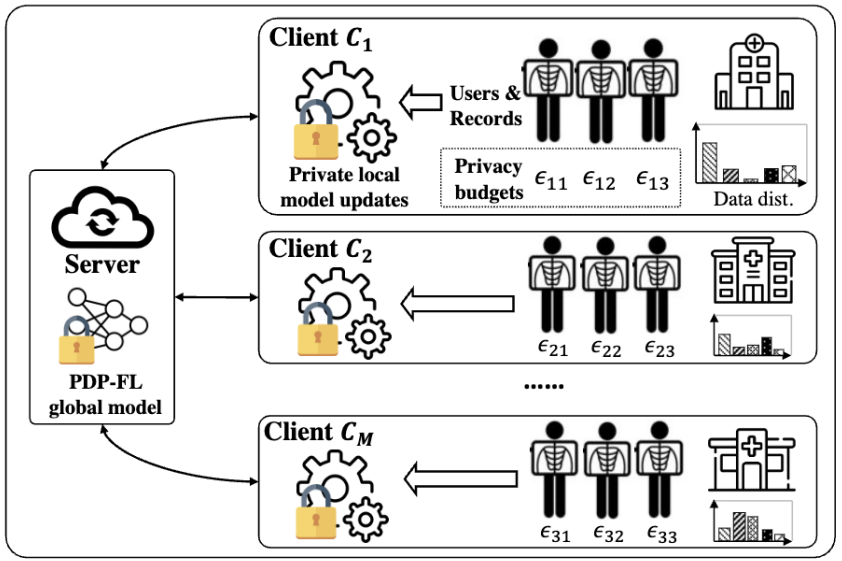
Cross-silo Federated Learning with Record-level Personalized Differential Privacy

Contents

Chapter 1: Problem Setting

Cross-silo Federated Learning :

Multiple clients collaboratively train a joint model under the coordination of a central server while keeping the data local. Each client holds a local dataset comprising personal data records.



Record-level :

Each record is associated with a single user who does not contribute the same record or multiple records the multiple client simultaneously.

Personalized Differential Privacy:

Each record has its own privacy budget, reflecting the user’s desired level of protection for their sensitive data.

Threats:

untrusted clients or third parties, accessible to the intermediate or final global model;
honest-but-curious server, accessible to the intermediate model updates

Chapter 2: Preliminaries

I. Federated Learning : A collaborative learning paradigm where multiple clients jointly train a global model without sharing raw data

II. Differential Privacy: A de facto standard for protecting sensitive data.

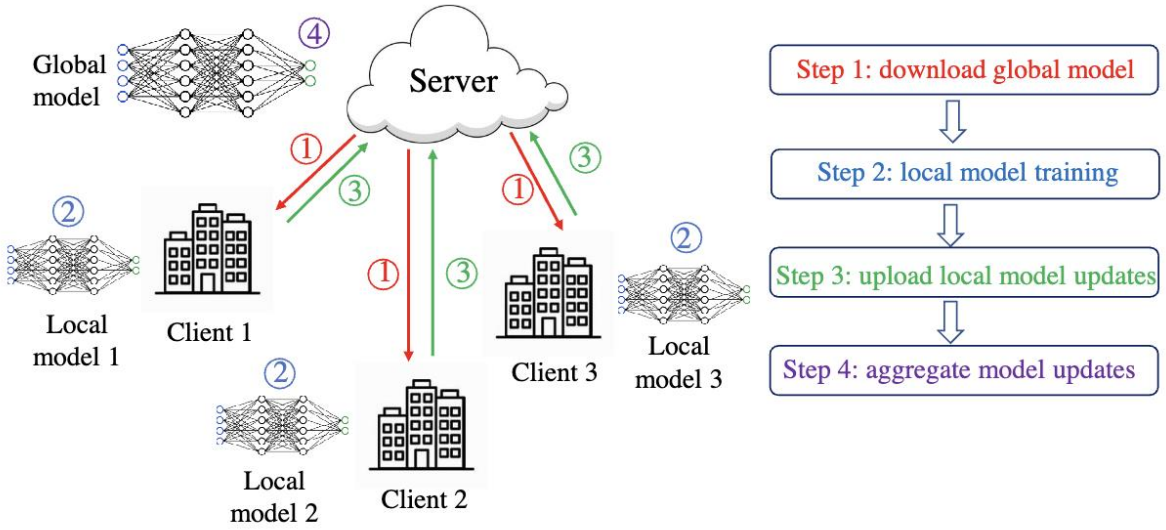
III. Rényi Differential Privacy: An alternative privacy framework that quantifies the privacy guarantee using Rényi divergence, enabling tighter composition analysis.

