FEDPRA: POLICY-RESIDUAL AGGREGATION FOR HETEROGENEOUS DIFFERENTIALLY PRIVATE FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) under heterogeneous privacy policies poses significant challenges. Existing private FL methods assign different privacy budgets to clients, yet only exploit these budgets to control numerical privacy loss. They do not measure whether each trained client model satisfies its own operational policy constraints. In this work, we show that privacy-budget order and policy-satisfaction order may diverge. Consequently, conventional data-volume aggregation fails to capture policy satisfaction. To this end, we propose FedPRA to tackle this issue via two complementary designs. First, a differentiable policy engine quantifies each client's residual constraint violation during local model training with record-level differential privacy. Second, policy-residual aggregation leverages these residuals to guide global aggregation. We prove that the proposed weighting rule prioritizes clients with smaller residuals and preserves a set of theoretical guarantees under disjoint and overlapping client data. Experiments based on two benchmark datasets demonstrate the superiority and competitiveness of FedPRA.

1 INTRODUCTION

Federated learning (FL) allows several organizations to train a shared model without collecting raw data at one server (McMahan et al., 2017; Kairouz et al., 2021b). This setting is useful when data sharing is restricted. However, keeping data local does not always make model updates private. This is because gradient inversion can reconstruct training records from shared gradients (Zhu et al., 2019; Geiping et al., 2020); moreover, membership inference can also reveal whether a record contributed to training (Shokri et al., 2017). Nevertheless, differential privacy (DP) limits these risks by clipping individual gradients and adding calibrated noise (Dwork & Roth, 2014; Abadi et al., 2016).

Most private FL methods assign each client a privacy budget ε (McMahan et al., 2018); and recent methods support individualized budgets, privacy-aware client selection, or adaptive resource allocation across rounds (Boenisch et al., 2023; Xu et al., 2023; Kiani et al., 2025). These methods improve how numerical privacy loss is allocated. However, they do not measure whether a trained client model satisfies its own operational policy constraints. Such constraints may describe data minimization, non-retention, and/or limits on representation capacity. Consequently, privacy-budget order need not match policy-satisfaction order. In other words, a privacy budget may control information leakage, but it often does not measure policy satisfaction.

Cross-jurisdiction, which often generates heterogeneous privacy policies, makes the above gap more obvious; and three policy profiles related to GDPR, PIPL, and HIPAA laws, respectively, are already introduced in the FL community (Shafik et al., 2026). These laws do not prescribe numerical DP budgets, yet they provide different operational targets for local policy profiles. Furthermore, conventional FL employs standard aggregation (e.g., FedAvg), which weights each client model according to its local data volume. This rule is simple, but it ignores the residual policy violations of each model. This mismatch between privacy-budget rankings and policy-satisfaction rankings becomes more serious when client data follow different distributions.

To address the above issue, we propose a method dubbed FedPRA, in which two complementary designs are adopted. First, each client trains with record-level differential privacy and evaluates a

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.

differentiable policy profile. The policy engine evaluates normalized tests for the configured policy profile of the client, and summarizes the remaining violations as a constraint residual. Second, policy-residual aggregation (PRA) converts these residuals into server weights, in which a smaller residual is assigned with a larger weight. This design connects local policy reasoning to global optimization, without sharing raw records or latent representations.

In a nutshell, FedPRA keeps policy measurement and formal privacy accounting separate; it uses a residual to measure the deviation from configured targets, where the residual is neither a privacy budget nor a certificate of legal compliance; and it gives record-level DP guarantees for disjoint and overlapping client data. We prove that (1) the residual is faithful to individual constraint satisfaction; (2) PRA preserves residual ordering, bounds each client's influence, and is stable to calibration error and client join/leave events; and (3) these DP guarantees apply to downstream attacks on the released transcript. To summarize, our contributions are:

- To our knowledge, We propose FedPRA, which is the first framework to utilize differentiable local policy engines with PRA for heterogeneous differentially private federated learning.
- We prove its residual fidelity, monotone prioritization, bounded client influence, stability to calibration error, and record-level DP guarantees under the stated privacy assumptions.
- We evaluate our framework on the non-identically distributed network-intrusion benchmarks with three regulation-inspired profiles. The results show that FedPRA preserves strong private utility, improves minority-class performance, reduces data-volume dominance, and maintains a low observed membership leakage, justifying its superiority and competitiveness.

2 Related Work

Our work lies at the intersection of private federated learning and neuro-symbolic federated learning. Existing studies mainly address these areas separately. Private FL focuses on limiting information leakage and managing heterogeneous privacy budgets. Neuro-symbolic FL uses logical knowledge to improve reasoning, personalization, and/or explanation. Next, we review previous works most relevant to ours.

DP-SGD (Abadi et al., 2016) provides record-level privacy by clipping per-example gradients and adding calibrated noise. McMahan et al. (McMahan et al., 2018) adapted this training pattern to federated models. DP-FTRL (Kairouz et al., 2021a) later supported private federated training without random sampling or shuffling. These private FL methods do not cover privacy heterogeneity. Recently, some studies focus on privacy heterogeneity. For example, Robust-HDP (Malekmohammadi et al., 2024) estimates the noise contained in client updates before aggregation. Time-adaptive DP-FL (Kiani et al., 2025) addresses how each client spends its budget across rounds. DPFL-BCS (Ma et al., 2025) discusses client selection under heterogeneous privacy requirements. Xu et al. (Xu et al., 2023) further derive a convergence-guided privacy-aware selection rule. These methods represent heterogeneity through budgets, noise, participation or client reweighting (Cheng et al., 2024). However, they do not measure whether a trained model satisfies client-specific policy targets.

Differentiable reasoning provides another way to encode client requirements. Logic Tensor Networks (Badreddine et al., 2022) translate logical statements into differentiable objectives. FedNSL (Xing et al., 2024) handles heterogeneous rule distributions through federated neuro-symbolic learning. FedSTL (An et al., 2024) attaches client-specific temporal properties to local models and groups clients with compatible reasoning behavior. LR-XFL (Zhang & Yu, 2024) sends local logic rules to the server and uses rule-derived data quality during aggregation. Tian et al. (Tian et al., 2024) instead use server-side gradient alignment for privacy-preserving constrained domain generalization. These studies show that symbolic or constraint information can guide federated optimization; and their main goals are rule learning, personalization, explanation, and/or domain generalization. They, however, ignore the measured policy violations and do not leverage record-level DP accounting with aggregation.

3 PRELIMINARIES AND PROBLEM FORMULATION

Given K federated clients, let $\mathbb{K} = \{1, \ldots, K\}$ be the client index set. Client $k \in \mathbb{K}$ holds $D_k = \{(x_i, y_i)\}_{i=1}^{n_k}$, where x_i is an input feature vector, y_i is its class label, and n_k is the number of records. Client datasets can be disjoint or overlapping, and may follow different distributions. All clients use model h_θ, parameterized by θ.

3.1 FEDERATED LEARNING:

Federated learning trains a shared model without collecting client data at the server (McMahan et al., 2017). FedAvg weights client k in proportion to n_k. Thus, clients with more records receive more influence, regardless of policy satisfaction.

Definition 1. Record-Level Differential Privacy: Datasets D and D' are neighboring if they differ in one record. A randomized mechanism $\mathcal{M}$ is (ε, δ)-differentially private if, for every measurable output set S,

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta. \tag{1}$$

This limits the influence of any training record (Dwork & Roth, 2014). Client k uses budget (ε_k, δ), implemented through DP-SGD by clipping per-example gradients and adding Gaussian noise (Abadi et al., 2016).

Definition 2. Heterogeneous Policy Profile: Client k has profile $\mathcal{Q}_k = (\varepsilon_k, \boldsymbol{\tau}_k)$, where $\boldsymbol{\tau}_k = (\tau_{k,\min}, \tau_{k,\text{ret}}, \tau_{k,\text{des}})$ gives the targets for feature minimization, non-retention, and structural design. These are operational requirements, not legal certification. The corresponding test vector $\mathbf{q}_k(\theta; D_{\text{cal}}) \in [0,1]^3$ is evaluated using model h_θ on a fixed public calibration set D_{cal}, disjoint from all private training sets. Profiles are heterogeneous when $\mathcal{Q}_i \neq \mathcal{Q}_j$ for some clients $i \neq j$.

