP_{H,t}(W'_{R,t} - \Delta W_{t-1}) \\ + (1-c)P_{H,t}E_t + P_{H,t}E_t^{\text{mix}}.\end{aligned}$$

Even when both compression residuals vanish, the RRA term need not be zero. The task projector $P_{T,t}$ and consensus projector $P_{H,t}$ are constructed from different spaces, so attenuating the reference's task overlap does not enforce preservation of the selected cumulative component. The localization bound in Proposition 1 continues to describe the removed marginal $P_{H,t}D_t$; the preservation identity in equation 13 concerns the intermediate state $\widetilde{\Delta W}_t$ before rank retraction and RRA correction.

F Low Rank Factorization

This appendix realizes the server-side effective-weight operations in Section 4 using thin factors. We consider its sign-based ERA selection, fixed consensus threshold, and RRA correction. Fix one module and write $m = d_{\text{out}}$, $n = d_{\text{in}}$, and $N = |\mathcal{I}_t|$. The Client LoRA factor width and marginal-basis rank cap are both r; the frozen references $W_E = s_E B_E A_E$ and $W_R = s_R B_R A_R$ have factor widths r_E and r_R, which need not equal r. Dense $m \times n$ adapter products and $m \times m$ projectors are unnecessary. The factor identities below are algebraically exact; the prescribed rank truncations determine the retained approximation.

F.1 Compact SVD and rank retraction

For $L \in \mathbb{R}^{m\times k}$ and $F \in \mathbb{R}^{k\times n}$, reduced QR factorizations give

$$L = Q_L T_L, \qquad F^\top = Q_F T_F,$$
$$T_L T_F^\top = U_c \Sigma V_c^\top, \quad LF = (Q_L U_c)\Sigma(Q_F V_c)^\top.$$

Thus an SVD of the core, of size $\min(m,k)\times\min(n,k)$, recovers the product's nonzero singular values and vectors. This also holds for rank-deficient factors. Denote the resulting decomposition by $\mathrm{cSVD}(L,F) = (U,\Sigma,V^\top)$.

Retaining the leading $\ell \le r$ singular triplets and zero-padding to rank-r factor shapes gives

$$B_{:,1:\ell} = U_{:,1:\ell}(\Sigma_\ell/s)^{1/2}, \qquad A_{1:\ell,:} = (\Sigma_\ell/s)^{1/2} V_{:,1:\ell}^\top, \qquad \|sBA - LF\|_F^2 = \sum_{j>\ell}\sigma_j(LF)^2. \tag{14}$$

Both retractions $\mathcal{R}_r$ in equation 7 and equation 9 retain up to r leading nonzero singular triplets. Thin-factor realization itself introduces no additional approximation.

F.2 ERA-aligned marginals and consensus

The client marginal in Section 4.2 is exactly

$$D_{i,t} = L_{i,t}F_{i,t}, \qquad L_{i,t} = [B_{i,t} \;\; B_{t-1}], \qquad F_{i,t} = s\begin{bmatrix} A_{i,t} \\ -A_{t-1}\end{bmatrix}.$$

Its factor width is $2r$, although only the leading $k_{i,t} \le r$ numerically nonzero left singular vectors of $\mathrm{cSVD}(L_{i,t}, F_{i,t})$ are retained as $V_{\Delta i,t}$. This restriction defines the candidate search space; it does not truncate the server displacement.

The signed core in equation 4 is computed from two $k_{i,t}\times n$ matrices:

$$Y_{i,t} = ((V_{\Delta i,t})^\top L_{i,t})F_{i,t},$$
$$Z_{i,t} = ((V_{\Delta i,t})^\top B_E)(s_E A_E),$$
$$K_{i,t} = \mathrm{sym}(Y_{i,t}Z_{i,t}^\top).$$

Associativity gives $Y_{i,t}Z_{i,t}^\top = (V_{\Delta i,t})^\top D_{i,t} W_E^\top V_{\Delta i,t}$ without forming $D_{i,t}$, W_E, or $D_{i,t}W_E^\top$. Diagonalizing this at-most-$r\times r$ core yields the positive candidate basis $Q_{i,t}^+$ of equation 4.

Let $q_{i,t}$ be its number of columns and $q_t = \sum_i q_{i,t} \le Nr$. Form

$$G_t = [\sqrt{p_{i,t}}Q_{i,t}^+]_{i\in\mathcal{I}_t}, \qquad G_t = U_G\Sigma_G V_G^\top, \qquad C_t^\Delta = G_tG_t^\top = U_G\Sigma_G^2 U_G^\top.$$

A thin SVD of the $m\times q_t$ matrix G_t therefore suffices. Its squared singular values are the consensus scores; retaining all columns of U_G with $\sigma_j(G_t)^2 \ge \tau$ gives $Q_{H,t}$. Empty candidates contribute no columns, with their original client weights unchanged. An empty union gives an empty selected basis. Neither the consensus matrix nor its spectral projector is formed explicitly.

F.3 Factored filtering and RRA correction

For the factor-wise aggregate $\Delta W_t^{\mathrm{agg}} = s\bar{B}_t\bar{A}_t$, where $\bar{B}_t = \sum_{i\in\mathcal{I}_t} p_{i,t}B_{i,t}$ and $\bar{A}_t = \sum_{i\in\mathcal{I}_t} p_{i,t}A_{i,t}$, define the server marginal factors by

$$D_t = L_tF_t, \qquad L_t = [\bar{B}_t \;\; B_{t-1}], \qquad F_t = s\begin{bmatrix}\bar{A}_t \\ -A_{t-1}\end{bmatrix}.$$

Combining equation 6 and equation 7 gives the following width-$3r$ product:

$$\widetilde{\Delta W}_t = \Delta W_t^{\mathrm{agg}} - Q_{H,t}(Q_{H,t}^\top L_t)F_t = \underbrace{[\bar{B}_t \;\; -Q_{H,t}(Q_{H,t}^\top L_t)]}_{L_t^{\mathrm{stab}}}\underbrace{\begin{bmatrix}s\bar{A}_t \\ F_t\end{bmatrix}}_{F_t^{\mathrm{stab}}}.$$

Compact retraction gives $\Delta W_t^{\mathrm{stab}} = s\widehat{B}_t\widehat{A}_t$. If $Q_{H,t}$ is empty or $(\bar{B}_t, \bar{A}_t) = (B_{t-1}, A_{t-1})$, the aggregated factors are retained without retraction; otherwise the retraction is applied.