Chapter 2: Preliminaries

I. Federated Learning (FL)

Definition:

A learning paradigm where multiple clients collaboratively train a shared global model under the coordination of a central server, without sharing raw data.



Loss Function:

$$\mathbf{w}^{\star} \triangleq \min_{\mathbf{w}} F(\mathbf{w}), \text{ where } F(\mathbf{w}_t) \triangleq \sum_{k=1}^{K} p_k L_k(\mathbf{w}_t^k)$$

In this article, the authors adopt FedAVG as the aggregation function, with uniform client weights pk = 1/K

Chapter 2: Preliminaries

II. Differential Privacy (DP)

A rigorous mathematical framework that formally defines data privacy, practically adding an appropriate amount of noise to the output of statistical queries or to the model gradients, can resist multiple attacks.

Definition:

The randomized algorithm $\mathcal{A}$:$\mathbb{X}^n \rightarrow \mathbb{Y}$ satisfies (ϵ, δ)-DP if any two neighboring datasets D and D' that differ in only a single entry and $D \simeq D' \in$ Xn , we have

$$\forall S \subseteq \mathbb{Y} : \Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in S] + \delta$$

$\epsilon > 0$ controls the level of privacy guarantee in the worst case
$\delta \geqslant 0$ is the failure probability that the property does not hold

Chapter 2: Preliminaries

II. Differential Privacy

Differentially Private SGD (DP-SGD):

Proposed by Abadi et al. in 2016, a framework adopting DP in SGD, by clipping per-sample gradients to a fixed norm and adding Gaussian noise before aggregation, thus limiting the influence of any individual record and providing formal (ϵ, δ)-DP guarantees.

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Chapter 2: Preliminaries

II. Differential Privacy

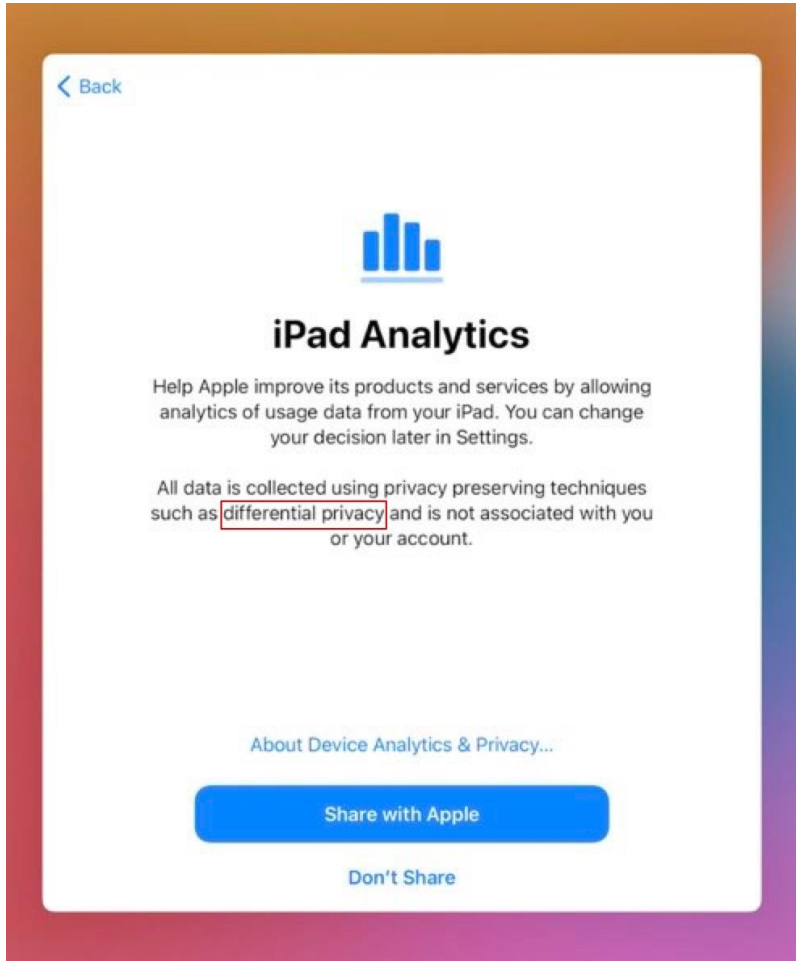
Apple
https://www.apple.com › privacy › docs › Differe... PDF