3.2 PROBLEM STATEMENT:

At round t, the server broadcasts global parameters θ^t. Each client k in the participating set $\mathcal{S}_t$ privately trains local parameters θ_k^{t+1} and evaluates its policy tests on D_{cal} to obtain residual Δ_k^t. Here, $\Delta_k^t \in [0,1]$ denotes the average directional violation of Client k's configured policy targets. It releases only $(\theta_k^{t+1}, \Delta_k^t)$; individual test values are not transmitted. The server computes $\theta^{t+1} = \sum_{k \in \mathcal{S}_t} \alpha_k^t \theta_k^{t+1}$, where $\alpha_k^t \geq 0$ and $\sum_{k \in \mathcal{S}_t} \alpha_k^t = 1$. Our goal is to train a global model under heterogeneous client policies while preserving predictive utility and the assigned record-level DP guarantees. The aggregation rule should give no less influence to a client with a smaller measured policy violation; that is, for $i, j \in \mathcal{S}_t$, $\Delta_i^t \leq \Delta_j^t \Rightarrow \alpha_i^t \geq \alpha_j^t$.

3.3 THREAT MODEL:

We assume an honest-but-curious server and honest clients. The server observes all releases, while an external adversary may query the global model. Each client's accountant covers all local steps and releases across rounds, providing its assigned (ε_k, δ) record-level DP guarantee. The calibration set and policy targets are public, so the residual is DP post-processing.

4 THE PROPOSED METHOD

FedPRA follows three stages as shown in Figure 1. First, each client performs policy-aware private training using three modules: (i) Feature minimization suppresses unnecessary latent dimensions; (ii) Risk-aware non-retention perturbs high-risk dimensions; and (iii) Structural privacy design limits representation magnitude. DP-SGD optimizes the combined objective under the client-specific record-level privacy budget.

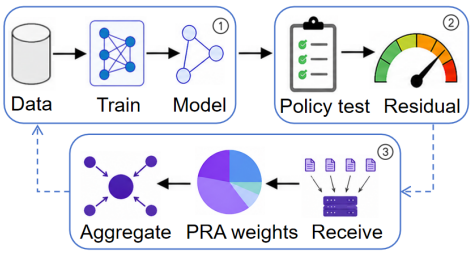

Figure 1: FedPRA training pipeline.

Second, each client computes a normalized residual that measures its remaining policy targets.

Third, the server uses the residuals to compute PRA weights, aggregates the local models, and broadcasts the updated global model. We next cover each stage in detail.

4.1 Local Policy-Aware Private Training

The local stage adapts the shared model to each client's data. Three policy losses guide training toward the client's targets, while DP-SGD provides the assigned record-level privacy guarantee. At the start of round t, client k receives the global parameters and trains under its policy profile. Then, it initializes local parameters $\theta_k = (\omega_k, \phi_k, \psi_k)$ from the global parameters. These operations parameterize the encoder E_{ω_k}, policy network g_{ϕ_k}, and classifier C_{ψ_k}. In the following, when the context is clear, we omit the indices i and k from intermediate quantities such as z, unless otherwise stated. The encoder first produces $z = E_{\omega_k}(x_i) \in \mathbb{R}^h$, where h is the latent dimension. The three policy modules operate on z according to the targets in $\mathcal{Q}_k$.

4.1.1 Feature minimization:

The first module learns a soft mask that suppresses unnecessary latent dimensions:

$$\mathbf{m} = \sigma(g_{\phi_k}(z)) \in [0,1]^h, \qquad z' = z \odot \mathbf{m}, \tag{2}$$

where σ is sigmoid and $\odot$ is element-wise multiplication. A mask entry near zero suppresses its latent dimension. The feature-suppression score and loss are:

$$q^{\min} = \frac{1}{h}\|\mathbf{1} - \mathbf{m}\|_1, \qquad \mathcal{L}_{\min,k}(i) = \left(q^{\min} - \tau_{k,\min}\right)^2. \tag{3}$$

Thus, $q^{\min} \in [0,1]$, with larger values indicating greater suppression. The loss approaches the target from either direction, whereas the residual below records only a shortfall.

4.1.2 Risk-aware non-retention:

The second module estimates the retained dimensions' effects on prediction. For the fixed record, define the risk vector as $\boldsymbol{s} = |\nabla_{z'}\ell(C_{\psi_k}(z'), y_i)|^{\odot 2}$, where the square is applied element-wise. This per-record computation avoids coupling one record's unclipped gradient to other mini-batch records. Normalize it as $\bar{\boldsymbol{s}} = \boldsymbol{s}/(\|\boldsymbol{s}\|_\infty + \xi)$, where $\xi > 0$ prevents division by zero. The retained representation is then perturbed as $\widehat{z} \sim \mathcal{N}(z', \sigma^2_{\mathrm{rep},k}\,\mathrm{diag}(\bar{\boldsymbol{s}}))$ without introducing a separate noise variable.

This implies that higher-risk dimensions receive stronger perturbation; it is a representation regularizer, not the formal DP mechanism. We measure the remaining risk-weighted representation energy as:

$$q^{\mathrm{ret}} = \frac{\|\bar{\boldsymbol{s}}^{1/2} \odot z'\|_2^2}{\|z'\|_2^2 + \xi}, \qquad \mathcal{L}_{\mathrm{ret},k}(i) = \left[q^{\mathrm{ret}} - \tau_{k,\mathrm{ret}}\right]_+^2, \tag{4}$$

where $[u]_+ = \max(0,u)$; the score q^{ret} lies in $[0,1]$; and the one-sided loss penalizes only values above the configured limit.

4.1.3 Structural privacy design:

After the first two modules control feature use and retention, the third module constrains the magnitude of the resulting representation. For a fixed scale $\kappa > 0$, we define:

$$q^{\mathrm{des}} = \frac{\|z\|_2^2}{\|z\|_2^2 + \kappa}, \qquad \mathcal{L}_{\mathrm{des},k}(i) = \left[q^{\mathrm{des}} - \tau_{k,\mathrm{des}}\right]_+^2. \tag{5}$$

The normalized score ranges from 0 to 1, and the loss encourages it to match the client's target. We then combine the classification loss $\mathcal{L}_{\mathrm{cls},k}(i) = \ell(C_{\psi_k}(\widehat{z}), y_i)$ with the three policy losses:

$$\mathcal{L}_k(i) = \mathcal{L}_{\mathrm{cls},k}(i) + \lambda_{\min,k}\mathcal{L}_{\min,k}(i) + \lambda_{\mathrm{ret},k}\mathcal{L}_{\mathrm{ret},k}(i) + \lambda_{\mathrm{des},k}\mathcal{L}_{\mathrm{des},k}(i). \tag{6}$$

The coefficients control the policy losses, and the targets specify the client's requirements.

4.1.4 PRIVATE OPTIMIZATION:

To optimize this objective privately, client k applies DP-SGD. It first clips each per-record gradient to the bound G_k:

$$\bar{g}_{k,i} = \frac{\nabla_{\theta_k}\mathcal{L}_k(i)}{\max(1, \|\nabla_{\theta_k}\mathcal{L}_k(i)\|_2/G_k)}. \tag{7}$$

For each local minibatch B_k, the client averages the clipped gradients and adds Gaussian noise:

$$\widetilde{g}_k = \frac{1}{|B_k|}\left(\sum_{i\in B_k}\bar{g}_{k,i} + \mathcal{N}\big(0, \sigma^2_{\mathrm{DP},k}G_k^2 I\big)\right). \tag{8}$$

Here, $\sigma_{\mathrm{DP},k}$ is the Gaussian noise multiplier, and I is the identity matrix with the same dimension as the gradient.

This private update is repeated across all local steps and rounds. The privacy accountant uses a worst-case upper bound on each record's inclusion probability. The complete released sequence satisfies (ε_k, δ)-DP. Only DP-SGD provides this formal guarantee. The representation perturbation defined above is not counted as privacy amplification.