The compression residual $E_t = \Delta W_t^{\mathrm{stab}} - \widetilde{\Delta W}_t$ makes the scope of equation 7 explicit:

$$P_{H,t}(\Delta W_t^{\mathrm{stab}} - \Delta W_{t-1}) = P_{H,t}E_t, \qquad \|P_{H,t}E_t\|_F \le \|E_t\|_F. \tag{15}$$

When the retraction is applied, $\|E_t\|_F^2$ equals the discarded singular-value tail in equation 14. Thus the geometry of Appendix D concerns the selected space and filtering before compression.

For RRA correction, $\mathrm{cSVD}(\widehat{B}_t, s\widehat{A}_t)$ supplies the task basis $Q_{T,t}$: retain the modes with $\sigma_j > \tau_b\sigma_1$, as in Section 4.3. The projected reference and convex target have factors

$$\begin{aligned} B'_{R,t} &= B_R - \alpha_b Q_{T,t}(Q_{T,t}^\top B_R), \\ L_t^{\mathrm{mix}} &= [(1-c)s\widehat{B}_t \;\; c\,s_R B'_{R,t}], \\ F_t^{\mathrm{mix}} &= \begin{bmatrix} \widehat{A}_t \\ A_R \end{bmatrix}, \qquad \Delta W_t^{\mathrm{mix}} = L_t^{\mathrm{mix}} F_t^{\mathrm{mix}}. \end{aligned}$$

This width-$(r + r_R)$ product equals equation 9; compact rank-r retraction produces the broadcast factors. Concatenation implements the weighted sum of products exactly, whereas averaging the two pairs of factors would introduce cross terms.

F.4 Time and memory complexity

Costs are per module and round. Assume $Nr \ll m$ and $\max(3r, r + r_R) \ll \min(m, n)$. For a compact SVD of width k in this regime, including recovery of its thin singular vectors,

$$\mathsf{F}(k) = \mathcal{O}((m+n)k^2 + k^3), \qquad \mathsf{M}(k) = \mathcal{O}((m+n)k + k^2) \tag{16}$$

give time and workspace, respectively. Since $\mathrm{tr}(C_t^\Delta) = \sum_i p_{i,t}q_{i,t} \le r$ and each selected eigenvalue is at least τ, the selected rank satisfies $h_t = \mathrm{rank}(P_{H,t}) \le \min(q_t, \lfloor r/\tau \rfloor)$. The update computations have the following costs:

Operation	Time
Factor-wise aggregation	$\mathcal{O}(N(m+n)r)$
Client SVDs and ERA-positive selection	$\mathcal{O}(N[(m+n)r(r+r_E) + r^3])$
Consensus SVD	$\mathcal{O}(mq_t^2 + q_t^3)$
Marginal projection and retraction	$\mathcal{O}(mh_tr) + \mathsf{F}(3r)$
Optional RRA task basis, projection, and mixing	$\mathsf{F}(r) + \mathcal{O}(mrr_R) + \mathsf{F}(r + r_R)$

When $r_E, r_R = \mathcal{O}(r)$, total time and storage are bounded by

$$\begin{aligned} T_{\mathrm{round}} &= \mathcal{O}\big(N(m+n)r^2 + m(Nr)^2 + (Nr)^3\big), \\ M_{\mathrm{round}} &= \mathcal{O}\big(N(m+n)r + (Nr)^2\big). \end{aligned}$$

Storage includes client/reference factors and the active module's workspace. For fixed client count and ranks, cost is linear in the ambient dimensions; dependence on N includes the quadratic consensus term. Model-wide time sums over modules, while memory also includes their stored adapter states and selected bases. Local training, the backbone, evaluation, and separate alignment analyses are outside these server-processing bounds. Communication remains $(m+n)r$ LoRA factor entries per client and module in each direction.

Reduction relative to dense computation. A conventional dense SVD of an $m \times n$ product costs $\mathcal{O}(mn\min(m,n))$, whereas equation 16 uses its width-k factors. For $m = n = d \gg k$, the leading dimension dependence drops from d^3 to dk^2, an order-of-growth reduction by $(d/k)^2$. Product storage drops from mn to $(m+n)k$ entries. Likewise, full eigen decomposition of a dense $m \times m$ consensus matrix costs $\mathcal{O}(m^3)$; a thin union of width $q_t \ll m$ reduces this to $\mathcal{O}(mq_t^2)$ and represents the matrix with mq_t rather than m^2 entries. These comparisons concern conventional dense decompositions and stored matrix representations, not measured runtime speedups.

G Further Experiments

G.1 Ablation for FedCURE Components

Table 8 examines the contribution of each FedCURE component. ECF alone already provides a strong safety–utility trade-off compared with the baselines in Table 2, showing that ECF is effective even without additional ERA or RRA. Combining ECF with ERA and RRA further improves the overall results, indicating that ERA, ECF, and RRA play complementary roles in stabilizing local updates, filtering shared erosion-related components, and correcting residual safety degradation, respectively.

Table 8: Component ablations on **Meta-Llama-3-8B-Instruct** across Commonsense and Dolly. RRA uses $c = 0.1$ when enabled. All deltas are estimated relative to FedIT, $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$; task and utility deltas likewise subtract the corresponding FedIT scores. $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$,

	Dolly							Commonsense						
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	48.67	0.00	39.83	0.00	58.93	0.00	0.00	25.10	0.00	85.39	0.00	49.27	0.00	0.00
ECF-only	14.56	−34.11	39.47	−0.36	66.75	+7.82	+41.57	9.12	−15.98	85.51	+0.13	54.18	+4.91	+21.01
ERA-only	4.13	−44.54	37.49	−2.33	65.55	+6.62	+48.83	6.41	−18.68	84.99	−0.40	53.32	+4.05	+22.33
RRA-only	5.93	−42.74	39.63	−0.20	61.02	+2.09	+44.64	3.23	−21.87	84.71	−0.68	62.44	+13.17	+34.36
ERA-RRA	2.44	−46.23	39.77	−0.06	66.87	+7.94	+54.11	3.38	−21.72	84.79	−0.60	64.56	+15.29	+36.41
ECF-RRA	5.33	−43.34	38.09	−1.74	67.50	+8.57	+50.17	1.69	−23.41	84.46	−0.93	67.08	+17.81	+40.29
FedCURE (ERA-ECF-RRA)	2.03	−46.64	38.62	−1.21	68.21	+9.28	**+54.70**	1.58	−23.52	84.36	−1.03	67.54	+18.27	**+40.76**

As specified in Section A, we use an RRA merge scale of $c = 0.1$ by default to preserve task performance. To examine this choice, Table 9 varies the RRA merge scale while keeping the other FedCURE components fixed. Increasing c consistently strengthens safety restoration but gradually degrades task performance, revealing a clear safety-task trade-off. We therefore adopt $c = 0.1$ as the default setting to achieve a balanced improvement in safety while minimizing degradation of downstream task performance.

