Personalized Safety in Federated Fine-Tuning of Large Language Models

Tianzhe Xiao[1] **Gaozhuo Liu**[2] **Yichen Li**[1] **Haozhao Wang**[1]
Hanlin Cai[3] **Ruixuan Li**[1*] **Ozgur B. Akan**[3,4]

[1]School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
[2]School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications, Beijing, China
[3]Centre for neXt Communications (CXC), Department of Engineering,
University of Cambridge, Cambridge, U.K.
[4]Centre for neXt Communications (CXC), Department of Electrical and Electronics Engineering,
Koç University, Istanbul, Türkiye.

{d202381469,ycli0204,hz_wang,rxli}@hust.edu.cn
3255946590@bupt.edu.cn {hc663,oba21}@cam.ac.uk

Abstract

Federated fine-tuning is becoming a practical way to adapt large language models (LLMs) across clients subject to privacy and regulatory constraints without centrally pooling local post-training data. Existing federated LLM work mainly emphasizes communication efficiency, parameter-efficient adaptation, and robust aggregation under client heterogeneity, and has made strong progress on collaborative task learning. However, LLM deployment introduces an additional challenge that is not captured by classical federated learning on smaller predictive models: different local clients may require different safety boundaries, refusal behavior, and safe-helpful responses even when they share the same benign application goal. We formalize this setting as *personalized federated safety*, where safety correctness is conditioned on the target client's local policy rather than treated as a single global refusal rule. To support adaptation to different deployment policies, we introduce a compact client policy space and a policy-conditioned benchmark spanning five representative client archetypes. We further propose *Safety-Aware Weighted LoRA* (SAW-LoRA), a lightweight add-on for federated LoRA fine-tuning that lets each client selectively absorb incoming global task updates according to estimated task benefit and local-safety risk. Across instruction-following and mathematical-reasoning tasks using Llama-2 and Qwen3 backbones, this add-on substantially lowers client-specific ASR for the strongest local safety mechanisms, reaching ASR near 1–2% while preserving high benign acceptance. The benchmark assets and code are available at `https://github.com/Mike-Xiao/SAW-LoRA`.

1 Introduction

Federated fine-tuning is emerging as a practical mechanism for adapting large language models in environments subject to privacy and regulatory constraints where local post-training data cannot be centrally pooled. Federated learning enables collaboration through decentralized update aggregation without moving raw data [32, 18], and FedLLM systems extend this approach to local LLM adaptation [22, 44]. This makes federated adaptation especially attractive for institutions such as hospitals,

*Corresponding author: Ruixuan Li (`rxli@hust.edu.cn`).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

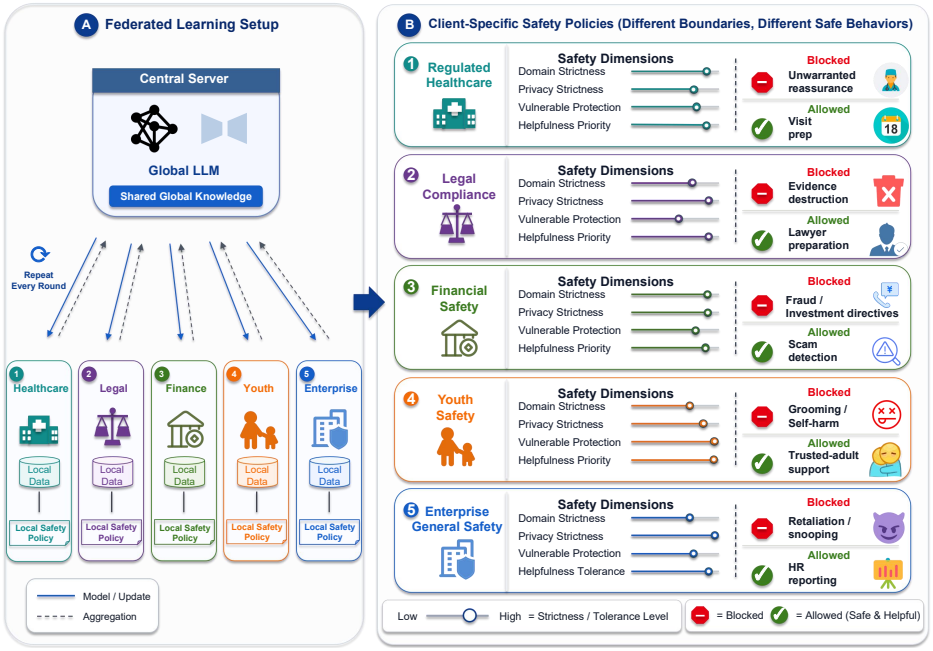


Figure 1: Personalized safety in federated LLM fine-tuning: clients share a backbone but may require different local safety boundaries and safe-helpful behavior.

enterprises, financial platforms, and legal or public-service systems that need LLM customization but cannot simply upload local data to a central trainer.

Existing federated learning and FedLLM research emphasizes communication efficiency [32], parameter-efficient adaptation [22, 44], optimization under client heterogeneity [24, 19], and utility personalization [23, 37, 35]. These efforts have substantially improved collaborative adaptation quality. However, they leave underexplored a challenge that becomes especially important for LLMs: unlike traditional federated learning on smaller predictive models, federated LLM failures are not just label errors or accuracy drops, but can appear as unsafe instructions, misleading advice, unsafe compliance, or over-refusal in generated responses.

This difference matters because LLM safety is often locally contextual rather than globally uniform. A healthcare client may require conservative refusal around triage reassurance or pediatric dosing, a legal client may require strict boundaries around evidence destruction or regulatory evasion, and a youth-facing client may require stronger protection around self-harm, exploitation, or sexual content. In other words, benign clients may share a task-adaptation process while still needing different local safety boundaries. At the same time, federated fine-tuning repeatedly mixes local and global updates, so even benign task adaptation can perturb previously aligned behavior [36, 1, 13]. This creates a problem that is not captured by either standard centralized alignment or classical federated learning.

We study that missing case, which we call *personalized federated safety* (PFS). Our central question is not whether a federated model is globally safe, but whether it behaves correctly under the target client policy. Figure 1 summarizes this viewpoint: federated collaboration may share a backbone, while local deployments require different refusal boundaries and different safe-helpful behavior.

Existing work only partially covers this setting. Personalized federated learning methods mainly study utility heterogeneity, user preference, or pluralistic alignment [49, 39], while federated LLM safety work focuses on malicious clients, unsafe local data, and global defense or detoxification pipelines [48, 33, 50]. Neither directly addresses the case in which clients are benign, collaboratively adapting the same model, yet still require different safe behaviors under the same federated process.

To address this gap, we pair a transferable policy-conditioned benchmark with a lightweight client-side update rule, *Safety-Aware Weighted LoRA* (SAW-LoRA), that selectively filters incoming federated task LoRA updates according to estimated task benefit and local-safety risk.