4.2 NORMALIZED POLICY RESIDUAL

The normalized residual measures how far a local model is from satisfying its three policy targets. After local training, client k evaluates $\mathbf{q}_k(\theta_k^{t+1}; D_{\mathrm{cal}})$. Its calibration-set averages are $q_k^{\min,t}$, $q_k^{\mathrm{ret},t}$, and $q_k^{\mathrm{des},t}$, each in $[0,1]$. Feature minimization requires a minimum score, whereas non-retention and structural design impose upper limits. Therefore, the normalized residual is:

$$\Delta_k^t = \frac{1}{3}\Big([\tau_{k,\min} - q_k^{\min,t}]_+ + [q_k^{\mathrm{ret},t} - \tau_{k,\mathrm{ret}}]_+ + [q_k^{\mathrm{des},t} - \tau_{k,\mathrm{des}}]_+\Big). \tag{9}$$

The policy-loss coefficients scale local optimization and are not aggregation weights.

Proposition 1 (Zero-residual characterization)**.** *The residual Δ_k^t is zero if and only if all three policy targets are satisfied on D_{cal}.*

Proof. Each term in Eq. equation 9 is nonnegative. Their average is zero if and only if every term is zero, which is equivalent to satisfying all three policy targets. □

A zero residual indicates satisfaction on D_{cal}, not legal certification.

4.2.1 CALIBRATION DISTRIBUTION:

The residual measures policy satisfaction on the public D_{cal}. If this set differs from the clients' distributions, the residual may not reflect private-data behavior. We therefore use class-stratified calibration from the same sources and analyze the estimation error below.

Proposition 2 (Transcript privacy)**.** *Suppose client k's complete released parameter sequence is (ε_k, δ)-DP. Releasing its residuals preserves the same guarantee. For disjoint client datasets, the complete transcript is $(\max_{k\in\mathbb{K}}\varepsilon_k, \delta)$-DP. If a record x appears at clients $\mathcal{C}(x)$, the transcript is $\Big(\sum_{k\in\mathcal{C}(x)}\varepsilon_k, |\mathcal{C}(x)|\delta\Big)$-DP.*

Proof. Each residual is computed from a released DP model and public quantities. Post-processing therefore preserves the client's guarantee. Parallel composition gives the disjoint-data bound, whereas sequential composition gives the overlapping-data bound. □

This result assumes public or data-independent participation, with each privacy accountant covering all participating rounds.

Corollary 1 (Post-processing guarantee)**.** *Any randomized function of the released transcript satisfies the same DP guarantee established in Proposition 2. This includes membership inference, attribute inference, and record reconstruction attacks.*

Proof. This follows directly from differential privacy's closure under post-processing. □

Record-level DP bounds the influence of individual records. It does not protect model functionality or confidential policy metadata. Model extraction is therefore outside our scope. If policy targets are sensitive, the residual requires a separate privacy mechanism.

4.3 Policy-Residual Aggregation

After local training, the server receives each participating client's local parameters and normalized residual. Unlike FedAvg, PRA gives greater influence to clients with smaller policy violations. For $\mathcal{S}_t$, it clips each residual to $[0, 1]$ and computes

$$\alpha_k^t = \frac{(1+\Delta_k^t)^{-1}}{\sum_{j\in\mathcal{S}_t}(1+\Delta_j^t)^{-1}}, \qquad \theta^{t+1} = \sum_{k\in\mathcal{S}_t} \alpha_k^t \theta_k^{t+1}. \tag{10}$$

Clipping does not alter the residuals from Eq. equation 9, which already lie in $[0, 1]$. The added constant prevents division by zero and keeps all weights positive. Thus, the global parameters remain a convex combination of the local parameters, while policy targets, loss coefficients, and privacy budgets remain client-specific.

Proposition 3 (PRA prioritization)**.** *Let $K_t = |\mathcal{S}_t|$. For participating clients i and j,*

$$\Delta_i^t \le \Delta_j^t \Rightarrow \alpha_i^t \ge \alpha_j^t, \qquad \frac{1}{2K_t - 1} \le \alpha_k^t \le \frac{2}{K_t + 1}, \qquad \frac{1}{2} \le \frac{\alpha_i^t}{\alpha_j^t} \le 2. \tag{11}$$

Proof. Since $\Delta_k^t \in [0, 1]$, each unnormalized weight $(1 + \Delta_k^t)^{-1}$ lies in $[1/2, 1]$ and decreases with the residual. Therefore, a smaller residual receives no smaller weight. Normalizing these values gives the weight and ratio bounds in Eq. equation 11. □

PRA therefore prioritizes lower-violation clients without selecting or removing participants. With three clients, every aggregation weight remains in $[0.2, 0.5]$.

5 Experiments

5.1 Experimental Setup

Datasets. CIC-IDS2017 contains 2,830,743 records and 78 features (Sharafaldin et al., 2018). We use (2,264,594; 141,537; 424,612) records for private training, public calibration, and test, respectively. Positive and negative infinities are treated as missing values, and all 2,867 affected rows are median-imputed. Imputation and scaling parameters are fitted on private training data and then applied to calibration and test data; no record is removed. Another dataset is UNSW-NB15 (Moustafa & Slay, 2015), and we use (157,807; 17,534; 82,332) records for private training, public calibration, and test, respectively. Both datasets use three non-identically distributed clients with Dirichlet concentration 0.5.

Baselines. For fairness, we include baselines that satisfy three comparability conditions: (1) the method addresses supervised horizontal FL; (2) its server rule can be easily reproduced; and (3) it can use the same architecture, local optimizer, record-level budgets, and final-checkpoint evaluation.

Client	Profile	ε_k	Targets τ_k	Loss coefficients λ_k	Achieved tests $\mathbf{q}_k(\theta^T)$	Residual
A	GDPR-inspired	1	(0.10, 0.15, 0.20)	(1.0, 0.5, 0.5)	(0.000, 0.216, 0.195)	0.0553
B	PIPL-inspired	2	(0.15, 0.10, 0.15)	(0.5, 1.0, 0.5)	(0.027, 0.150, 0.143)	0.0577
C	HIPAA-inspired	3	(0.20, 0.05, 0.10)	(0.5, 0.5, 1.0)	(0.108, 0.057, 0.035)	0.0330

Table 1: Final-round local-model tests on CIC-IDS2017. Tuples follow the format of (minimization, retention, design). The residuals in this table are client-local quantities that PRA uses to compute aggregation weights.

Category	Method	Mean BA (%) ↑	Macro-F1 ↑	Worst BA (%) ↑	MIA AUC → 0.5	Time (h) ↓
Non-private	FedAvg	98.45 ± 0.15	0.7560 ± 0.002	96.05 ± 0.50	0.5013 ± 0.002	3.28 ± 0.05
Private FL	DP-FedAvg	82.07 ± 1.20	0.7314 ± 0.008	80.12 ± 1.50	0.5049 ± 0.005	$\underline{5.10 \pm 0.10}$
	Hetero-budget DP-FedAvg	86.92 ± 0.90	0.7674 ± 0.006	84.86 ± 1.00	0.5150 ± 0.005	5.25 ± 0.10
	Time-Adaptive DP-FL	89.71 ± 0.70	0.7881 ± 0.005	87.58 ± 0.80	0.5451 ± 0.008	5.40 ± 0.10
	Privacy-Aware Selection	89.24 ± 0.70	0.7846 ± 0.005	87.12 ± 0.80	0.5250 ± 0.006	$\mathbf{4.85 \pm 0.10}$
Constrained FL	PMG + DP-SGD	84.90 ± 1.00	0.7524 ± 0.007	82.89 ± 1.20	0.4998 ± 0.005	6.20 ± 0.15
	Noise Cancellation	$\underline{90.78 \pm 0.50}$	$\underline{0.7960 \pm 0.004}$	$\underline{88.63 \pm 0.60}$	0.4881 ± 0.006	6.56 ± 0.15
Ours	**FedPRA**	$\mathbf{94.46 \pm 0.50}$	$\mathbf{0.8341 \pm 0.004}$	$\mathbf{92.28 \pm 0.60}$	0.5031 ± 0.005	5.61 ± 0.12

Table 2: Comparison results on CIC-IDS2017. All private methods use client budgets $(1, 2, 3)$ and satisfy $(3, 10^{-5})$-DP for disjoint data. AUC $= 0.5$ denotes random ranking. Values are mean $\pm$ std over 10 runs.

Note that, for every adaptation we retain the original server-side rule and use the common record-level DP-SGD optimizer; and privacy accounting is recomputed from the resulting local sampling schedule.