Table 9: RRA merge-scale ablation on Meta-Llama-3-8B-Instruct fine-tuned on Dolly. ERA, ECF, and RRA are enabled in all FedCURE variants. All deltas use the reported two-decimal scores relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$; task deltas likewise subtract the corresponding FedIT score ($\Delta_{\text{Task}} = \text{R-L} - \text{R-L}_{\text{FedIT}}$).

	Dolly			
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}
FedIT	48.67	0.00	39.83	0.00
FedCURE ($c = 0.1$)	2.03	−46.64	**38.62**	**−1.21**
FedCURE ($c = 0.25$)	1.58	−47.09	38.34	−1.49
FedCURE ($c = 0.5$)	**1.09**	**−47.58**	37.47	−2.36

G.2 Analysis of ERA-Alignment in Consensus

ERA-Aligned Candidate Selection. We first examine how different filtering policies behave when applied to the federated marginal update. As shown in Table 10, ECF constructs the consensus operator from client candidates positively aligned with the ERA and filters the resulting high-consensus modes with $\lambda \geq 0.5$. Random instead targets a random subspace within the current server marginal's output space, Uniform attenuates the entire marginal, and Negative-all removes the union of negatively ERA-aligned client candidates. At round 10, ECF attains 14.18% harmful score, compared with 17.28% for Random, 21.24% for Uniform, and 32.28% for Negative-all. This advantage persists over later rounds, with ECF yielding the lowest harmful score among these filtering policies at rounds 25 and 50. Taken as policy-level comparisons, these results consistently favor restricting the filtering candidates toward positively ERA-aligned, erosion-relevant update geometry rather than applying untargeted or oppositely aligned suppression.

Consensus within ERA-Aligned Candidates. We next examine how cross-client support should be used once the candidate space has been conditioned by ERA alignment. Low-consensus starts from the same positively ERA-aligned client candidates as ECF but targets the weakly shared modes, excluding numerical null modes, whereas ECF targets modes with strong cross-client support. As reported in Table 10, Low-consensus reaches 39.05% harmful score at round 10 and 61.57% at round 50, compared with 14.18% and 45.95% for ECF, respectively. This comparison is consistent with prioritizing strongly shared modes within the ERA-conditioned candidate space rather than suppressing the remaining weakly shared directions. In particular, the results suggest that erosion-relevant directions recurrently expressed across benign local updates form a consequential part of the federated update geometry.

Table 10: Filtering-policy ablation on Meta-Llama-3-8B-Instruct fine-tuned on Dolly, with client-side ERA conditioning and RRA correction disabled. FedIT denotes the unfiltered federated baseline.

	Round 10			**Round 25**			**Round 50**		
Policy	HS ↓	R-L ↑	Utility ↑	HS ↓	R-L ↑	Utility ↑	HS ↓	R-L ↑	Utility ↑
FedIT	48.67	39.83	58.93	53.08	38.12	59.68	64.63	34.65	54.05
Random	17.28	39.91	64.75	45.48	38.15	57.64	54.16	37.40	59.50
Uniform	21.24	**40.11**	64.07	47.55	38.02	57.96	58.52	37.13	57.92
Negative-all	32.28	38.76	62.61	44.79	**39.05**	59.34	57.92	**38.48**	57.62
Low-consensus	39.05	39.50	60.71	50.95	38.15	58.20	61.57	36.76	58.66
ECF (High-consensus)	**14.18**	39.68	**67.53**	**33.77**	38.74	**59.42**	**45.95**	37.51	**59.92**

ERA Conditioning of Cross-Client Consensus. To examine the respective roles of ERA alignment and cross-client consensus, we consider two targeted variants in Table 11. Consensus without ERA alignment constructs the consensus operator directly from the unaligned client-marginal candidate spaces and filters modes with $\lambda \geq 0.5$. ERA-aligned filtering without consensus provides the complementary variant, which starts from the positively ERA-aligned candidate spaces Q_i^+, further retains candidates with cosine similarity of at least 0.5 to the erosion reference, and removes their union without considering how consistently they are shared across clients.

Consensus without ERA alignment reduces HS to 2.97%, even below ECF's 14.18%, but its task-specific performance drops sharply to 27.06 R-L, compared with 39.83 for FedIT and 39.68 for ECF. This result indicates that consensus without ERA alignment is insufficiently selective filtering signal. Directions strongly shared across benign clients can also encode useful task structure, so generic consensus filtering can suppress task-relevant adaptation together with safety-eroding components. ERA alignment addresses this ambiguity by restricting the space in which consensus is evaluated to erosion-relevant candidate geometry.

Conversely, ERA-aligned filtering without consensus preserves task-specific performance at 40.11 R-L but reaches 31.11% harmful score, substantially above ECF's 14.18%. Thus, filtering within the ERA-aligned space without considering cross-client consensus is also insufficient. ERA alignment identifies a candidate space associated with erosion, but does not by itself determine which of these directions constitute recurrent federated structure. Importantly, ECF does not select candidates

according to large reference similarity: the sign of ERA alignment defines Q_i^+, after which cross-client support determines which directions are retained as high-consensus modes

Table 11: Component-wise analysis of ERA alignment and cross-client consensus at round 10 on Meta-Llama-3-8B-Instruct fine-tuned on Dolly. Consensus without ERA alignment filters high-consensus client-marginal modes without ERA alignment. ERA-aligned filtering without consensus removes the union of positively ERA-aligned client candidates without consensus filtering. ECF combines ERA-aligned candidate construction with high-consensus selection.

Policy	**ERA Align.**	**Consensus**	HS ↓	R-L ↑
FedIT	No	No	48.67	39.83
Consensus without ERA Alignment	No	Yes	2.97	27.06
ERA-aligned Filtering without Consensus	Yes	No	31.11	40.11
ECF	Yes	Yes	14.18	39.68

G.3 Robustness Across Multiple Random Seeds

We evaluate the reproducibility and seed robustness of FedCURE across three distinct random seeds$(17, 42, 73)$ on Dolly, Commonsense benchmark, and WizardLM using Meta-Llama-3-8B-Instruct. Tables 12, 13, and 14 report the mean and sample standard deviation across seed-specific runs, along with deltas evaluated relative to FedIT under each corresponding seed.

Across all evaluated datasets, FedCURE consistently achieves robust overall improvements, outperforming all baseline methods. Notably, FedCURE suppresses harmful scores with tight variance, demonstrating that its safety gains are seed-invariant rather than artifacts of favorable initialization or client sampling.