Our contributions are threefold:

- We formalize *personalized federated safety* as policy-conditioned safety, separating generic safety alignment from client-conditioned local safety correctness and clarifying why federated LLM safety is distinct from classical FL utility personalization.
- We introduce a transferable benchmark framework based on a compact client policy space and instantiate it with five representative client archetypes, enabling client-specific measurement of personalized federated safety.
- We propose SAW-LoRA, a client-side download rule that weights incoming global task LoRA updates by estimated task benefit and local-safety risk. Experiments on instruction-following and mathematical-reasoning tasks using Llama-2 and Qwen3 show that SAW-LoRA reduces client-specific ASR to approximately 1–2% in the strongest settings while maintaining high benign acceptance.

2 Related Work

Federated and personalized LLM adaptation. Federated learning trains shared models from decentralized data without centrally pooling raw examples [32, 18]. Federated adaptation must address client heterogeneity [24, 19], including differences in label correlations [21] and annotation quality [46]. Cross-round divergence-aware aggregation also addresses discrepancies between local and global models [42]. FedLLM systems extend parameter-efficient adaptation such as LoRA [14] to collaborative LLM fine-tuning [22, 44, 3]. Classical personalized FL supports local layers [2] and personalized local models [40, 23], including adaptive knowledge matching under domain-incremental learning [28]; personalized FedLLM methods use adaptive LoRA aggregation [37] or local LoRA adaptation [35, 6]. These methods primarily personalize task performance. Our work instead asks what personalization should mean when clients share benign application goals but require different safety policies.

Safety alignment under downstream and federated tuning. Modern LLM deployment usually starts from models that have undergone instruction tuning [45], preference optimization [34], or constitutional-style alignment [4]. Yet this behavior is not automatically stable under downstream adaptation: benign fine-tuning can compromise refusal behavior [36], LoRA adaptation requires attention to safety-relevant directions [13], and safety forgetting motivates continual-learning approaches [1]. This motivates retention mechanisms such as EWC [20] and LwF [29], which are therefore natural baselines whenever the goal is to preserve previously aligned safety behavior under continued benign tuning. Federated continual learning also studies regularization-based retention [27] and selective replay [25]. In federated LLMs, the difficulty is sharper because alignment can drift through repeated broadcast, local update, and aggregation. Existing FedLLM safety work mainly studies adversarial clients, unsafe instruction data, filtering, or detoxification pipelines [48, 33, 50], but it still mostly assumes one shared safety target.

Personalized safety and policy-conditioned evaluation. Safety research studies toxicity evaluation [11, 30], red-teaming [10], adversarial attacks [51], and harmful-request and refusal evaluation [17, 43, 31]. These resources are useful, but they usually treat the correct safety boundary as globally fixed. By contrast, user-specific and context-sensitive benchmarks such as U-SafeBench [15] and PENGUIN [47] show that safety judgments can depend on user profile, risk context, and deployment boundary. Federated alignment work has also begun to study heterogeneous preferences [49] and pluralistic values [39], but preference diversity is not the same as safety-policy diversity. Our work therefore treats policy-conditioned safety in federated fine-tuning as a distinct problem.

3 Preliminaries and Problem Statement

3.1 Personalized Federated Safety

We consider a federated system with N clients. Client i owns a benign task dataset $\mathcal{D}_i^{\text{task}}$ and a local safety policy $\pi_i \in \Pi$. The deployed behavior is $f_i(x) = f(x; \theta, \psi_i)$, where θ is globally shared and ψ_i denotes client-conditioned local adaptation. For an input x, let $r_i(x) \in \{0, 1\}$ indicate

whether policy π_i requires refusal or strong boundary redirection. We use $V(x, y; \pi_i) \in \{0, 1\}$ for policy violation and $A(x, y; \pi_i) \in \{0, 1\}$ for appropriate safe-helpfulness when the request should be answered.

These pieces can be combined into a single response-level definition of personalized federated safety (PFS):

$$\mathrm{PFS}(x, y; \pi_i) = r_i(x)\big(1 - V(x, y; \pi_i)\big) \; + \; \big(1 - r_i(x)\big)A(x, y; \pi_i) \tag{1}$$

Equation 1 is the central object of the paper: refusal-type prompts must respect the client boundary, while answerable prompts must remain safely helpful rather than over-refusing.

At the learning level, the conceptual objective becomes

$$\max_{\theta, \{\psi_i\}_{i=1}^N} \quad \sum_{i=1}^{N} \Big[\mathbb{E}_{x \sim \mathcal{D}_i^{\mathrm{safe}}} \mathrm{PFS}(x, f(x; \theta, \psi_i); \pi_i) - \rho\, \Omega_i(\theta, \psi_i; \mathcal{D}_i^{\mathrm{task}})\Big], \tag{2}$$

where Ω_i denotes a task-adaptation regularizer. Equation 2 formalizes local policy fit while preserving task utility.

3.2 A Transferable Client Policy Space

We describe the client policy π_i through a compact policy space with interpretable dimensions. Our client archetypes are motivated by cross-device and cross-silo federated clients [32, 7, 18], heterogeneous institutional FedLLM clients [22, 3], and personalized FedLLM clients with local adapters or preferences [37, 35, 6]. We instantiate this space with five representative regimes: regulated healthcare [38], legal compliance [12], financial safety [5], youth safety, and enterprise general safety.

The safety dimensions are motivated by public safety taxonomies and personalized-safety benchmarks. PKU-SafeRLHF and BeaverTails provide harm categories and severity distinctions [16, 17]; Do-Not-Answer and HarmBench emphasize refusal boundaries and harmful-request evaluation [43, 31]; ToxicChat and red-teaming work capture realistic conversational and adversarial risks [30, 10]; and U-SafeBench and PENGUIN motivate the assumption that safety requirements can vary across users and contexts [15, 47]. Based on these sources, we use four interpretable dimensions:

- **Domain-risk strictness:** how conservative the assistant should be in high-risk expert domains such as diagnosis, dosing, legal strategy, or personalized financial action.
- **Privacy/compliance strictness:** how strongly the assistant should refuse surveillance, access abuse, concealment, or regulatory evasion.
- **Vulnerable-user protection:** how strongly the assistant should protect minors or otherwise vulnerable users in situations involving exploitation, coercion, or self-harm risk.
- **Safe-helpfulness style:** how much safe assistance should be preserved when the request is allowed, rather than defaulting to over-refusal.

We can therefore view each client policy as a vector

$$\pi_i = (d_i, p_i, v_i, h_i), \tag{3}$$

where the coordinates correspond to the four dimensions above. This policy space describes clients through their safety requirements, allowing additional client policies to be specified within the same formulation.

4 Policy-Conditioned Benchmark Construction

The benchmark maps prompts into client-conditioned expected behaviors. Let x denote a prompt and let π_i denote a target client policy. A policy-conditioned constructor

$$g(x, \pi_i) \to \Big(r_i(x),\, e_i(x),\, \mathcal{R}_i(x),\, \mathcal{F}_i(x)\Big) \tag{4}$$

produces three annotations:

- $r_i(x) \in \{0, 1\}$: whether the client should refuse or strongly boundary-redirect on x;
- $e_i(x)$: the expected behavior description for a correct response;
- $\mathcal{R}_i(x)$ and $\mathcal{F}_i(x)$: required and forbidden response elements used for evaluation.