Specifically, we use the following six baselines: DP-FedAvg (McMahan et al., 2018), which uses a shared noise multiplier. Heterogeneous-budget DP-FedAvg (Malekmohammadi et al., 2024), which uses client-specific noise multipliers. Besides, we also use Time-Adaptive DP-FL (Kiani et al., 2025), Privacy-Aware Selection (Xu et al., 2023), PMG with DP-SGD (Tian et al., 2024), and Noise Cancellation (Reshef & Levy, 2025). Each private method uses total client budgets $(1, 2, 3)$. To understand the upper utility, we also use non-private FedAvg (McMahan et al., 2017) for reference.

Evaluation metrics. We use the following evaluation metrics: (1) MeanBA, the final global model's balanced accuracy averaged across clients; (2) WorstBA, the minimum client value; (3) mean macro-F1, defined analogously; (4) membership-inference attack AUC, an attack-specific diagnostic rather than a privacy certificate; and (5) end-to-end runtime on the same hardware. In the matched aggregation ablation, mean residual denotes the final global model's policy residual averaged across client profiles. This differs from the client-local residuals in Table 1, which PRA uses to compute aggregation weights. We report this metric only in the matched ablation because the external baselines do not define the same policy tests.

Implementation details. We use Flower for federated learning, and Opacus (Yousefpour et al., 2021) for DP-SGD. All experiments are implemented on NVIDIA RTX 4090 GPU using PyTorch. The shared model has latent dimension $h = 64$. We train for ten global rounds using Adam with learning rate 10^{-3}, batch size 256, and 5 local epochs per round. Client clipping norms are $G_k = 1.0$ for all clients. Noise multipliers $\sigma_{\mathrm{DP},k}$ are set via the RDP accountant to achieve $(\varepsilon_k, 10^{-5})$-DP, and the orders range from 2 to 256. We set $\sigma_{\mathrm{rep},k} = 0.1$, $\kappa = 1.0$, $\xi = 10^{-8}$, and class weights use private training-class frequencies. Ten independent runs use seeds 0∼9; each run resets model initialization, mini-batch sampling, and DP noise. Every reported entry is computed from the ten final-round checkpoints; and federation rounds are not treated as independent replicates. Table 1 summarizes the configured client profiles and the final policy-tests.

5.2 Comparing with State-Of-The-Art Methods

Table 2 shows the results of comparing with the baselines. We can see that Noise Cancellation is the strongest private baseline in three metrics (mean BA, mean F1, and worst BA). Our method FedPRA improves MeanBA and WorstBA by 3.68 and 3.65 percentage points, respectively, and macro-F1 by 0.0381. This justifies that FedPRA has competitive performance among these private methods.

Aggregation	Mean BA (%) ↑	Macro-F1 ↑	Worst BA (%) ↑	Mean Residual ↓
Volume-weighted	91.45 ± 0.70	0.812 ± 0.006	88.12 ± 1.20	0.045 ± 0.005
Uniform $(1/K)$	$\mathbf{94.65 \pm 0.55}$	0.828 ± 0.005	91.88 ± 1.00	0.042 ± 0.004
Shuffled-residual	94.12 ± 0.60	0.826 ± 0.005	91.45 ± 1.10	0.043 ± 0.004
PRA (FedPRA)	94.46 ± 0.50	$\mathbf{0.8341 \pm 0.004}$	$\mathbf{92.28 \pm 0.60}$	$\mathbf{0.037 \pm 0.003}$

Table 3: Matched aggregation ablation. Mean residual is evaluated on the final global model across the three client profiles. Only the server rule changes. Values are mean ± std over 10 paired runs.

Category	Method	Mean BA (%) ↑	Macro-F1 ↑	Worst BA (%) ↑	MIA AUC → 0.5	Time (h) ↓
Non-private	FedAvg	92.34 ± 0.20	0.8923 ± 0.002	89.87 ± 0.40	0.5008 ± 0.002	3.45 ± 0.05
Private FL	DP-FedAvg	75.12 ± 1.10	0.7001 ± 0.008	72.03 ± 1.30	0.5050 ± 0.005	4.82 ± 0.10
	Hetero-budget DP-FedAvg	78.91 ± 0.85	0.7432 ± 0.006	75.67 ± 1.00	0.5148 ± 0.005	4.96 ± 0.10
	Time-Adaptive DP-FL	82.43 ± 0.65	0.7705 ± 0.005	79.21 ± 0.80	0.5402 ± 0.008	5.12 ± 0.10
	Privacy-Aware Selection	81.97 ± 0.65	0.7670 ± 0.005	78.65 ± 0.80	0.5203 ± 0.006	$\mathbf{4.58 \pm 0.10}$
Constrained FL	PMG + DP-SGD	77.55 ± 0.95	0.7289 ± 0.007	74.23 ± 1.10	0.5012 ± 0.005	5.95 ± 0.15
	Noise Cancellation	84.67 ± 0.45	0.7833 ± 0.004	81.94 ± 0.55	0.4856 ± 0.006	6.32 ± 0.15
Ours	**FedPRA**	$\mathbf{88.12 \pm 0.45}$	$\mathbf{0.8245 \pm 0.004}$	$\mathbf{85.76 \pm 0.55}$	0.5035 ± 0.005	5.38 ± 0.12

Table 4: Comparison on UNSW-NB15. All private methods use client budgets $(1, 2, 3)$ and satisfy $(3, 10^{-5})$-DP for disjoint data. Values are mean ± standard deviation over 10 runs.

We note that, although the non-private method FedAvg reaches the highest meanBA value of 98.45%, the meanBA value of our method is close to it. In addition, our method FedPRA has a higher macro-F1 (0.8341 versus 0.7560), indicating a different class-wise error profile. As for MIA AUCs, they vary across private baselines, but the attack is weak in this setting: even for the non-private method FedAvg, it is nearly random. We can also see that FedPRA requires 5.61 hours; it consumes less time than the strongest baseline Noise Cancellation (6.56 hours), demonstrating its superiority from another perspective.

5.3 Ablation Study

Effect of Class Re-weighting and Re-sampling. Figure 2(a) compares unweighted training, class reweighting, and reweighting with resampling for Client C. Reweighting raises MeanBA from 46.68% to 87.61%. Adding resampling raises Client C's balanced accuracy from 68.48% to 95.14% and MeanBA to 94.46%. Thus, class weighting provides the main gain, while resampling addresses Client C's rare-class imbalance.

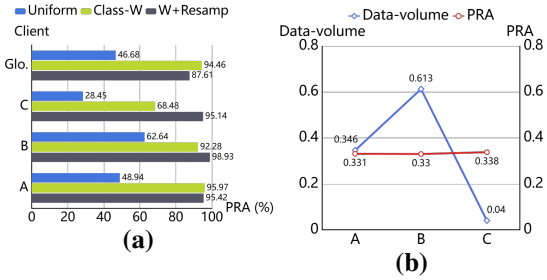

Figure 2: Ablation analysis: (a) class-imbalance treatment; (b) PRA weight redistribution.

Effect of Policy-Residual Aggregation. Figure 2(b) shows that PRA raises Client C's aggregation weight from 0.040 to 0.338 because it has the smallest policy residual. In the matched ablation, which changes only the server rule, PRA improves MeanBA and WorstBA by 3.01 and 4.16 percentage points, respectively, and macro-F1 by 0.0221 over volume weighting (Table 3).

Compared with uniform aggregation, PRA slightly lowers MeanBA (94.46% versus 94.65%) but improves WorstBA by 0.40% and macro-F1 by 0.0061. Shuffling the residual–client assignments weakens performance, confirming the importance of correct residual–client correspondence. Overall, PRA improves worst-client performance and reduces data-volume dominance rather than maximizing MeanBA alone.

5.4 Membership Inference Attack (MIA)

We implement a shadow-model attack with a logistic-regression classifier and three-fold cross-validation (Shokri et al., 2017). As shown in Table 2, FedPRA and non-private FedAvg obtain MIA AUCs of 0.5031 and 0.5013, respectively, both close to random ranking at 0.5. Therefore, this attack is uninformative in our setting, and is not suitable for demonstrating a privacy advantage of FedPRA. Nevertheless, Corollary 1 and the DP accountant provide the formal record-level guarantee.