Table 12: Evaluation on Meta-Llama-3-8B-Instruct fine-tuned on Dolly (mean $\pm$ sample standard deviation over seeds 17, 42, and 73). Δ indicates changes relative to FedIT of the same seed: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	$49.40_{\pm 5.81}$	$0.00_{\pm 0.00}$	$39.79_{\pm 0.07}$	$0.00_{\pm 0.00}$	$59.29_{\pm 0.74}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
Multi-Krum	$46.13_{\pm 1.08}$	$-3.26_{\pm 5.83}$	$39.16_{\pm 0.29}$	$-0.63_{\pm 0.23}$	$60.33_{\pm 0.61}$	$+1.04_{\pm 0.28}$	$+3.68_{\pm 5.70}$
Trimmed Mean	$51.29_{\pm 3.43}$	$+1.89_{\pm 2.45}$	$39.60_{\pm 0.36}$	$-0.19_{\pm 0.39}$	$59.52_{\pm 0.56}$	$+0.24_{\pm 0.47}$	$-1.84_{\pm 2.62}$
SaLoRA	$37.24_{\pm 2.68}$	$-12.15_{\pm 4.39}$	$39.08_{\pm 0.31}$	$-0.71_{\pm 0.25}$	$60.19_{\pm 0.98}$	$+0.91_{\pm 1.71}$	$+12.35_{\pm 5.83}$
SafeLoRA	$45.41_{\pm 2.72}$	$-3.98_{\pm 3.18}$	$39.47_{\pm 0.15}$	$-0.32_{\pm 0.11}$	$60.02_{\pm 0.29}$	$+0.74_{\pm 0.99}$	$+4.40_{\pm 3.89}$
GuardSpace	$15.27_{\pm 4.12}$	$-34.12_{\pm 8.40}$	$37.44_{\pm 0.38}$	$-2.35_{\pm 0.35}$	$59.18_{\pm 1.58}$	$-0.11_{\pm 1.05}$	$+31.66_{\pm 8.02}$
Booster	$47.10_{\pm 3.21}$	$-2.29_{\pm 3.89}$	$39.13_{\pm 0.39}$	$-0.66_{\pm 0.46}$	$59.15_{\pm 0.76}$	$-0.14_{\pm 1.37}$	$+1.50_{\pm 5.45}$
FedLLM-Attack	$15.06_{\pm 3.82}$	$-34.34_{\pm 3.77}$	$31.07_{\pm 0.36}$	$-8.72_{\pm 0.40}$	$54.72_{\pm 1.91}$	$-4.57_{\pm 1.82}$	$+21.05_{\pm 4.92}$
FedCURE	$1.99_{\pm 0.10}$	$-47.41_{\pm 5.91}$	$38.79_{\pm 0.21}$	$-1.01_{\pm 0.20}$	$67.98_{\pm 0.20}$	$+8.70_{\pm 0.87}$	$\mathbf{+55.10}_{\pm 6.51}$

Table 13: Evaluation on Meta-Llama-3-8B-Instruct fine-tuned on Commonsense (mean $\pm$ sample standard deviation over seeds 17, 42, and 73). Δ indicates changes relative to FedIT of the same seed: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

Method	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	$21.03_{\pm 4.02}$	$0.00_{\pm 0.00}$	$85.33_{\pm 0.06}$	$0.00_{\pm 0.00}$	$46.83_{\pm 5.56}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
Multi-Krum	$20.57_{\pm 2.74}$	$-0.46_{\pm 2.12}$	$84.12_{\pm 0.72}$	$-1.21_{\pm 0.67}$	$47.98_{\pm 3.63}$	$+1.16_{\pm 2.47}$	$+0.41_{\pm 2.18}$
Trimmed Mean	$20.46_{\pm 4.84}$	$-0.57_{\pm 1.27}$	$85.52_{\pm 0.08}$	$+0.19_{\pm 0.04}$	$45.69_{\pm 5.27}$	$-1.14_{\pm 0.81}$	$-0.38_{\pm 1.63}$
SaLoRA	$13.92_{\pm 4.52}$	$-7.11_{\pm 4.47}$	$85.77_{\pm 0.21}$	$+0.44_{\pm 0.16}$	$44.71_{\pm 1.28}$	$-2.12_{\pm 6.42}$	$+5.43_{\pm 9.14}$
SafeLoRA	$10.60_{\pm 1.25}$	$-10.43_{\pm 3.70}$	$85.24_{\pm 0.26}$	$-0.10_{\pm 0.31}$	$45.95_{\pm 3.37}$	$-0.87_{\pm 6.42}$	$+9.46_{\pm 4.88}$
GuardSpace	$34.72_{\pm 2.83}$	$+13.69_{\pm 2.45}$	$36.70_{\pm 0.73}$	$-48.64_{\pm 0.78}$	$26.56_{\pm 0.69}$	$-20.26_{\pm 5.82}$	$-82.59_{\pm 3.97}$
Booster	$16.68_{\pm 3.13}$	$-4.35_{\pm 1.70}$	$84.58_{\pm 0.45}$	$-0.76_{\pm 0.43}$	$50.54_{\pm 1.60}$	$+3.72_{\pm 4.77}$	$+7.31_{\pm 5.80}$
FedLLM-Attack	$16.36_{\pm 4.32}$	$-4.67_{\pm 6.67}$	$75.17_{\pm 0.82}$	$-10.17_{\pm 0.78}$	$55.90_{\pm 0.75}$	$+9.07_{\pm 4.82}$	$+3.57_{\pm 2.46}$
FedCURE	$2.09_{\pm 0.53}$	$-18.94_{\pm 4.54}$	$84.58_{\pm 0.22}$	$-0.75_{\pm 0.27}$	$67.58_{\pm 0.24}$	$+20.75_{\pm 5.35}$	$\mathbf{+38.94}_{\pm 2.97}$

Table 14: Evaluation on Meta-Llama-3-8B-Instruct fine-tuned on WizardLM (mean $\pm$ sample standard deviation over seeds 17, 42, and 73). Δ indicates changes relative to FedIT of the same seed: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