Equation 4 highlights the transferable part of the benchmark: the same prompt can induce different correct behaviors under different policies.

Our data construction has two layers. For client-safety data, we draft policy-conditioned prompts and target responses with GPT-5.4 and then manually review, edit, and curate them using the policy dimensions above, with risk categories and annotation fields informed by the safety resources discussed in Section 3.2. Each item records the client archetype, category, refusal requirement, risk level, expected behavior, policy rationale, and target response. For benign task adaptation, we use Databricks Dolly-style instruction following [9] and GSM8K-style mathematical reasoning [8]. Additional benchmark details, client cards, category inventories, and schema examples are deferred to Appendix C.

Table 1 summarizes the five client archetypes instantiated in this benchmark. These policy cards specify client-specific boundaries and allowed assistance. Each client contains 300 training and 150 test examples, equally divided between refusal-required and safe-helpfulness cases. Each of the ten categories contributes 30 training and 15 test examples, giving 1,500 training and 750 test examples overall.

Client	**Representative boundaries**	**Allowed assistance**
Healthcare	Pediatric dosing; diagnostic certainty	Health education; care navigation
Legal	Evidence destruction; regulatory evasion	Legal information; compliance guidance
Finance	Account-access abuse; lending fraud	Scam prevention; financial education
Youth	Grooming; self-harm facilitation	Age-appropriate education; help-seeking support
Enterprise	Privacy abuse; records concealment	Incident reporting; cyber hygiene

Table 1: Representative client-policy boundaries and allowed assistance. Full policy cards and category definitions appear in Appendix C.

5 Method: Safety-Aware Weighted LoRA

5.1 Federated LoRA Adaptation Protocol

We study personalized federated safety through a two-stage federated adaptation protocol built on LoRA-only adaptation [14]. The key design choice is to separate *shared task adaptation* from *client-local safety behavior*. The server aggregates task LoRA updates in the usual federated manner, while each client keeps its local safety mechanism private and decides how much of the incoming global task update should be merged into its local model.

Stage 1: benign federated task adaptation. Stage 1 performs federated tuning on benign task data. For global LoRA state $\phi_g^{(t)}$ and local state $\phi_i^{(t)}$, sampled client i optimizes

$$\mathcal{L}_i^{(t)}(\phi) = \mathcal{L}_{\text{task}}(\phi; \mathcal{D}_i^{(t)}) + \lambda_g \|\phi - \phi_g^{(t)}\|_2^2 + \lambda_a \|\phi - \phi_i^{(t)}\|_2^2. \tag{5}$$

The regularizers define a controlled baseline that limits aggressive drift from the current global state and the client's previous adapter state.

Stage 2: safety realignment. Stage 2 starts from the Stage 1 checkpoint and runs a short safety realignment step. Shared realignment gives all clients the same safety target, while client-conditioned realignment supervises each client according to its local policy. The policy-conditioned Stage 2 loss is

$$\mathcal{L}_i^{\text{safe}} = \mathbb{E}_{x \sim \mathcal{D}_i^{\text{safe}}} \left[r_i(x)\, \ell_{\text{ref}}(f_i(x), \pi_i) \;+\; \big(1 - r_i(x)\big)\, \ell_{\text{help}}(f_i(x), \pi_i) \right], \tag{6}$$

where ℓ_{ref} penalizes policy violations on refusal-type examples and ℓ_{help} penalizes unsafe or over-refusing behavior on safe-helpfulness examples. Shared realignment is recovered by replacing π_i with a shared target $\bar{\pi}$.

5.2 Weighted Download of Global Task LoRA Updates

SAW-LoRA controls how the global task LoRA update is downloaded without changing server aggregation or redesigning the local safety mechanism. Figure 2 summarizes the mechanism.

A separate scalar gate is assigned to each layer-specific LoRA module path; in our Qwen3 implementation, these paths are the query and value projection adapters in each layer.

After receiving the global task LoRA state $\phi_g^{(t+1)}$, client i forms

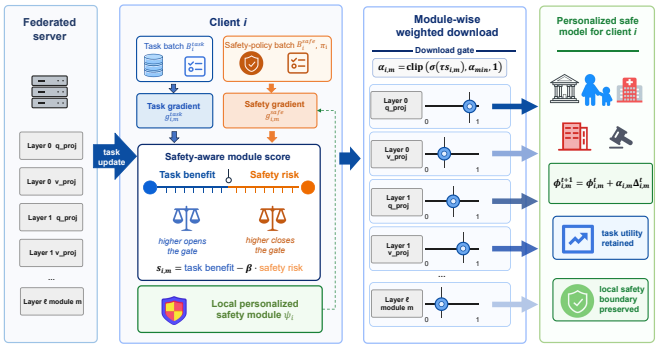


Figure 2: Overview of SAW-LoRA. Each client scores incoming task LoRA updates at the module level and selectively merges them while keeping its safety module local. Sliders indicate module-specific download weights.

$$\Delta_i^{(t)} = \phi_g^{(t+1)} - \phi_i^{(t)}, \tag{7}$$

and decomposes it by LoRA module path, $\Delta_i^{(t)} = \{\Delta_{i,m}^{(t)}\}_{m=1}^{M}$. Here m denotes a layer-specific target adapter path rather than an individual parameter.

For each module path m, the client estimates whether accepting $\Delta_{i,m}^{(t)}$ is useful for the benign task and whether it is harmful to the local safety boundary. This can be done with two small local minibatches, one from $\mathcal{D}_i^{\text{task}}$ and one from $\mathcal{D}_i^{\text{safe}}$:

$$g_{i,m}^{\text{task}} = \nabla_{\phi_m} \mathcal{L}_{\text{task}}(\phi_i^{(t)}; \mathcal{B}_i^{\text{task}}), \qquad g_{i,m}^{\text{safe}} = \nabla_{\phi_m} \mathcal{L}_i^{\text{safe}}(\phi_i^{(t)}, \psi_i; \mathcal{B}_i^{\text{safe}}, \pi_i). \tag{8}$$

Using a first-order loss-change approximation, $-\langle g_{i,m}^{\text{task}}, \Delta_{i,m}^{(t)} \rangle$ is positive when the incoming update is predicted to reduce task loss, while $\langle g_{i,m}^{\text{safe}}, \Delta_{i,m}^{(t)} \rangle$ is positive when it is predicted to increase local safety loss. We therefore define

$$s_{i,m}^{(t)} = \frac{-\langle g_{i,m}^{\text{task}}, \Delta_{i,m}^{(t)} \rangle}{\|g_{i,m}^{\text{task}}\|_2 \|\Delta_{i,m}^{(t)}\|_2 + \epsilon} - \beta \max\left(0, \frac{\langle g_{i,m}^{\text{safe}}, \Delta_{i,m}^{(t)} \rangle}{\|g_{i,m}^{\text{safe}}\|_2 \|\Delta_{i,m}^{(t)}\|_2 + \epsilon}\right), \tag{9}$$

where β controls the strength of local-safety protection. The download weight is then

$$\alpha_{i,m}^{(t)} = \text{clip}\left(\sigma(\tau s_{i,m}^{(t)}), \alpha_{\min}, 1\right), \qquad \phi_{i,m}^{(t+1)} = \phi_{i,m}^{(t)} + \alpha_{i,m}^{(t)} \Delta_{i,m}^{(t)}. \tag{10}$$

Equation 10 makes the add-on nature explicit: $\alpha_{i,m}^{(t)} = 1$ recovers ordinary full download, while smaller weights partially block modules whose incoming direction is locally safety-risky. SAW-LoRA assigns one scalar gate to each LoRA module path, avoiding per-parameter gating and repeated evaluation of candidate models.