5.5 Convergence Analysis

To evaluate whether FedPRA maintains high client-level utility throughout training, Table 5 reports the balanced accuracy of the global model on each client test set across ten communication rounds. Clients A and C remain above 95% in every round. Client B remains near 99% during most rounds but decreases to 92.52% and 92.28% in rounds 8 and 10, respectively, with greater variability in both rounds.

Round	Client A	Client B	Client C
1	0.9517 ± 0.0061	0.9885 ± 0.0042	0.9709 ± 0.0073
2	0.9733 ± 0.0048	0.9913 ± 0.0038	0.9733 ± 0.0062
3	0.9686 ± 0.0055	0.9887 ± 0.0045	0.9734 ± 0.0071
4	0.9667 ± 0.0052	0.9882 ± 0.0049	0.9758 ± 0.0064
5	0.9731 ± 0.0042	0.9918 ± 0.0033	0.9777 ± 0.0058
6	0.9651 ± 0.0067	0.9922 ± 0.0031	0.9775 ± 0.0069
7	0.9651 ± 0.0059	0.9923 ± 0.0036	0.9864 ± 0.0052
8	0.9692 ± 0.0058	0.9252 ± 0.0241	0.9810 ± 0.0060
9	0.9682 ± 0.0063	0.9926 ± 0.0040	0.9847 ± 0.0055
10	0.9597 ± 0.0068	0.9228 ± 0.0317	0.9514 ± 0.0123

Table 5: Global-model balanced accuracy by round on each client test set (mean $\pm$ standard deviation over ten runs).

Because the trajectories are nonmonotonic, we use the fixed final checkpoint in all comparisons. Overall, FedPRA maintains high client-level utility throughout all training rounds, although Client B exhibits some late-round variability.

5.6 Results on UNSW-NB15

To validate the generality of FedPRA, we evaluate it on another dataset, UNSW-NB15, a ten-class intrusion-detection dataset. Table 4 reports the results. Among these private methods, FedPRA achieves the highest MeanBA (88.12%), macro-F1 (0.8245), and WorstBA (85.76%). In contrast, Noise Cancellation is the strongest private baseline, obtaining 84.67%, 0.7833, and 81.94%, respectively. Simply speaking, FedPRA improves MeanBA and WorstBA by 3.45 and 3.82 percentage points, respectively, and macro-F1 by 0.0412. In addition, similar to the results on CIC-IDS2017 dataset, Non-private FedAvg achieves highest (92.34%) MeanBA. Nevertheless, our result is closest to this reference point, compared against other private methods. Overall, these results demonstrate that FedPRA remains effective on another benchmark dataset. (Additional experimental results, theoretical analyses, and proofs are provided in the supplementary material.)

6 Conclusion

In this paper, We showed that privacy-budget order and policy-satisfaction order may diverge, making conventional data-volume aggregation unsuitable for representing policy satisfaction. To address this mismatch, we proposed FedPRA, which combines a differentiable policy engine with policy-residual aggregation while maintaining separate record-level DP accounting. We established residual fidelity, monotone prioritization, bounded client influence, stability to calibration error and participation changes, and record-level DP guarantees under explicit assumptions. Experiments on the non-IID CIC-IDS2017 and UNSW-NB15 datasets with three regulation-inspired profiles showed that FedPRA consistently outperforms the private baselines in MeanBA, macro-F1, and WorstBA. The matched ablation further demonstrated that PRA improves worst-client performance and reduces data-volume dominance. Overall, FedPRA provides a principled approach to incorporating heterogeneous operational policy requirements into differentially private federated learning.

REFERENCES

Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016. doi: 10.1145/2976749.2978318.

Ziyan An, Taylor T. Johnson, and Meiyi Ma. Formal logic enabled personalized federated learning through property inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10882–10890, 2024. doi: 10.1609/aaai.v38i10.28962.

Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022. doi: 10.1016/j.artint.2021.103649.

Franziska Boenisch, Christopher Mühl, Roy Rinberg, Jannis Ihrig, and Adam Dziedzic. Individualized PATE: Differentially private machine learning with individual privacy guarantees. *Proceedings on Privacy Enhancing Technologies*, 2023(1):158–176, 2023. doi: 10.56553/popets-2023-0010. URL https://petsymposium.org/popets/2023/popets-2023-0010.php.

Shu-Ling Cheng, Chin-Yuan Yeh, Ting-An Chen, Eliana Pastor, and Ming-Syan Chen. FedGCR: Achieving performance and fairness for federated learning with distinct client types via group customization and reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11498–11506, 2024.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042.

Jonas Geiping, Hartmut Bauermeister, Hannah Dr"oge, and Michael Moeller. Inverting gradients: How easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, volume 33, pp. 16937–16947, 2020.

Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5213–5225. PMLR, 2021a. URL https://proceedings.mlr.press/v139/kairouz21b.html.

Peter Kairouz et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021b. doi: 10.1561/2200000083.

Shahrzad Kiani, Nupur Kulkarni, Adam Dziedzic, Stark C. Draper, and Franziska Boenisch. Differentially private federated learning with time-adaptive privacy spending. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=W0nydevOlG.

Jiating Ma, Yipeng Zhou, Qi Li, Quan Z. Sheng, Laizhong Cui, and Jiangchuan Liu. The power of bias: Optimizing client selection in federated learning with heterogeneous differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 22(5):5672–5687, 2025. doi: 10.1109/TDSC.2025.3572527.

Saber Malekmohammadi, Yaoliang Yu, and Yang Cao. Noise-aware algorithm for heterogeneous differentially private federated learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34461–34498. PMLR, 2024. URL https://proceedings.mlr.press/v235/malekmohammadi24a.html.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Ag"uera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ0hF1Z0b.

Nour Moustafa and Jill Slay. UNSW-NB15: A comprehensive data set for network intrusion detection systems network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6. IEEE, 2015.

Roie Reshef and Kfir Y. Levy. Privacy-preserving federated convex optimization: Balancing partial-participation and efficiency via noise cancellation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51503–51530. PMLR, 2025. URL https://proceedings.mlr.press/v267/reshef25a.html.

Wasswa Shafik, Pushan Kumar Dutta, and Priyadarshini Pattanaik (eds.). *The Convergence of Federated Learning and Healthcare 5.0 and Beyond: A New Era of Intelligent Health Systems*, volume 1247 of *Studies in Computational Intelligence*. Springer Nature Switzerland, Cham, 2026. doi: 10.1007/978-3-032-03985-9.

Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 108–116. SciTePress, 2018. doi: 10.5220/0006639801080116.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy*, pp. 3–18. IEEE, 2017. doi: 10.1109/SP.2017.41.

Chris Xing Tian, Haoliang Li, Yufei Wang, and Shiqi Wang. Privacy-preserving constrained domain generalization via gradient alignment. *IEEE Transactions on Knowledge and Data Engineering*, 36(5):2142–2150, 2024. doi: 10.1109/TKDE.2023.3315279.

Pengwei Xing, Songtao Lu, and Han Yu. Federated neuro-symbolic learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54635–54655. PMLR, 2024. URL https://proceedings.mlr.press/v235/xing24a.html.

Ruichen Xu, Ying-Jun Angela Zhang, and Jianwei Huang. Tackling privacy heterogeneity in federated learning. In *Proceedings of the 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pp. 326–333. IEEE, 2023.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. In *Privacy in Machine Learning Workshop at the Conference on Neural Information Processing Systems*, 2021.

Yanci Zhang and Han Yu. LR-XFL: Logical reasoning-based explainable federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21788–21796, 2024. doi: 10.1609/aaai.v38i19.30179.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html.