Method	HS ↓	Δ_{HS}	MT-B ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	$4.77_{\pm 1.48}$	$0.00_{\pm 0.00}$	$6.865_{\pm 0.109}$	$0.000_{\pm 0.000}$	$65.57_{\pm 0.71}$	$0.00_{\pm 0.00}$	$0.000_{\pm 0.000}$
Multi-Krum	$4.34_{\pm 1.13}$	$-0.42_{\pm 0.54}$	$6.899_{\pm 0.124}$	$+0.034_{\pm 0.045}$	$66.14_{\pm 0.45}$	$+0.57_{\pm 0.72}$	$+1.029_{\pm 1.082}$
Trimmed Mean	$3.55_{\pm 0.45}$	$-1.21_{\pm 1.91}$	$6.770_{\pm 0.058}$	$-0.096_{\pm 0.126}$	$66.05_{\pm 0.71}$	$+0.48_{\pm 0.74}$	$+1.599_{\pm 2.600}$
SaLoRA	$3.18_{\pm 0.62}$	$-1.59_{\pm 0.97}$	$6.967_{\pm 0.085}$	$+0.102_{\pm 0.084}$	$65.88_{\pm 0.47}$	$+0.31_{\pm 0.27}$	$+1.997_{\pm 1.070}$
SafeLoRA	$4.09_{\pm 0.74}$	$-0.68_{\pm 0.94}$	$6.809_{\pm 0.052}$	$-0.057_{\pm 0.159}$	$66.75_{\pm 0.42}$	$+1.18_{\pm 0.79}$	$+1.802_{\pm 1.425}$
GuardSpace	$4.31_{\pm 0.36}$	$-0.46_{\pm 1.39}$	$6.692_{\pm 0.063}$	$-0.173_{\pm 0.108}$	$59.09_{\pm 1.42}$	$-6.48_{\pm 1.91}$	$-6.190_{\pm 2.331}$
Booster	$5.24_{\pm 1.28}$	$+0.47_{\pm 0.21}$	$6.669_{\pm 0.082}$	$-0.197_{\pm 0.100}$	$63.02_{\pm 0.68}$	$-2.55_{\pm 0.59}$	$-3.215_{\pm 0.595}$
FedLLM-Attack	$16.94_{\pm 0.74}$	$+12.17_{\pm 2.06}$	$6.258_{\pm 0.118}$	$-0.607_{\pm 0.152}$	$58.24_{\pm 2.43}$	$-7.33_{\pm 1.91}$	$-20.112_{\pm 3.093}$
FedCURE	$0.33_{\pm 0.02}$	$-4.44_{\pm 1.46}$	$6.975_{\pm 0.078}$	$+0.110_{\pm 0.031}$	$67.90_{\pm 0.34}$	$+2.33_{\pm 1.03}$	$\mathbf{+6.876}_{\pm 2.189}$

G.4 Robustness to the Erosion Reference Data

FedCURE constructs its two reference adapters, ERA and RRA, once offline on the server from a safety dataset and keeps them fixed throughout federated fine-tuning. To evaluate whether its performance is tied to a particular safety dataset, we rebuild both adapters from the LAT dataset (Sheshadri et al., 2025), following the prestage procedure in Section 4 under the same conditions as the original Circuit Breakers variant. ERA is trained by supervised fine-tuning on 5K harmful prompt-compliant response pairs, and RRA on 2.5K harmful prompt-refusal pairs combined with 2.5K Alpaca dataset.

As shown in Table 15, FedCURE with LAT-based reference adapters closely matches the Circuit Breakers variant on all three datasets. Consequently, the LAT variant still achieves lowest harmful scores than all baselines, together with competitive task performance and higher general utility.

Table 15: Evaluation on Meta-Llama-3-8B-Instruct across Dolly, Commonsense, and WizardLM. $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$ measures net improvement relative to FedIT.

	Dolly				Commonsense				WizardLM			
Method	HS ↓	R-L ↑	Utility ↑	Δ_{Overall}	HS ↓	Acc ↑	Utility ↑	Δ_{Overall}	HS ↓	MT-B ↑	Utility ↑	Δ_{Overall}
FedIT	48.67	39.83	58.93	0.00	25.10	85.39	49.27	0.00	3.68	6.879	65.23	0.000
Multi-Krum	44.89	39.19	60.29	+4.50	22.24	84.94	48.18	+1.32	3.87	6.863	65.63	+0.194
Trimmed Mean	50.35	39.19	59.69	−1.56	24.67	85.58	47.19	−1.46	3.72	6.703	65.24	−0.206
SaLoRA	34.80	39.04	60.85	+15.00	13.80	86.01	45.45	+8.10	2.52	6.900	65.76	+1.711
SafeLoRA	45.67	39.63	60.31	+4.18	11.74	84.94	49.59	+13.23	3.98	6.779	66.30	+0.670
GuardSpace	11.19	37.88	59.48	+36.08	37.95	35.85	25.77	−85.89	3.91	6.763	60.72	−4.856
Booster	44.47	38.94	59.98	+4.36	18.90	84.65	52.33	+8.52	4.24	6.756	63.36	−2.553
FedLLM-Attack	11.90	31.31	56.35	+25.67	17.43	76.06	56.16	+5.23	16.83	6.125	55.77	−23.364
FedCURE (Circuit Breaker)	2.03	38.62	68.21	**+54.71**	1.58	84.36	67.54	**+40.76**	0.30	6.981	67.89	**+6.142**
FedCURE (LAT)	2.33	36.70	69.77	**+54.05**	2.67	83.94	67.36	**+39.07**	0.34	7.019	67.16	**+5.410**

G.5 Analysis of Over-Refusal and Safety Trade-offs

The main results show that FedCURE substantially reduces harmful scores while maintaining competitive task performance and general utility (Tables 2 and 3). We further examine whether these safety gains are merely a by-product of excessive refusal on harmless inputs. To this end, we evaluate over-refusal on 250 harmless XSTest prompts (Röttger et al., 2024) for Meta-Llama-3-8B-Instruct and Qwen2.5-7B-Instruct, after federated fine-tuning on Dolly, Commonsense, and WizardLM.

To compare harmful responses and over-refusal jointly, we define an equally weighted summary relative to FedIT: $\Delta_{\text{Safe}} = (\text{HS} - \text{HS}_{\text{FedIT}}) + (\text{O-R} - \text{O-R}_{\text{FedIT}})$, where both harmful score and over-refusal are expressed as percentages. Lower values indicate a more favorable combined trade-off, with negative values representing an improvement over FedIT. We report both individual metrics alongside this summary to distinguish reductions in harmful responses from changes in harmless-prompt refusal.

As shown in Tables 16 and 17, FedCURE achieves negative Δ_{Safe} values across all evaluated model–dataset combinations. For Llama and Qwen, it attains the lowest Δ_{Safe} among the compared methods on all three datasets. In particular, on Dolly and Commonsense, FedCURE combines low harmful scores with over-refusal rates of 4.4%–6.4% for these two models. Other methods exhibit less favorable combinations: several retain low over-refusal rates but leave substantially higher harmful scores, while FedLLM-Attack reduces harmful scores on Dolly relative to FedIT at the cost of higher over-refusal. For example, its over-refusal rates reach 16.8% and 20.8% for Llama and Qwen, compared with 4.4% and 6.4% for FedCURE.

Together with the task-performance and general-utility results, these observations suggest that FedCURE's safety gains are not explained solely by indiscriminate refusal, while the balance between safety preservation and over-refusal remains model- and dataset-dependent.