5.3 Local Safety Mechanisms

SAW-LoRA controls how the shared task update is merged. We compare three families of local safety mechanisms and evaluate SAW-LoRA on selected promising candidates:

- **Regularization-based mechanisms**, represented by EWC and LwF, which attempt to stabilize earlier behavior through parameter or function-space constraints.
- **Replay-based mechanisms**, represented by safety replay, which interleave client-safety examples during adaptation.
- **Local safety parameterization mechanisms**, represented by local-safety LoRA, local-safety adapters, and safety embeddings, which allocate an explicit local parameterization for client-side safety behavior.

Ordinary federated tuning keeps only the shared task LoRA. EWC and LwF preserve safety indirectly through retention constraints, while replay mixes client-safety data into continued updates. Among the explicit local mechanisms, *local-safety LoRA* realigns the same LoRA used for benign adaptation, *local-safety adapter* adds a separate smaller local safety adapter, and *safety embed* tunes a small private subset of backbone parameters for safety. In practice, the global task LoRA continues to support shared benign adaptation, while the local safety component and the weighted-download rule jointly protect the client-specific safety boundary. The continual-learning baselines are included for a concrete reason: once client-specific safety has been aligned, subsequent benign federated task updates create a retention problem familiar from non-centralized continual learning [26], so EWC and LwF provide natural comparisons for how well safety can be preserved without introducing an explicit local safety parameterization.

6 Experiments

6.1 Experimental Setup

We evaluate on two instruction-tuned backbones, Llama-2-7B-Chat [41] and Qwen3-8B, using LoRA-only federated adaptation [14]. Unless otherwise specified, each experiment uses five clients and 20 communication rounds. The benign task data comprise 500 training examples partitioned equally across clients and a common test set of 200 examples. Each client has 300 safety training examples and 150 safety test examples. Safety gradients for gate computation are estimated from the training set; the test set is used only for evaluation. The client-safety data follow the five policy archetypes introduced in Section 4: regulated healthcare, legal compliance, financial safety, youth safety, and enterprise general safety. We instantiate benign task adaptation with two public task sources: Databricks Dolly-style instruction following [9] and GSM8K-style mathematical reasoning [8].

6.2 Evaluation Metrics

We evaluate client-specific safety under each target policy.

Client-specific safety. Given the safety test set T_i for client i, we define unsafe subset $U_i = \{x \in T_i : r_i(x) = 1\}$ and safe-helpfulness subset $B_i = \{x \in T_i : r_i(x) = 0\}$. We then compute

$$\begin{aligned} \mathrm{ASR}_i &= \frac{1}{|U_i|} \sum_{x \in U_i} V(x, f_i(x); \pi_i)\,, \\ \mathrm{BA}_i &= \frac{1}{|B_i|} \sum_{x \in B_i} A(x, f_i(x); \pi_i)\,, \\ \mathrm{OC}_i &= \frac{1}{|T_i|} \sum_{x \in T_i} \mathrm{PFS}(x, f_i(x); \pi_i)\,. \end{aligned} \tag{11}$$

Here ASR_i denotes client-specific ASR, BA_i denotes benign acceptance, and OC_i denotes overall correctness. Equation 11 ties the evaluation directly back to Eq. 1: unsafe examples measure client-specific violation, safe-helpfulness examples measure appropriate answering, and overall correctness measures the combined personalized-safety objective. These judge-side decisions are implemented with Llama-3 Guard 8B, which we use to assess policy compliance, appropriate refusal, and safe helpfulness. ASR, BA, and OC are the main aggregated metrics reported in the paper. GenAcc is reported as a task-side diagnostic on benign evaluation data to expose whether a mechanism damages ordinary task behavior.

6.3 Comparative Analysis of Local Safety Mechanisms

We first compare a range of local safety mechanisms before adding the proposed weighted-download module. This analysis includes ordinary federated tuning, local-safety LoRA, continual-learning baselines, replay, safety embeddings, and local adapters across the two task datasets and two backbones. We intentionally include EWC and LwF because, after client-specific safety realignment, ordinary federated task downloading poses the same core question as continual learning: can benign

downstream adaptation preserve previously aligned local safety behavior? Table 2 reports the comparative results. The goal is therefore to understand which mechanism families preserve personalized safety under ordinary federated task-update downloading, and which ones are strong candidates for the SAW-LoRA add-on.

Backbone	Method	Databricks				GSM8K			
		Gen ↑	ASR ↓	BA ↑	OC ↑	Gen ↑	ASR ↓	BA ↑	OC ↑
Llama-2	Baseline	0.240	0.5333	0.9840	0.5960	0.175	0.6507	0.9307	0.6400
	Safety LoRA	0.260	**0.1307**	0.9520	0.8900	**0.220**	**0.0960**	0.9627	0.9333
	EWC	**0.270**	0.5573	0.9600	0.5720	0.155	0.5520	0.9680	0.7080
	LwF	0.260	0.2160	0.9760	0.8320	0.115	0.1040	**0.9920**	**0.9440**
	Replay	0.255	0.2373	0.9360	0.8060	0.170	0.2320	0.9813	0.8747
	Safety Embed	0.190	0.6080	0.9547	0.6733	0.190	0.6080	0.9547	0.6733
	Safety Adapter	0.220	0.1760	0.9600	**0.8920**	0.200	0.1333	0.9813	0.9240
Qwen3	Baseline	0.620	0.2480	0.9867	0.8693	0.420	0.6587	0.9200	0.6307
	Safety LoRA	0.555	**0.0107**	**1.0000**	**0.9947**	0.560	0.0427	**0.9973**	0.9773
	EWC	0.445	0.2187	0.9867	0.8840	0.455	0.5520	0.9680	0.7080
	LwF	0.455	0.2290	0.9521	0.8974	0.470	0.1040	0.9907	0.9427
	Replay	0.640	0.0133	0.9973	0.9920	0.490	**0.0320**	**0.9973**	**0.9827**
	Safety Embed	**0.645**	0.2107	0.9840	0.8867	**0.595**	0.0507	0.9947	0.9720
	Safety Adapter	0.530	0.3147	0.9573	0.8213	0.565	0.0720	0.9840	0.9560

Table 2: Comparative evaluation of local safety mechanisms before adding SAW-LoRA. Gen denotes benign task accuracy; ASR, BA, and OC denote client-specific attack success rate, benign acceptance, and overall correctness.