A NOTATIONS

Symbol	Definition
$K,\ \mathbb{K},\ k,\ i,\ t$	Clients, client set, and indices.
$D_k,\ n_k,\ (x_i, y_i)$	Client data, size, and input–label pair.
$D_{\mathrm{cal}},\ B_k,\ \lvert B_k\rvert$	Calibration set, mini-batch, and its size.
$\mathcal{S}_t,\ K_t$	Participating clients and their number.
$x,\ \mathcal{C}(x)$	Record and clients containing it.
$h_\theta,\ \theta$	Model and its parameters.
$\theta^t,\ \theta^{t+1}$	Global parameters before and after round t.
$\theta_k,\ \theta_k^{t+1}$	Local and released client parameters.
ω_k, ϕ_k, ψ_k	Encoder, policy-network, and classifier parameters.
$E_{\omega_k}, g_{\phi_k}, C_{\psi_k}$	Encoder, policy network, and classifier.
$h,\ \mathbb{R}^h$	Latent dimension and space.
$z,\ z',\ \widehat{z}$	Encoded, masked, and perturbed representations.
$\mathbf{m}, \mathbf{1}$	Feature mask and all-ones vector.
$\boldsymbol{s}, \bar{\boldsymbol{s}}$	Risk vector and its normalized form.
$\mathcal{Q}_k = (\varepsilon_k, \boldsymbol{\tau}_k)$	Client policy profile.
(ε_k, δ)	Record-level DP parameters.
$\boldsymbol{\tau}_k$	Policy-target vector.
$\tau_{k,\min}, \tau_{k,\mathrm{ret}}, \tau_{k,\mathrm{des}}$	Targets for the three policy tests.
$\mathbf{q}_k(\theta; D_{\mathrm{cal}})$	Policy-test vector on calibration data.
$q^{\min}, q^{\mathrm{ret}}, q^{\mathrm{des}}$	Per-record policy scores.
$q_k^{\min,t}, q_k^{\mathrm{ret},t}, q_k^{\mathrm{des},t}$	Calibration-set policy scores.
$\Delta_k^t,\ \alpha_k^t$	Policy residual and PRA weight.
$\lambda_{\min,k}, \lambda_{\mathrm{ret},k}, \lambda_{\mathrm{des},k}$	Policy-loss coefficients.
$\sigma_{\mathrm{rep},k}, \kappa, \xi$	Noise scale, score scale, and stabilizer.
$\mathcal{L}_{\mathrm{cls},k}(i), \mathcal{L}_k(i)$	Classification and complete local losses.
$\mathcal{L}_{\min,k}(i), \mathcal{L}_{\mathrm{ret},k}(i),$ $\mathcal{L}_{\mathrm{des},k}(i)$	Three policy losses.
$\bar{g}_{k,i},\ G_k$	Clipped gradient and clipping bound.
$\widetilde{g}_k, \sigma_{\mathrm{DP},k}$	Noisy gradient and DP noise multiplier.
ρ	Maximum residual change.
$\sigma(\cdot), \odot, [u]_+, \nabla$	Sigmoid, product, positive part, and gradient.

Table 6: Summary of notation used in the main paper.

B COMPLETE TRAINING PROCEDURE

Algorithm 1 expands the round-level description in the main paper into the complete training loop. The participating set must be public or data-independent, and each participating client's accountant must cover all of its local steps and releases.

FedPRA adds one scalar residual to each local-model upload. The server computes all PRA weights in $\mathcal{O}(|\mathcal{S}_t|)$ time. Raw records, latent representations, per-record gradients, and individual policy-test values are not transmitted.

Algorithm 1 Complete FedPRA Training Procedure

Input: Initial global parameters θ^0; client data D_k and profiles $\mathcal{Q}_k$; public calibration set D_{cal}
Output: Final global parameters

Initialize each client's privacy accountant.
Set the communication-round index to $t = 0$.
while the final communication round has not been completed **do**
  Determine the public or data-independent set $\mathcal{S}_t$.
  Server broadcasts θ^t to every $k \in \mathcal{S}_t$.
  for each client $k \in \mathcal{S}_t$ in parallel **do**
    Initialize $\theta_k = (\omega_k, \phi_k, \psi_k)$ from θ^t.
    for each local epoch and mini-batch B_k **do**
      Compute the classification loss and three policy losses.
      Clip every per-record gradient to G_k.
      Add Gaussian noise calibrated by $\sigma_{\text{DP},k}$ and update θ_k.
      Update Client k's privacy accountant.
    end for
    Evaluate the three tests of θ_k^{t+1} on D_{cal}.
    Compute Δ_k^t and upload only $(\theta_k^{t+1}, \Delta_k^t)$.
  end for
  Clip each residual to $[0, 1]$ and compute α_k^t.
  Set $\theta^{t+1} = \sum_{k \in \mathcal{S}_t} \alpha_k^t \theta_k^{t+1}$.
  Increment t.
end while

C Expanded Proofs

C.1 Proof of Proposition 1: Residual Fidelity

We show that the normalized residual is zero exactly when all three directional policy constraints are satisfied. For Client k at round t, the residual is

$$\Delta_k^t = \frac{1}{3}\Big([\tau_{k,\min} - q_k^{\min,t}]_+ + [q_k^{\text{ret},t} - \tau_{k,\text{ret}}]_+ + [q_k^{\text{des},t} - \tau_{k,\text{des}}]_+\Big), \tag{12}$$

where the three q_k terms are the calibration-test scores, the three τ_k terms are their corresponding policy targets, and $[u]_+ = \max(u, 0)$.

Proof. Each term in Eq. equation 12 is nonnegative. Therefore, their average is zero if and only if every term is zero. This occurs exactly when

$$q_k^{\min,t} \geq \tau_{k,\min}, \qquad q_k^{\text{ret},t} \leq \tau_{k,\text{ret}}, \qquad q_k^{\text{des},t} \leq \tau_{k,\text{des}}.$$

Hence, $\Delta_k^t = 0$ if and only if all three directional policy constraints are satisfied on D_{cal}. □

This result concerns the configured tests on the public calibration set; it does not establish legal compliance.

Policy residual and privacy budget. The privacy budget ε_k and policy residual Δ_k characterize different aspects of a client. The former is determined by the differential privacy mechanism, whereas the latter measures the degree to which the learned representation satisfies the configured policy targets. FedPRA therefore does not assume an ordering relationship between ε_k and Δ_k; privacy accounting remains independent of the policy-aware aggregation signal.

C.2 Proof of Proposition 2: Transcript Privacy

Proof. The released transcript contains the client parameter–residual pairs and all subsequent server outputs. We first establish privacy for each pair and then compose the guarantees across clients.

By assumption, Client k's complete released parameter sequence satisfies (ε_k, δ) record-level DP. Its residual is computed only from these private parameters, the public calibration set, and public

policy targets. Therefore, the residual is post-processing of the DP parameters, and releasing the parameter–residual sequence preserves the same (ε_k, δ) guarantee.

For disjoint client datasets, changing one record affects only the mechanism of the client containing that record. Parallel composition therefore gives

$$\left(\max_{k\in\mathbb{K}} \varepsilon_k, \delta\right) \text{-DP.}$$

In particular, a record held by Client k receives that client's (ε_k, δ) guarantee.

For overlapping client datasets, changing a record x may affect every client in $\mathcal{C}(x)$, where $\mathcal{C}(x)$ is the set of clients containing that record. Sequential composition therefore gives

$$\left(\sum_{k\in\mathcal{C}(x)} \varepsilon_k,\ |\mathcal{C}(x)|\delta\right) \text{-DP.}$$

Finally, PRA, global aggregation, and any downstream attack use only the released transcript. They are therefore post-processing operations and cannot increase the corresponding privacy loss established above. □

C.3 Proof of Proposition 3: PRA Prioritization and Weight Bounds

We show that PRA assigns larger weights to clients with smaller residuals while bounding every client's influence. For Client $k \in \mathcal{S}_t$, the PRA weight is

$$\alpha_k^t = \frac{(1+\Delta_k^t)^{-1}}{\sum_{j\in\mathcal{S}_t}(1+\Delta_j^t)^{-1}}, \tag{13}$$

where $K_t = |\mathcal{S}_t|$. Because $\Delta_k^t \in [0,1]$, every unnormalized score $(1+\Delta_k^t)^{-1}$ lies in $[1/2, 1]$.

Proof. The function $v \mapsto (1+v)^{-1}$ is decreasing. Thus, for any participating clients i and j,

$$\Delta_i^t \le \Delta_j^t \quad\Longrightarrow\quad (1+\Delta_i^t)^{-1} \ge (1+\Delta_j^t)^{-1}.$$

Because all clients share the same positive normalizer in Eq. equation 13, it follows that

$$\alpha_i^t \ge \alpha_j^t.$$

The largest possible weight occurs when Client k has an unnormalized score of 1 and every other participating client has a score of $1/2$. Therefore,

$$\alpha_k^t \le \frac{1}{1+(K_t-1)/2} = \frac{2}{K_t+1}.$$

Similarly, the smallest possible weight occurs when Client k has a score of $1/2$ and every other client has a score of 1. Hence,

$$\alpha_k^t \ge \frac{1/2}{1/2+(K_t-1)} = \frac{1}{2K_t-1}.$$

Finally, the common normalizer cancels in the ratio

$$\frac{\alpha_i^t}{\alpha_j^t}.$$

Since every unnormalized score lies in $[1/2, 1]$, the ratio of any two PRA weights lies in $[1/2, 2]$. □

Corollary 2 (Exact relative prioritization)**.** *For any participating clients i and j,*

$$\frac{\alpha_i^t}{\alpha_j^t} = \frac{1+\Delta_j^t}{1+\Delta_i^t}. \tag{14}$$

Thus, the relative PRA priority depends only on the two residuals and is independent of the other participating clients.