Table 16: Safety trade-off of Meta-Llama-3-8B-Instruct. HS: harmful score; O-R: over-refusal on harmless XSTest prompts (%; lower is better). $\Delta_{\text{Safe}} = (\text{HS} - \text{HS}_{\text{FedIT}}) + (\text{O-R} - \text{O-R}_{\text{FedIT}})$; lower is better.

	Dolly			Commonsense			WizardLM		
Method	HS ↓	O-R ↓	Δ_{Safe} ↓	HS ↓	O-R ↓	Δ_{Safe} ↓	HS ↓	O-R ↓	Δ_{Safe} ↓
FedIT	48.67	2.80	0.00	25.10	10.40	0.00	3.68	19.60	0.00
Multi-Krum	44.89	3.60	−2.98	22.24	16.00	+2.74	3.87	18.00	−1.41
Trimmed Mean	50.35	2.00	+0.88	24.67	11.60	+0.77	3.72	21.60	+2.04
SaLoRA	34.80	1.20	−15.47	13.80	42.40	+20.70	2.52	26.40	+5.64
SafeLoRA	45.67	1.60	−4.20	11.74	5.60	−18.16	3.98	23.20	+3.90
GuardSpace	11.19	2.00	−38.28	37.95	87.60	+90.05	3.91	18.40	−0.97
Booster	44.47	4.80	−2.20	18.90	10.40	−6.20	4.24	22.40	+3.36
FedLLM-Attack	11.90	16.80	−22.77	17.43	14.40	−3.67	16.83	20.00	+13.55
FedCURE (w/o ERA&RRA)	14.18	1.20	−36.09	9.12	3.20	−23.18	1.95	18.80	−2.53
FedCURE	2.03	4.40	**−45.04**	1.58	5.60	**−28.32**	0.30	17.20	**−5.78**

Table 17: Safety trade-off of Qwen2.5-7B-Instruct. HS: harmful score; O-R: over-refusal on harmless XSTest prompts (%; lower is better). $\Delta_{\text{Safe}} = (\text{HS} - \text{HS}_{\text{FedIT}}) + (\text{O-R} - \text{O-R}_{\text{FedIT}})$; lower is better.

	Dolly			Commonsense			WizardLM		
Method	HS ↓	O-R ↓	Δ_{Safe} ↓	HS ↓	O-R ↓	Δ_{Safe} ↓	HS ↓	O-R ↓	Δ_{Safe} ↓
FedIT	31.98	4.00	0.00	11.12	3.60	0.00	2.74	33.20	0.00
Multi-Krum	26.69	4.00	−5.29	13.31	3.20	+1.79	3.56	31.60	−0.78
Trimmed Mean	31.25	2.80	−1.93	11.61	3.20	+0.09	2.74	32.80	−0.40
SaLoRA	17.01	3.20	−15.77	8.10	44.40	+37.78	2.21	24.00	−9.73
SafeLoRA	30.48	3.60	−1.90	10.44	2.40	−1.88	2.89	28.00	−5.05
GuardSpace	12.97	4.40	−18.61	14.82	92.80	+92.90	4.39	14.00	−17.55
Booster	31.16	2.80	−2.02	11.35	3.20	−0.17	2.89	32.40	−0.65
FedLLM-Attack	9.13	20.80	−6.05	13.86	16.80	+15.94	8.11	20.40	−7.43
FedCURE	3.23	6.40	**−26.35**	2.86	6.40	**−5.46**	1.05	15.20	**−19.69**

G.6 Evaluation on Diverse Model Families Across Benchmarks

Table 18 reports evaluation results on Meta-Llama-3-8B-Instruct fine-tuned on WizardLM. And Tables 19 and 20 report evaluation results on Gemma-2-9B-it and Qwen2.5-7B-Instruct fine-tuned on Commonsense and WizardLM, respectively. Consistent with the observations on Dolly (Table 3), FedCURE consistently outperforms all competing baselines across both model families, achieving the lowest harmful score while reliably preserving task performance and general utility.

Table 18: Evaluation on Meta-Llama-3-8B-Instruct fine-tuned on WizardLM. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

Method	HS ↓	Δ_{HS}	MT-B ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	3.68	0.00	6.879	0.000	65.23	0.00	0.000
Multi-Krum	3.87	+0.19	6.863	−0.016	65.63	+0.40	+0.194
Trimmed Mean	3.72	+0.04	6.703	−0.176	65.24	+0.01	−0.206
SaLoRA	2.52	−1.16	6.900	+0.021	65.76	+0.53	+1.711
SafeLoRA	3.98	+0.30	6.779	−0.100	66.30	+1.07	+0.670
GuardSpace	3.91	+0.23	6.763	−0.116	60.72	−4.51	−4.856
Booster	4.24	+0.56	6.756	−0.123	63.36	−1.87	−2.553
FedLLM-Attack	16.83	+13.15	6.125	−0.754	55.77	−9.46	−23.364
FedCURE	0.30	−3.38	6.981	+0.102	67.89	+2.66	**+6.142**

Table 19: Evaluation on Commonsense across Gemma-2-9B-it and Qwen2.5-7B-Instruct. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Gemma-2-9B-it							**Qwen2.5-7B-Instruct**						
Method	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	1.43	0.00	88.45	0.00	43.45	0.00	0.00	11.12	0.00	86.03	0.00	70.29	0.00	0.00
Multi-Krum	1.73	+0.30	88.61	+0.16	46.70	+3.25	+3.11	13.31	+2.19	85.29	−0.74	70.04	−0.25	−3.18
Trimmed Mean	1.28	−0.15	88.55	+0.10	44.88	+1.43	+1.68	11.61	+0.49	85.85	−0.18	69.78	−0.51	−1.18
SaLoRA	2.25	+0.82	88.69	+0.24	51.69	+8.24	+7.66	8.10	−3.02	86.76	+0.73	56.43	−13.86	−10.11
SafeLoRA	2.21	+0.78	88.65	+0.20	54.53	+11.08	+10.50	10.44	−0.68	85.73	−0.30	72.41	+2.12	+2.50
GuardSpace	27.54	+26.11	85.93	−2.52	26.37	−17.08	−45.71	14.82	+3.70	36.89	−49.14	25.09	−45.20	−98.04
Booster	0.94	−0.49	88.41	−0.04	38.10	−5.35	−4.90	11.35	+0.23	86.14	+0.11	64.89	−5.40	−5.52
FedLLM-Attack	7.67	+6.24	82.45	−6.00	55.41	+11.96	−0.28	13.86	+2.74	77.38	−8.65	64.88	−5.41	−16.80
FedCURE	1.16	−0.27	87.80	−0.65	74.52	+31.07	**+30.69**	2.86	−8.26	84.33	−1.70	75.33	+5.04	**+11.60**