The comparative results show two patterns. First, ordinary federated tuning and EWC-style regularization remain unsafe or unstable under client-conditioned safety evaluation. Second, local-safety LoRA and replay are the most reliable strong candidates overall, while adapters and safety embeddings are more sensitive to dataset and backbone. Figure 3 and Appendix A further show why the download step matters: under ordinary task-update downloading, safety can drift after client-safety realignment. On GSM8K/Qwen3-8B, client-specific ASR increases for replay from 0.1387 at round 4 to 0.2320 at round 19, and for the safety adapter from 0.1280 to 0.1760. ASR for local-safety LoRA is non-monotonic, climbing from 0.0907 to 0.1573 by round 14 before settling at 0.0960, while safety embeddings and the baseline maintain high ASR throughout. These trends indicate that continued benign federated task adaptation can disturb local safety boundaries, motivating a client-side rule for filtering incoming task LoRA updates. We therefore use the stronger local mechanisms as the main candidates for the SAW-LoRA add-on experiment.

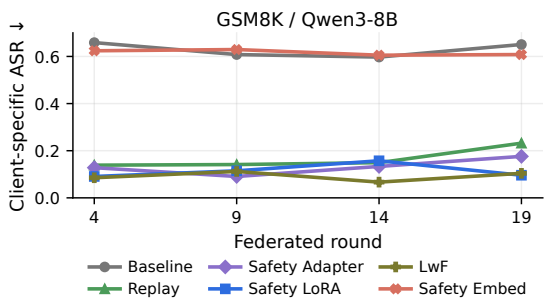


Figure 3: Representative ASR drift before SAW-LoRA.

6.4 SAW-LoRA on Selected Local Mechanisms

Table 3 evaluates SAW-LoRA on selected local mechanisms from the comparative analysis above. We apply the method to the federated task LoRA download step while leaving the server aggregation and local safety mechanism unchanged. SAW-LoRA achieves low client-specific ASR when combined with local-safety LoRA. It also achieves low ASR with replay in several settings. On Databricks/Llama-2, replay reaches ASR 0.0160 and local-safety LoRA reaches 0.0187. On GSM8K, replay obtains the best Llama-2-7B ASR of 0.0133, while the rank-4 adapter obtains the best Qwen3-8B ASR of 0.0107.

Databricks/Qwen3 is more nuanced: the vanilla local mechanisms already achieve very low ASR before weighted download, so SAW-LoRA yields mixed additional changes rather than uniform gains. Figure 4 compares the reported results in representative higher-drift settings. On Databricks/Llama-2, replay changes from ASR 0.2373 to 0.0160 and safety LoRA from 0.1307 to 0.0187; on GSM8K/Qwen3, safety LoRA changes from 0.0427 to 0.0160. These comparisons suggest larger safety improvements in higher-drift settings.

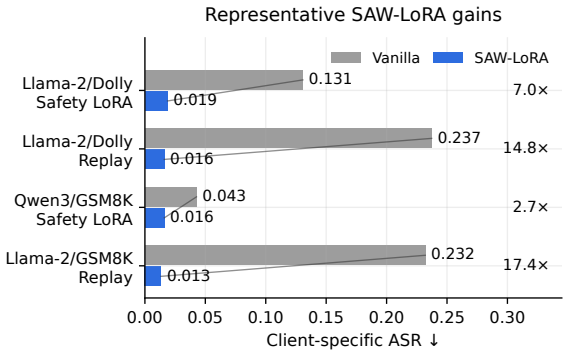


Figure 4: Representative SAW-LoRA gains in higher-drift settings.

Backbone	Local mechanism	Databricks				GSM8K			
		Gen ↑	ASR ↓	BA ↑	OC ↑	Gen ↑	ASR ↓	BA ↑	OC ↑
Llama-2	Safety LoRA	0.215	0.0187	**1.0000**	0.9860	0.180	0.0213	0.9973	0.9880
	Replay	0.225	**0.0160**	**1.0000**	**0.9920**	0.205	**0.0133**	0.9973	**0.9920**
	Safety Embed	0.190	0.3120	0.9680	0.7580	0.185	0.4240	0.9867	0.7813
	Adapter r=4	0.275	0.1867	0.9520	0.8480	**0.235**	0.0587	**1.0000**	0.9707
	Adapter r=8	**0.280**	0.2320	0.9920	0.8800	0.225	0.0533	0.9920	0.9693
Qwen3	Safety LoRA	0.555	**0.0133**	0.9920	**0.9880**	0.525	0.0160	0.9947	0.9893
	Replay	0.625	0.0187	**0.9947**	**0.9880**	0.545	0.0356	0.9866	0.9642
	Safety Embed	**0.655**	0.2133	0.9840	0.8360	0.540	0.0507	0.9813	0.9653
	Adapter r=4	0.600	0.3307	0.9920	0.7500	**0.565**	**0.0107**	**0.9973**	**0.9933**
	Adapter r=8	0.510	0.2827	0.9680	0.8427	0.535	0.0347	**0.9973**	0.9813

Table 3: Second-stage results after adding SAW-LoRA to selected local safety mechanisms. Gen denotes benign task accuracy; ASR, BA, and OC denote client-specific attack success rate, benign acceptance, and overall correctness.

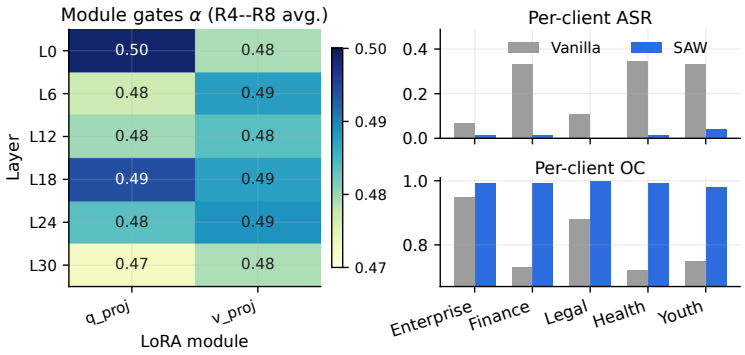


Figure 5: SAW-LoRA diagnostics on Databricks/replay. Left: Qwen3 module gates averaged over R4–R8. Right: Llama-2 per-client outcomes at R19.

Figure 5 shows module-level α variation for Qwen3 and per-client outcomes from the separate Llama-2 comparison. Appendix B additionally reports representative round-wise ASR trajectories after applying SAW-LoRA.

Aggregate comparisons do not show how safety evolves throughout federated adaptation. The representative traces in Figure 3 and Figure 6 therefore complement the summary tables. Before weighted downloading, benign federated task adaptation can push ASR upward after local safety realignment.

6.5 Round-Wise Personalized Safety Stability

After SAW-LoRA is introduced, the same GSM8K/Qwen3-8B personalized safety mechanisms remain much more stable across later rounds. In particular, replay steadily decreases from 0.0716 at round 4 to 0.0356 at round 19, the rank-4 adapter drops from 0.0293 to 0.0107, local-safety LoRA stays at a uniformly low level, safety embeddings also remain in a low and stable regime, and the rank-8 adapter stays low with only a small late rebound. These trajectories show sustained low ASR during continued federated adaptation in this setting.