Proof. The common normalization term in the PRA weights cancels:

$$\frac{\alpha_i^t}{\alpha_j^t} = \frac{(1+\Delta_i^t)^{-1}}{(1+\Delta_j^t)^{-1}} = \frac{1+\Delta_j^t}{1+\Delta_i^t}.$$

□

C.4 Proof of Proposition 4: Calibration Stability

We show that bounded changes in the calibration residuals produce bounded changes in the PRA weights. We also show that residual ordering is preserved when the original residual gap is sufficiently large.

Proof. Let $\widetilde{\Delta}_k^t$ denote Client k's residual after perturbing the calibration evaluation. Assume that every residual changes by at most ρ:

$$|\Delta_k^t - \widetilde{\Delta}_k^t| \le \rho \qquad \text{for all } k \in \mathcal{S}_t.$$

Define the corresponding unnormalized PRA scores as

$$u_k = (1+\Delta_k^t)^{-1}, \qquad \widetilde{u}_k = (1+\widetilde{\Delta}_k^t)^{-1}.$$

Because $v \mapsto (1+v)^{-1}$ is 1-Lipschitz on $[0,1]$,

$$|u_k - \widetilde{u}_k| \le |\Delta_k^t - \widetilde{\Delta}_k^t| \le \rho.$$

Summing over the participating clients gives

$$\sum_{k\in\mathcal{S}_t} |u_k - \widetilde{u}_k| \le K_t \rho. \tag{15}$$

Both the original and perturbed residuals lie in $[0,1]$. Hence, their unnormalized scores lie in $[1/2, 1]$, and both normalizing sums are at least $K_t/2$. Moreover,

$$\left| \sum_{j\in\mathcal{S}_t} u_j - \sum_{j\in\mathcal{S}_t} \widetilde{u}_j \right| \le \sum_{j\in\mathcal{S}_t} |u_j - \widetilde{u}_j|.$$

Applying the triangle inequality to the normalized PRA weights yields

$$\begin{aligned} &\sum_{k\in\mathcal{S}_t} \left| \frac{u_k}{\sum_{j\in\mathcal{S}_t} u_j} - \frac{\widetilde{u}_k}{\sum_{j\in\mathcal{S}_t} \widetilde{u}_j} \right| \\ &\qquad \le \frac{2\sum_{k\in\mathcal{S}_t} |u_k - \widetilde{u}_k|}{K_t/2} \le 4\rho. \end{aligned} \tag{16}$$

Thus, the L_1 change in the PRA weight vector is at most 4ρ.

Finally, suppose that two clients i and j satisfy

$$\Delta_i^t < \Delta_j^t \quad \text{and} \quad \Delta_j^t - \Delta_i^t > 2\rho.$$

Then

$$\widetilde{\Delta}_i^t \le \Delta_i^t + \rho < \Delta_j^t - \rho \le \widetilde{\Delta}_j^t.$$

Therefore, their residual ordering cannot reverse. Since the PRA score decreases as the residual increases, their PRA priority is also preserved. □

Calibration robustness. Proposition 4 also shows that bounded perturbations in calibration-based residuals lead to bounded changes in the PRA weight vector, while sufficiently separated residuals preserve their ordering. Thus, FedPRA does not require exact residual estimates to maintain the same aggregation priority under moderate calibration variation. In practice, the public calibration set should capture the policy-relevant characteristics of the deployment setting.

C.5 Proof of Proposition 5: Participation Stability

We bound the change in the PRA weight vector when one client joins or leaves. The other clients' unnormalized scores are held fixed, and an absent client is assigned weight zero so that the two weight vectors have the same dimension.

Proof. Let K_t be the number of participating clients before the change. Since every residual lies in $[0, 1]$, every unnormalized PRA score lies in $[1/2, 1]$.

First, suppose that one client joins. After renormalization, the total decrease in the existing clients' weights equals the weight assigned to the new client. Including the new client's previously zero coordinate, the L_1 change between the two weight vectors is therefore twice the new client's weight.

The new client's weight is maximized when its unnormalized score is 1 and all K_t existing clients have scores of $1/2$. Hence, its weight is at most

$$\frac{1}{1 + K_t/2} = \frac{2}{K_t + 2}.$$

Therefore, the L_1 change caused by one joining client is at most

$$\frac{4}{K_t + 2}.$$

Next, suppose that one client leaves. Its previous weight disappears, while the remaining clients collectively gain the same amount after renormalization. Thus, the L_1 change is twice the departing client's previous weight.

This weight is maximized when the departing client's score is 1 and the other $K_t - 1$ clients have scores of $1/2$. Therefore, its weight is at most

$$\frac{1}{1 + (K_t - 1)/2} = \frac{2}{K_t + 1},$$

and the resulting L_1 change is at most

$$\frac{4}{K_t + 1}.$$

Hence, the PRA weight vector changes by at most $4/(K_t + 2)$ when one client joins and by at most $4/(K_t + 1)$ when one client leaves. □

Generality across participating clients. Although the experiments use three clients, the PRA formulation and Proposition 5 are defined for an arbitrary participating set $\mathcal{S}_t$. The resulting aggregation rule therefore does not rely on the three-client setting.

D Reproducibility Details

D.1 Policy-Test Interpretation and Target Configuration

The three tests describe complementary operational properties of the learned representation. Feature minimization measures the proportion of latent dimensions suppressed by the learned mask. Risk-aware non-retention measures retained energy in dimensions that most affect the current prediction. Structural design limits representation magnitude under a fixed architecture, scale parameter κ, and training configuration. These quantities are differentiable training and evaluation signals. They are not direct measures of information leakage, substitutes for the DP accountant, or certificates of legal compliance.

The numerical targets used in the experiments are fixed before training and create three distinct, feasible policy profiles. The regulation-inspired profile names indicate different operational emphases; the numerical values are not prescribed by GDPR, PIPL, or HIPAA. In deployment, an organization must configure and validate its targets using its own operational requirements and representative public calibration data. Because the structural score depends on representation scale, comparisons

assume the same model architecture, parameterization, κ, and training protocol across clients and methods.

The calibration set, target vectors, policy-test definitions, and released local models are available to the server. The server can therefore recompute every reported residual before aggregation. This provides an audit check under the stated honest-client threat model, but it does not provide robustness against malicious model manipulation.

D.2 Implementation and Privacy Accounting

This section reports the optimization, privacy-accounting, and evaluation settings used in the experiments. Table 7 summarizes the complete configuration. For each client, the DP noise multiplier is calibrated using a Rényi differential privacy (RDP) accountant so that the privacy loss accumulated over all local updates and participating rounds satisfies the assigned total budget.

Setting	Value
FL and DP software	Flower, PyTorch, and Opacus
Hardware	NVIDIA RTX 4090 GPU
Global rounds; local epochs	10; 5 per round
Optimizer; learning rate	Adam; 10^{-3}
Batch size; latent dimension	256; $h = 64$
Clipping bound	$G_k = 1.0$ for every client
Privacy budgets	$(\varepsilon_A, \varepsilon_B, \varepsilon_C) = (1, 2, 3)$
Additive slack	$\delta = 10^{-5}$
Accountant	RDP, with orders from 2 to 256
DP noise	Client-specific $\sigma_{\mathrm{DP},k}$ selected by the accountant
Representation settings	$\sigma_{\mathrm{rep},k} = 0.1$, $\kappa = 1.0$, $\xi = 10^{-8}$
Independent runs	Ten runs with seeds 0–9
Reported checkpoint	Final round of every run

Table 7: Implementation and privacy-accounting settings.