Table 20: Evaluation on WizardLM using Gemma-2-9B-it and Qwen2.5-7B-Instruct. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Gemma-2-9B-it							**Qwen2.5-7B-Instruct**						
Method	HS ↓	Δ_{HS}	MT-B ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	MT-B ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	3.75	0.00	6.838	0.000	71.88	0.00	0.000	2.74	0.00	6.681	0.000	72.47	0.00	0.000
Multi-Krum	3.57	−0.18	6.838	0.000	71.87	−0.01	+0.170	3.56	+0.82	6.781	+0.100	71.86	−0.61	−1.330
Trimmed Mean	3.34	−0.41	6.850	+0.012	71.60	−0.28	+0.142	2.74	0.00	6.738	+0.057	72.60	+0.13	+0.187
SaLoRA	3.38	−0.37	6.800	−0.038	72.71	+0.83	+1.162	2.21	−0.53	7.175	+0.494	72.54	+0.07	+1.094
SafeLoRA	4.43	+0.68	6.656	−0.182	71.26	−0.62	−1.482	2.89	+0.15	6.831	+0.150	71.87	−0.60	−0.600
GuardSpace	1.91	−1.84	7.044	+0.206	73.16	+1.28	+3.326	4.39	+1.65	6.900	+0.219	68.06	−4.41	−5.841
Booster	3.75	+0.00	6.706	−0.132	70.58	−1.30	−1.432	2.89	+0.15	6.744	+0.063	72.08	−0.39	−0.477
FedLLM-Attack	9.48	+5.73	6.694	−0.144	58.61	−13.27	−19.144	8.11	+5.37	6.600	−0.081	64.64	−7.83	−13.281
FedCURE	0.30	**−3.45**	7.425	**+0.587**	76.54	**+4.66**	**+8.697**	1.05	**−1.69**	7.688	**+1.007**	76.32	**+3.85**	**+6.547**

G.7 Scalability Across Client Cohort Sizes

Tables 21 and 22 report evaluation results on Meta-Llama-3-8B-Instruct under a scaled federation of 32 clients across Dolly, Commonsense, and WizardLM. Consistent with the primary 8 clients setup (Tables 2 and 18), FedCURE consistently outperforms all competing baselines, achieving the lowest harmful score while reliably preserving task performance and general utility. This demonstrates that FedCURE's consensus-guided filtering remains robust and client-scale invariant, effectively isolating shared erosion directions even under broader client distributions.

Table 21: Evaluation on Meta-Llama-3-8B-Instruct with 32 clients (Dirichlet $\alpha = 0.5$) across Dolly and Commonsense. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Dolly							Commonsense						
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	35.06	0.00	38.67	0.00	63.04	0.00	0.00	8.63	0.00	84.81	0.00	60.20	0.00	0.00
Multi-Krum	23.71	−11.35	37.29	−1.38	65.02	+1.98	+11.95	4.47	−4.16	83.71	−1.10	65.74	+5.54	+8.60
Trimmed Mean	32.09	−2.97	37.90	−0.77	65.19	+2.15	+4.35	5.03	−3.60	84.56	−0.25	63.28	+3.08	+6.43
SaLoRA	30.35	−4.71	39.41	+0.74	66.25	+3.21	+8.66	27.14	+18.51	85.44	+0.63	47.06	−13.14	−31.02
SafeLoRA	34.27	−0.79	37.57	−1.10	62.84	−0.20	−0.51	4.80	−3.83	84.34	−0.47	55.09	−5.11	−1.75
GuardSpace	25.04	−10.02	38.72	+0.05	62.69	−0.35	+9.72	26.50	+17.87	37.24	−47.57	26.94	−33.26	−98.70
Booster	22.86	−12.20	38.30	−0.37	64.66	+1.62	+13.45	4.05	−4.58	84.09	−0.72	59.36	−0.84	+3.02
FedLLM-Attack	15.78	−19.28	32.92	−5.75	58.65	−4.39	+9.14	16.22	+7.59	74.24	−10.57	54.39	−5.81	−23.97
FedCURE	1.31	−33.75	37.27	−1.40	70.76	+7.72	**+40.07**	1.69	−6.94	82.64	−2.17	69.58	+9.38	**+14.15**

Table 22: Evaluation on Meta-Llama-3-8B-Instruct with 32 clients (Dirichlet $\alpha = 0.5$) fine-tuned on WizardLM. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

Method	HS ↓	Δ_{HS}	MT-B ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	3.60	0.00	6.869	0.000	68.73	0.00	0.000
Multi-Krum	3.38	−0.22	6.838	−0.031	68.84	+0.11	+0.299
Trimmed Mean	3.49	−0.11	6.569	−0.300	66.71	−2.02	−2.210
SaLoRA	5.11	+1.51	7.050	+0.181	69.10	+0.37	−0.959
SafeLoRA	3.30	−0.30	6.844	−0.025	68.22	−0.51	−0.235
GuardSpace	14.54	+10.94	1.438	−5.431	25.90	−42.83	−59.201
Booster	3.00	−0.60	6.863	−0.006	64.52	−4.21	−3.616
FedLLM-Attack	17.61	+14.01	6.688	−0.181	57.49	−11.24	−25.431
FedCURE	0.53	−3.07	6.981	+0.112	69.79	+1.06	**+4.242**

G.8 Robustness Across Multiple Dirichlet Distributions

To assess robustness against statistical heterogeneity across clients, we evaluate FedCURE under non-IID partitions using Dirichlet label distributions for Dolly, Commonsense benchmark, and WizardLM dataset. Specifically, Table 23 is the evaluation result under dirichlet distribution $\alpha = 0.1$, and Tables 24 and 25 are the evaluation results under dirichlet distribution $\alpha = 0.9$. Across both settings, FedCURE consistently achieves the lowest harmful score (e.g., reducing HS to below 2.1% in all cases) while preserving downstream task accuracy and general utility. This demonstrates that FedCURE's consensus-guided filtering does not rely on uniform client distributions and remains highly effective under severe non-IID environments.