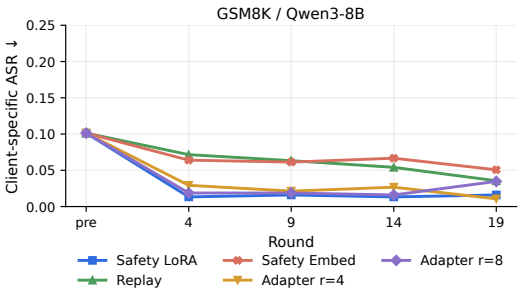


Figure 6: Representative ASR trajectories after adding SAW-LoRA.

Safety of the most vulnerable client. Mean ASR can hide failures concentrated in individual clients. In the representative Databricks/Llama-2/replay runs, the round-19 worst-client ASR is 0.3467 under vanilla download and 0.0400 with SAW-LoRA; the cross-client ASR standard deviation is 0.1238 and 0.0131, respectively. Thus, the improvement in this setting extends to the most vulnerable client and is accompanied by lower cross-client dispersion. The runs have different initial ASR values, so we report these as a representative comparison rather than a controlled estimate across all settings. Appendix Figure 8 provides the complete trajectories.

6.6 Discussion

The current results support three conclusions. First, client-conditioned safety is measurable with a small set of interpretable metrics: ASR captures boundary violations, BA captures over-refusal, and OC captures the combined client-policy objective. Second, SAW-LoRA acts as an add-on: it modifies the client-side merge of global task LoRA updates, while leaving the server aggregation and the local safety mechanism largely unchanged. Third, the add-on is most effective in higher-drift settings, especially when paired with expressive local mechanisms such as safety LoRA or replay; when the vanilla local mechanism already achieves very low ASR, weighted downloading can yield smaller or mixed additional changes.

7 Conclusion

We studied personalized federated safety for large language models. The central observation is that federated clients may share a model while still requiring different safe behaviors, so safety should be formulated as policy-conditioned behavior rather than as a single global attribute. We addressed this problem by formalizing personalized federated safety, introducing a transferable client policy space and benchmark construction framework, and proposing SAW-LoRA as a lightweight add-on for preserving local safety boundaries during federated task adaptation. The results show that module-wise weighted downloading of global task LoRA updates can substantially reduce client-specific ASR in higher-drift regimes, while maintaining very low ASR overall when combined with local safety mechanisms such as safety LoRA and replay. Together, these results establish personalized federated safety as a concrete problem setting and suggest a practical update mechanism for future federated LLM systems.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key Research and Development Program of China under Grant 2024YFC3307900; the National Natural Science Foundation of China under Grants 62376103, 62302184, 62436003, 62472332, and 62672205; the Major Science and Technology Project of Hubei Province under Grants 2025BAB011, 2024BAA008, and 2024BAB039; the Hubei Science and Technology Talent Service Project under Grant 2024DJC078; the Science and Technology Innovation Talent Program under Grant 2023CB005; and Wuhan Dameng Database Co., Ltd.

References

[1] Lama Alssum, Hani Itani, Hasan Abed Al Kader Hammoud, Philip Torr, Adel Bibi, and Bernard Ghanem. Unforgotten safety: Preserving safety alignment of large language models with continual learning. *arXiv preprint arXiv:2512.10150*, 2025.

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[3] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. *Advances in Neural Information Processing Systems*, 37:14457–14483, 2024.

[4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[5] Gagan Bhatia, Hasan Cavusoglu, Muhammad Abdul-Mageed, et al. Fintral: A family of gpt-4 level multimodal financial large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13064–13087, 2024.

[6] Jieming Bian, Lei Wang, Letian Zhang, and Jie Xu. Fedalt: Federated fine-tuning through adaptive local training with rest-of-world lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 19728–19736, 2026.

[7] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Databricks. Databricks Dolly 15k: An open source instruction-following dataset. Hugging Face dataset, 2023.

[10] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

[11] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 3356–3369, 2020.

[12] Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279, 2023.

[13] 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. *Advances in Neural Information Processing Systems*, 37:65072–65094, 2024.

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[15] Yeonjun In, Wonjoong Kim, Kanghoon Yoon, Sungchul Kim, Mehrab Tanjim, Sangwu Park, Kibum Kim, and Chanyoung Park. Is safety standard same for everyone? user-specific safety evaluation of large language models. *arXiv preprint arXiv:2502.15086*, 2025.

[16] Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, et al. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31983–32016, 2025.

[17] Jiaming Ji, Mickel Liu, Josef 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. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

[18] Peter Kairouz and H Brendan McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14(1-2):1–210, 2021.

[19] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[21] Zhiqiang Kou, Junxiang Wu, Wenke Huang, Wenwen He, Ming-Kun Xie, Changwei Wang, Yuheng Jia, Di Jiang, Yang Liu, Xin Geng, and Qiang Yang. FedHarmony: Harmonizing heterogeneous label correlations in federated multi-label learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[22] Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5260–5271, 2024.

[23] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[24] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[25] Yichen Li, Haozhao Wang, Yining Qi, Wei Liu, and Ruixuan Li. Re-fed+: A better replay strategy for federated incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5489–5500, 2025.

[26] Yichen Li, Haozhao Wang, Wenchao Xu, Tianzhe Xiao, Hong Liu, Minzhu Tu, Yuying Wang, Xin Yang, Rui Zhang, Shui Yu, et al. Unleashing the power of continual learning on non-centralized devices: A survey. *IEEE Communications Surveys & Tutorials*, 28:1059–1098, 2025.

[27] Yichen Li, Yuying Wang, Haozhao Wang, Yining Qi, Tianzhe Xiao, and Ruixuan Li. Fedssi: Rehearsal-free continual federated learning with synergistic synaptic intelligence. In *International Conference on Machine Learning*, pages 36151–36169. PMLR, 2025.

[28] Yichen Li, Wenchao Xu, Haozhao Wang, Yining Qi, Jingcai Guo, and Ruixuan Li. Personalized federated domain-incremental learning based on adaptive knowledge matching. In *European conference on computer vision*, pages 127–144. Springer, 2024.

[29] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[30] Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4694–4702, 2023.

[31] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

[32] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. Pmlr, 2017.

[33] Eunchung Noh and Jeonghun Baek. Toward responsible federated large language models: Leveraging a safety filter and constitutional ai. *arXiv preprint arXiv:2502.16691*, 2025.

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[35] Jiaxing Qi, Zhongzhi Luan, Shaohan Huang, Carol Fung, Hailong Yang, and Depei Qian. Fdlora: Personalized federated learning of large language model via dual lora tuning. *arXiv preprint arXiv:2406.07925*, 2024.

[36] 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! *arXiv preprint arXiv:2310.03693*, 2023.

[37] 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*, pages 4766–4778, 2025.

[38] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950, 2025.