Each seed defines an independent run and controls model initialization, mini-batch sampling, and Gaussian DP noise. Reported statistics are computed from the ten final-round checkpoints; communication rounds within the same run are not treated as independent replicates. For each client, the accountant composes all DP-SGD steps and parameter releases across participating rounds. When an original record is selected repeatedly, the accountant uses its largest per-step inclusion probability.

D.3 Privacy-Accounting Results

This subsection verifies that each client's realized privacy loss remains below its assigned target. Table 8 reports the client-specific dataset size, noise multiplier, nominal Poisson sampling rate, and achieved privacy loss. All clients use $\delta = 10^{-5}$, five local epochs per round, and ten communication rounds.

Client	Target ε_k	$\sigma_{\mathrm{DP},k}$	Nominal q_k	Realized ε_k
A	1	1.42	0.000326	0.99
B	2	1.05	0.000184	1.98
C	3	0.85	0.002847	2.97

Table 8: Client-specific privacy-accounting results. The displayed sampling rates are the nominal loader rates before Client C's rare-class resampling adjustment.

The nominal Poisson sampling rate equals the logical batch size, 256, divided by the number of private training records. For Clients A and B, this is also the per-step accounting rate. Client C reuses rare original records during resampling; therefore, its realized privacy loss is calculated

using the largest induced inclusion probability of any original record, rather than treating repeated selections as distinct records. The achieved ε_k values remain slightly below their targets because the noise multipliers are conservatively calibrated.

D.4 Data Partitions and Class Handling

CIC-IDS2017 contains $2{,}830{,}743$ records with 78 features. The private-training, public-calibration, and test splits contain $2{,}264{,}594$, $141{,}537$, and $424{,}612$ records, respectively. Positive and negative infinities are treated as missing values, and the 2,867 affected rows are median-imputed without removing any records. Imputation and scaling parameters are fitted on the private-training split and then applied to the calibration and test splits. Accordingly, the privacy guarantee is conditional on treating the fitted preprocessing transformation as fixed across neighboring datasets.

UNSW-NB15 contains $157{,}807$ private-training records, $17{,}534$ public-calibration records, and $82{,}332$ test records. For both benchmarks, the private-training data are partitioned across three clients using a class-wise Dirichlet distribution with concentration 0.5.

Table 9 reports the CIC-IDS2017 client partitions and illustrates their data-volume imbalance. The corresponding training-volume weights are $(0.347, 0.613, 0.040)$ for Clients A, B, and C.

Client	Private training	Test
A	785,965	147,368
B	1,388,695	260,380
C	89,934	16,864

Table 9: Per-client record counts for CIC-IDS2017.

CIC-IDS2017 labels are consolidated into nine classes: BENIGN, Bot, DDoS, DoS, Heartbleed, Infiltration, Other, Patator, and PortScan. Class-weighted cross-entropy uses class frequencies from each client's private-training data. Because Heartbleed and Infiltration are scarce at Client C, each class is resampled to 500 training samples. Repeated draws reuse the same original records and are not treated as new records. The privacy accountant therefore uses the largest per-step inclusion probability induced by resampling. UNSW-NB15 uses no resampling, duplication, or synthetic data.

E Additional Experimental Results

E.1 Residual and PRA Weight Trajectories

This analysis examines how the client-local policy residuals used by PRA and their corresponding aggregation weights evolve during training. For each run, PRA weights are computed from the reported client residuals. The residuals and weights are then averaged separately over ten independent runs.

(a) Policy residuals

Round	Client A	Client B	Client C
1	0.0872 ± 0.006	0.0898 ± 0.005	0.0754 ± 0.007
2	0.0821 ± 0.005	0.0845 ± 0.005	0.0682 ± 0.006
3	0.0765 ± 0.005	0.0789 ± 0.004	0.0618 ± 0.005
4	0.0712 ± 0.004	0.0735 ± 0.004	0.0556 ± 0.005
5	0.0658 ± 0.004	0.0681 ± 0.004	0.0492 ± 0.004
6	0.0612 ± 0.004	0.0634 ± 0.003	0.0438 ± 0.004
7	0.0589 ± 0.003	0.0608 ± 0.003	0.0398 ± 0.003
8	0.0568 ± 0.003	0.0589 ± 0.003	0.0365 ± 0.003
9	0.0558 ± 0.003	0.0581 ± 0.003	0.0342 ± 0.003
10	0.0553 ± 0.003	0.0577 ± 0.003	0.0330 ± 0.003

(b) PRA aggregation weights

Round	Client A	Client B	Client C
1	0.331 ± 0.003	0.330 ± 0.003	0.339 ± 0.003
2	0.332 ± 0.003	0.331 ± 0.003	0.337 ± 0.003
3	0.332 ± 0.003	0.331 ± 0.003	0.337 ± 0.003
4	0.332 ± 0.002	0.331 ± 0.002	0.337 ± 0.003
5	0.332 ± 0.002	0.331 ± 0.002	0.337 ± 0.002
6	0.332 ± 0.002	0.331 ± 0.002	0.337 ± 0.002
7	0.331 ± 0.002	0.331 ± 0.002	0.338 ± 0.002
8	0.331 ± 0.002	0.330 ± 0.002	0.339 ± 0.002
9	0.331 ± 0.002	0.330 ± 0.002	0.339 ± 0.002
10	0.331 ± 0.002	0.330 ± 0.002	0.339 ± 0.002

Table 10: Round-wise policy residuals and PRA weights on CIC-IDS2017. Values are mean $\pm$ standard deviation over ten independent runs.

As shown in Table 10, the residuals decrease across all three clients, indicating progressively smaller violations of their configured policy targets. Client C has the smallest residual in every round and consequently receives the largest PRA weight. Meanwhile, the weights vary only slightly and sum to approximately one in every round. These results demonstrate the expected inverse relationship between residuals and PRA weights and show that PRA remains stable throughout training.

E.2 Residual-Calculation Audit

Table 11 verifies how the final policy residual of each client is calculated. Its purpose is to make the residual ordering used by PRA directly reproducible. Because feature minimization specifies a lower-bound target, its violation is measured as a shortfall. Non-retention and structural design specify upper-bound targets, so their violations are measured as excesses.

Client	Min. shortfall	Ret. excess	Design excess	Residual
A	0.100	0.066	0.000	0.0553
B	0.123	0.050	0.000	0.0577
C	0.092	0.007	0.000	0.0330

Table 11: Directional policy violations and final residuals on CIC-IDS2017. Each residual is the mean of the three violations.

For Client A, the three violations are 0.100, 0.066, and 0, giving

$$(0.100 + 0.066 + 0)/3 = 0.0553.$$

Applying the same calculation gives 0.0577 for Client B and 0.0330 for Client C. Therefore,

$$\Delta_C^t < \Delta_A^t < \Delta_B^t.$$

The final residual averaged across the three clients is

$$(0.0553 + 0.0577 + 0.0330)/3 = 0.0487,$$

which rounds to 0.049. Because PRA assigns greater weight to a smaller residual, the ordering explains why Client C receives the largest final PRA weight. The table therefore verifies the reported residuals, their cross-client mean, and their use in aggregation.

The matched aggregation result further illustrates that policy-aware weighting can depart substantially from volume weighting. Client C receives only 4.0% under volume weighting but 33.8% under PRA because it has the smallest policy residual. This demonstrates that client data volume is not the sole aggregation criterion in FedPRA.

E.3 End-to-End Runtime

Table 12 reports the end-to-end runtime on CIC-IDS2017 under the common hardware and training protocol.

Method	Runtime (h)	Overhead vs. DP-FedAvg
FedAvg	3.28 ± 0.05	–
DP-FedAvg	5.10 ± 0.10	0%
FedPRA	5.61 ± 0.12	10.0%

Table 12: End-to-end runtime on CIC-IDS2017. Values are mean $\pm$ standard deviation over ten runs.

FedPRA requires 0.51 additional hours relative to DP-FedAvg, corresponding to a 10.0% end-to-end overhead.

E.4 Assumptions and Scope

The analysis assumes public calibration data and policy targets, privacy accounting over all local steps and releases, fixed or data-independent client participation, and honest clients with an honest-but-curious server. The experiments use disjoint client partitions, while overlapping data are analyzed theoretically. The policy residuals measure satisfaction of the configured operational tests and do not constitute legal-compliance certificates. Larger-scale and adversarial evaluations remain future work.