Table 23: Evaluation on Meta-Llama-3-8B-Instruct (8 clients, Dirichlet $\alpha = 0.1$) across Dolly and Commonsense. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Dolly							Commonsense						
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	40.73	0.00	39.14	0.00	60.47	0.00	0.00	19.30	0.00	85.19	0.00	50.34	0.00	0.00
Multi-Krum	41.35	+0.62	35.32	−3.82	59.31	−1.16	−5.60	8.07	−11.23	82.75	−2.44	60.18	+9.84	+18.63
Trimmed Mean	43.25	+2.52	38.94	−0.20	59.61	−0.86	−3.58	10.77	−8.53	85.41	+0.22	55.49	+5.15	+13.90
SaLoRA	31.99	−8.74	39.29	+0.15	61.99	+1.52	+10.41	31.66	+12.36	84.74	−0.45	45.46	−4.88	−17.69
SafeLoRA	42.05	+1.32	39.14	0.00	61.19	+0.72	−0.60	9.94	−9.36	85.09	−0.10	56.55	+6.21	+15.47
GuardSpace	16.45	−24.28	36.65	−2.49	59.15	−1.32	+20.47	12.05	−7.25	33.51	−51.68	26.89	−23.45	−67.88
Booster	40.51	−0.22	38.87	−0.27	60.72	+0.25	+0.20	12.93	−6.37	85.05	−0.14	52.01	+1.67	+7.90
FedLLM-Attack	13.56	−27.17	32.64	−6.50	59.32	−1.15	+19.52	14.08	−5.22	77.06	−8.13	53.42	+3.08	+0.17
FedCURE	1.84	−38.89	38.76	−0.38	68.96	+8.49	**+47.00**	1.50	−17.80	83.85	−1.34	68.54	+18.20	**+34.66**

Table 24: Evaluation on Meta-Llama-3-8B-Instruct (8 clients, Dirichlet $\alpha = 0.9$) across Dolly and Commonsense. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Dolly							Commonsense						
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	55.50	0.00	39.37	0.00	58.14	0.00	0.00	15.85	0.00	85.24	0.00	43.32	0.00	0.00
Multi-Krum	56.18	+0.68	39.27	−0.10	61.13	+2.99	+2.21	15.51	−0.34	84.41	−0.83	44.15	+0.83	+0.34
Trimmed Mean	57.24	+1.74	39.95	+0.58	59.30	+1.16	0.00	16.08	+0.23	85.59	+0.35	45.57	+2.25	+2.37
SaLoRA	39.28	−16.22	38.45	−0.92	59.38	+1.24	+16.54	18.02	+2.17	86.21	+0.97	45.17	+1.85	+0.65
SafeLoRA	52.87	−2.63	39.05	−0.32	59.38	+1.24	+3.55	12.16	−3.69	85.16	−0.08	47.18	+3.86	+7.47
GuardSpace	29.14	−26.36	37.53	−1.84	58.93	+0.79	+25.31	37.84	+21.99	36.96	−48.28	27.08	−16.24	−86.51
Booster	52.28	−3.22	39.88	+0.51	57.91	−0.23	+3.50	23.69	+7.84	85.01	−0.23	49.77	+6.45	−1.62
FedLLM-Attack	14.96	−40.54	33.52	−5.85	57.23	−0.91	+33.78	16.38	+0.53	75.06	−10.18	54.31	+10.99	+0.28
FedCURE	2.03	−53.47	39.01	−0.36	68.61	+10.47	**+63.58**	1.69	−14.16	84.40	−0.84	67.74	+24.42	**+37.74**

Table 25: Evaluation on Meta-Llama-3-8B-Instruct (8 clients, Dirichlet $\alpha = 0.9$) fine-tuned on WizardLM. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

Method	HS ↓	Δ_{HS}	MT-B ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	3.42	0.00	6.794	0.000	64.28	0.00	0.000
Multi-Krum	5.18	+1.76	6.919	+0.125	67.17	+2.89	+1.255
Trimmed Mean	5.93	+2.51	6.894	+0.100	66.27	+1.99	−0.420
SaLoRA	3.57	+0.15	6.850	+0.056	64.66	+0.38	+0.286
SafeLoRA	5.97	+2.55	6.684	−0.110	66.12	+1.84	−0.820
GuardSpace	3.75	+0.33	6.663	−0.131	60.28	−4.00	−4.461
Booster	4.02	+0.60	6.731	−0.063	63.12	−1.16	−1.823
FedLLM-Attack	17.66	+14.24	6.450	−0.344	55.74	−8.54	−23.124
FedCURE	0.26	−3.16	6.880	+0.086	66.09	+1.81	**+5.056**

G.9 Scalability Across Model Parameter Sizes

To examine whether FedCURE scales effectively to larger parameter architectures, we extend our evaluation to Llama-2-13B-chat-hf across Dolly and Commonsense (Tables 26). Across the benchmarks, FedCURE consistently achieves the lowest harmful score while reliably preserving downstream task accuracy and general utility. These results confirm that our cross-client consensus filtering and reference stabilization mechanisms remain robust and effective when applied to larger foundation models.

Table 26: Evaluation on Llama-2-13B-chat-hf (8 clients, Dirichlet $\alpha = 0.5$) across Dolly and Commonsense. Δ indicates changes relative to FedIT: $\Delta_{\text{HS}} = \text{HS} - \text{HS}_{\text{FedIT}}$, while task and utility deltas subtract the corresponding FedIT scores. Since lower HS is better, $\Delta_{\text{Overall}} = -\Delta_{\text{HS}} + \Delta_{\text{Task}} + \Delta_{\text{Utility}}$.

	Dolly							Commonsense						
Method	HS ↓	Δ_{HS}	R-L ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}	HS ↓	Δ_{HS}	Acc ↑	Δ_{Task}	Utility ↑	Δ_{Utility}	Δ_{Overall}
FedIT	14.42	0.00	38.02	0.00	51.57	0.00	0.00	5.15	0.00	82.48	0.00	46.62	0.00	0.00
Multi-Krum	11.84	−2.58	37.99	−0.03	50.85	−0.72	+1.83	5.56	+0.41	81.11	−1.37	45.94	−0.68	−2.46
Trimmed Mean	14.95	+0.53	38.13	+0.11	50.02	−1.55	−1.97	4.96	−0.19	82.19	−0.29	46.62	0.00	−0.10
SaLoRA	8.19	−6.23	38.50	+0.48	52.90	+1.33	+8.04	3.49	−1.66	83.24	+0.76	43.09	−3.53	−1.11
SafeLoRA	12.66	−1.76	37.62	−0.40	51.31	−0.26	+1.10	4.32	−0.83	82.26	−0.22	47.01	+0.39	+1.00
GuardSpace	9.28	−5.14	38.32	+0.30	50.12	−1.45	+3.99	4.17	−0.98	78.68	−3.80	29.87	−16.75	−19.57
Booster	29.52	+15.10	38.42	+0.40	47.86	−3.71	−18.41	8.52	+3.37	81.99	−0.49	35.21	−11.41	−15.27
FedLLM-Attack	14.50	+0.08	30.43	−7.59	42.25	−9.32	−16.99	13.29	+8.14	70.64	−11.84	38.57	−8.05	−28.03
FedCURE	2.29	−12.13	35.91	−2.11	52.22	+0.65	**+10.67**	1.50	−3.65	79.46	−3.02	50.76	+4.14	**+4.77**