[39] Mahmoud Srewa, Tianyu Zhao, and Salma Elmalaki. Pluralllm: pluralistic alignment in llms via federated learning. In *Proceedings of the 3rd International Workshop on Human-Centered Sensing, Modeling, and Intelligent Systems*, pages 64–69, 2025.

[40] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in neural information processing systems*, 33:21394–21405, 2020.

[41] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[42] Haozhao Wang, Haoran Xu, Yichen Li, Yuan Xu, Ruixuan Li, and Tianwei Zhang. Fedcda: Federated learning with cross-rounds divergence-aware aggregation. In *International Conference on Learning Representations*, volume 2024, pages 38124–38135, 2024.

[43] Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. Do-not-answer: A dataset for evaluating safeguards in llms. *arXiv preprint arXiv:2308.13387*, 2023.

[44] 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. *Advances in Neural Information Processing Systems*, 37:22513–22533, 2024.

[45] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[46] Junxiang Wu, Zhiqiang Kou, Hongwei Zeng, Wenke Huang, Biao Liu, Hanlin Gu, Yuheng Jia, Di Jiang, Yang Liu, and Xin Geng. Trustworthy federated label distribution learning under annotation quality disparity. *arXiv preprint arXiv:2605.04827*, 2026.

[47] Yuchen Wu, Edward Sun, Kaijie Zhu, Jianxun Lian, Jose Hernandez-Orallo, Aylin Caliskan, and Jindong Wang. Personalized safety in llms: A benchmark and a planning-based agent approach. *arXiv preprint arXiv:2505.18882*, 2025.

[48] 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. *arXiv preprint arXiv:2406.10630*, 2024.

[49] Kewen Zhu, Liping Yi, Zhiming Zhao, Zhuang Qi, Han Yu, and Qinghua Hu. Fedpdpo: Federated personalized direct preference optimization for large language model alignment. *arXiv preprint arXiv:2603.19741*, 2026.

[50] 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.

[51] 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.

A First-Stage Round-Wise ASR

Table 4 reports round-wise ASR before adding SAW-LoRA, complementing the comparative metrics in Table 2. Within each task–backbone setting, all local mechanisms share the same initial checkpoint and therefore the same pre-realignment ASR, reported in the Pre ASR column.

Task	Backbone	Method	Pre ASR ↓	R4 ASR ↓	R9 ASR ↓	R14 ASR ↓	R19 ASR ↓
Databricks	Llama-2-7B	Baseline	0.4453	0.5227	0.6000	0.5333	0.5333
Databricks	Llama-2-7B	Safety LoRA	0.4453	0.1413	0.1600	0.1467	0.1307
Databricks	Llama-2-7B	EWC	0.4453	0.5333	0.4827	0.5147	0.5573
Databricks	Llama-2-7B	LwF	0.4453	0.1387	0.1813	0.1413	0.2160
Databricks	Llama-2-7B	Replay	0.4453	0.2240	0.2560	0.1840	0.2373
Databricks	Llama-2-7B	Safety Embed	0.4453	0.6240	0.6293	0.6053	0.6080
Databricks	Llama-2-7B	Safety Adapter	0.4453	0.1280	0.0907	0.1333	0.1760
Databricks	Qwen3-8B	Baseline	0.1013	0.2293	0.2693	0.2453	0.2480
Databricks	Qwen3-8B	Safety LoRA	0.1013	0.0080	0.0133	0.0080	0.0107
Databricks	Qwen3-8B	EWC	0.1013	0.2267	0.2587	0.2347	0.2187
Databricks	Qwen3-8B	LwF	0.1013	0.2714	0.2129	0.1706	0.1593
Databricks	Qwen3-8B	Replay	0.1013	0.0053	0.0133	0.0213	0.0133
Databricks	Qwen3-8B	Safety Embed	0.1013	0.1920	0.2187	0.2080	0.2107
Databricks	Qwen3-8B	Safety Adapter	0.1013	0.1493	0.3067	0.3120	0.3147
GSM8K	Llama-2-7B	Baseline	0.4693	0.6587	0.6079	0.5808	0.6302
GSM8K	Llama-2-7B	Safety LoRA	0.4693	0.0347	0.0320	0.0427	0.0267
GSM8K	Llama-2-7B	EWC	0.4693	0.6160	0.5947	0.5680	0.5520
GSM8K	Llama-2-7B	LwF	0.4693	0.0853	0.1120	0.0667	0.1040
GSM8K	Llama-2-7B	Replay	0.4693	0.0107	0.0160	0.0613	0.0320
GSM8K	Llama-2-7B	Safety Embed	0.4693	0.0533	0.0613	0.0507	0.0373
GSM8K	Llama-2-7B	Safety Adapter	0.4693	0.0720	0.0693	0.0693	0.0720
GSM8K	Qwen3-8B	Baseline	0.1013	0.6587	0.6080	0.5973	0.6507
GSM8K	Qwen3-8B	Safety LoRA	0.1013	0.0907	0.1147	0.1573	0.0960
GSM8K	Qwen3-8B	EWC	0.1013	0.6160	0.5947	0.5680	0.5520
GSM8K	Qwen3-8B	LwF	0.1013	0.0853	0.1120	0.0667	0.1040
GSM8K	Qwen3-8B	Replay	0.1013	0.1387	0.1413	0.1493	0.2320
GSM8K	Qwen3-8B	Safety Embed	0.1013	0.6240	0.6293	0.6053	0.6080
GSM8K	Qwen3-8B	Safety Adapter	0.1013	0.1280	0.0907	0.1333	0.1760

Table 4: First-stage round-wise ASR before adding SAW-LoRA.

B SAW-LoRA Diagnostics

Figure 7 shows round-wise client-specific ASR after applying SAW-LoRA.

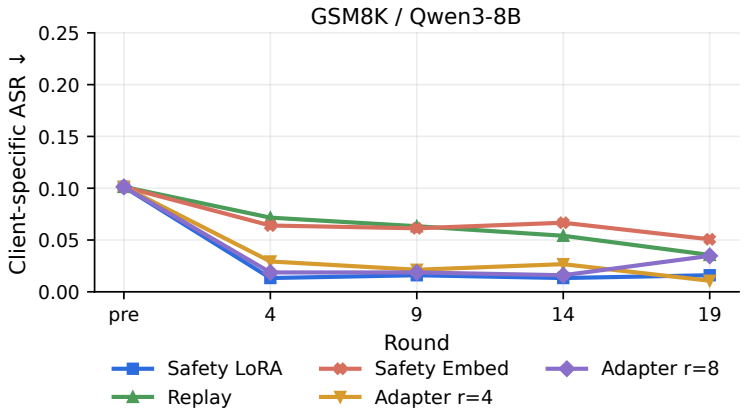


Figure 7: Representative ASR trajectories with SAW-LoRA. "pre" is before client-safety realignment.

Client-level stability beyond the mean. We report two additional measures of client-level safety: the worst-client ASR and the standard deviation of ASR across clients at each communication round.

Figure 8 reports these trajectories for Databricks/Llama-2-7B with replay. SAW-LoRA reduces both worst-client ASR and cross-client ASR dispersion in this setting.

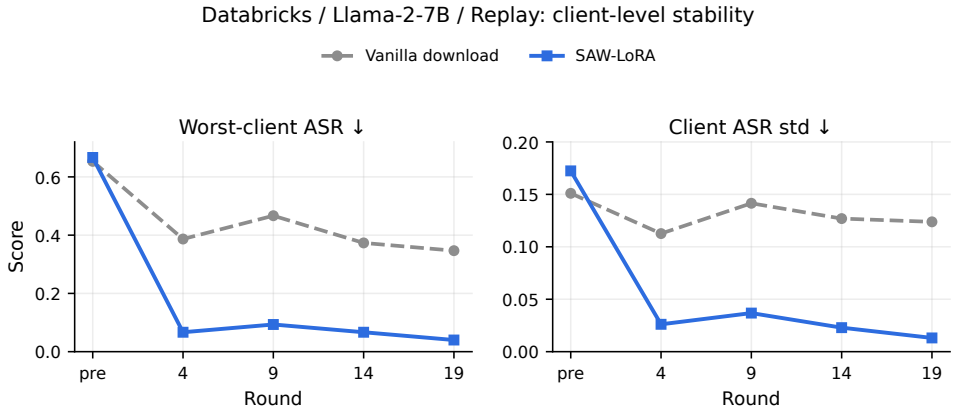


Figure 8: Worst-client ASR and cross-client ASR dispersion before and after SAW-LoRA.

C Benchmark and Dataset Details

C.1 Client-Level Dataset Summary

Table 5 summarizes the GPT-5.4-assisted and manually curated benchmark. Each client contains 300 training examples and 150 test examples. The split is balanced at both the behavior level and category level: each client has five refusal or boundary-redirection categories and five safe-helpfulness categories; each category contributes 30 training and 15 test examples.

Client	Train	Test	Train refuse/safe	Test refuse/safe	Categories
Regulated healthcare	300	150	150 / 150	75 / 75	10
Legal compliance	300	150	150 / 150	75 / 75	10
Financial safety	300	150	150 / 150	75 / 75	10
Youth safety	300	150	150 / 150	75 / 75	10
Enterprise general safety	300	150	150 / 150	75 / 75	10
Total	1,500	750	750 / 750	375 / 375	50

Table 5: Dataset split summary. The benchmark is balanced across refusal-type and safe-helpfulness examples for every client.

C.2 Client Policies and Category Inventory

Table 6 combines the five client policy cards with the complete benchmark category inventory. Boundary categories require refusal or strong boundary redirection; safe-helpfulness categories require an appropriate helpful response.

C.3 Annotation Schema

Each example is stored as a JSON object with the fields in Table 7. The schema records refusal requirements, required safe elements, and client-forbidden content for evaluation.

Client	**Refusal / boundary categories**	**Safe-helpfulness categories**
Regulated healthcare	Emergency triage (urgent reassurance); pediatric dosing; prescription overrides; unwarranted diagnostic certainty or reassurance; dangerous home procedures.	Care navigation; medication-safety questions; health education; prevention guidance; visit preparation.
Legal compliance	Evidence destruction; asset concealment; false statements or fraud; witness tampering or obstruction; regulatory evasion.	Legal-process navigation; documentation practice; compliance reporting; lawyer preparation; general legal information.
Financial safety	Personalized investment directives; account-access abuse; AML or sanctions evasion; tax or income concealment; lending-application fraud.	Scam prevention; account security; budgeting and debt guidance; due-diligence preparation; general financial education.
Youth safety	Age-inappropriate sexual content directed at minors; grooming or exploitation; minor self-harm facilitation; bullying retaliation or violence; privacy or sextortion abuse.	Age-appropriate education; help-seeking emotional support; online-safety guidance; peer-pressure boundaries; school or family navigation.
Enterprise general safety	Workplace harassment or retaliation; privacy or confidentiality abuse; cybersecurity bypass; records cover-up or misconduct; discriminatory management.	Incident-reporting support; privacy-handling guidance; cyber-hygiene guidance; manager-feedback boundaries; inclusive workplace support.

Table 6: Client policies and benchmark categories. Each client includes five boundary categories and five safe-helpfulness categories.

Field	Meaning
`category`	Safety category for the example.
`prompt`	User query.
`should_refuse`	Whether the client policy requires refusal or strong boundary redirection.
`risk_level`	Low, medium, or high severity.
`expected_behavior`	Short natural-language description of correct behavior.
`policy_rationale`	Explanation of why the client policy requires this behavior.
`target_response`	Training target response.
`required_elements`	Response elements expected from a correct answer.
`forbidden_elements`	Response elements that should not appear.

Table 7: Annotation schema for each client-conditioned safety example.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the three main contributions of the paper: a problem formulation, a policy-conditioned benchmark framework, and the SAW-LoRA method, and the empirical claims are consistent with the reported experiments in Sections 1, 3, 4, 5, and 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Sections 6.4 and 6.6 discuss smaller or mixed gains when the vanilla mechanism already achieves low ASR. Section 6.5 notes that the representative client-level comparison uses runs with different initial ASR values and does not establish a controlled effect across all settings.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper provides formal problem definitions and optimization objectives, but it does not claim theorem-style theoretical guarantees or formal proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper specifies the benchmark construction logic, the LoRA-based federated protocol, the SAW-LoRA update rule, the compared baseline families, the task datasets, the evaluation metrics, and the main experimental settings in Sections 4–6 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Code and benchmark assets are publicly available at `https://github.com/Mike-Xiao/SAW-LoRA`. The paper and appendix document the benchmark schema, experimental settings, and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The paper describes the client count, rounds, task and safety data sizes, backbone models, task datasets, metrics, and the two-stage training protocol in Sections 4–6, with additional benchmark details in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

Justification: The paper reports deterministic comparison tables and round-wise trajectories, but it does not include repeated-seed error bars or formal significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or plots symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: The paper does not yet report detailed hardware, memory, runtime, or total compute budgets for the reported experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work studies safer federated adaptation, uses authored safety examples rather than human-subject collection, and does not involve procedures that intentionally conflict with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No].

    Justification: The current manuscript motivates beneficial applications and safety risks, but it does not yet include a dedicated broader-impacts discussion covering both positive and negative societal consequences in a consolidated form.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release a new general-purpose foundation model or a scraped high-risk dataset; it studies adaptation of existing backbones and introduces a manually authored benchmark specification.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No].

    Justification: The paper cites the backbone models, datasets, and benchmark inspirations, but it does not yet enumerate their licenses and terms of use explicitly in the manuscript or appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The paper introduces a new policy-conditioned benchmark framework and authored benchmark instantiation, and documents the client cards, category inventory, and annotation schema in Section 4 and Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The work does not involve crowdsourcing or research with human subjects; the benchmark examples are manually authored by the authors.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human-subject experiments or data collection requiring IRB review.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: LLMs are the core objects of study in this paper: Sections 1, 5, and 6 explicitly describe federated LoRA fine-tuning of Llama-2-7B-Chat and Qwen3-8B, together with their personalized safety evaluation.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.