Differential Privacy

Differential privacy **transforms the information shared with Apple** before it ever leaves the user's device such that Apple can never reproduce the true data. The ...
3 pages

Differential Privacy

There are situations where Apple can improve the user experience by getting insight from what many of our users are doing, for example: What new words are trending and might make the most relevant suggestions? What websites have problems that could affect battery life? Which emoji are chosen most often? The challenge is that the data which could drive the answers to those questions—such as what the users type on their keyboards—is personal.



Chapter 2: Preliminaries

III. Rényi Differential Privacy

Rényi differential privacy utilizes the asymmetric measure of Rényi divergence to quantify the privacy guarantee. When $\alpha \neq 1$, the Rényi divergence of order α from distribution Q to P is:

$$D_\alpha(P\|Q) \triangleq \frac{1}{\alpha-1}\log \mathbb{E}_{o\sim Q}\left[\left(\frac{P(o)}{Q(o)}\right)^\alpha\right]$$

Let P =$\mathcal{A}$ (D) and Q = $\mathcal{A}$ (D'), then (α, ρ)-RDP is achieved by achieved by simultaneously bounding the Rényi divergence of two directions, denoted by $D_\alpha^{\Leftrightarrow}(\mathrm{P}||\mathrm{Q}) \triangleq \max\{D_\alpha(P||Q), D_\alpha(Q||P)\}$

Definition:
A randomized mechanism $\mathcal{A}$ satisfies (α, ρ)-RDP with order $\alpha \in (1, \infty)$ if for any pair of neighboring datasets D, $D' \in \mathbb{D}$, it holds that

$$D_\alpha^{\Leftrightarrow}(\mathcal{A}(D)\,||\,\mathcal{A}(D')) \leqslant \rho$$

Transition From RDP to DP:

If $\mathcal{A}$ is an (α, ρ)-RDP mechanism, it also satisfies ($\rho + \frac{\log 1/\delta}{\alpha - 1}$, δ)-DP for any $0 < \delta < 1$

Chapter 3: Methodology

I. Algorithm Overview : Two-stage training with FedAVG and DP-SGD

II. Privacy Analysis Key Results: RDP-based bound on per-record privacy cost

III. Simulation-CurveFitting: Approximate the ϵ–q relation to compute sampling probability for each record

Chapter 3: Methodology

I. Algorithm Overview

Step 1:
Pre-compute the sampling probability for each record

Step 2:
Sample clients in the current round with probability λ to Run FedAVG

Step 3:
Perform τ steps of DP-SGD on selected clients, using their records' personalized sampling probabilities

Step4:
Aggregate the updated models from the selected clients to from the global model

Algorithm 2: Record-level Personalized Differentially Private Federated Learning (rPDP-FL, Pseudocode)

input : M clients with their local datasets $(D_1, \ldots, D_M)$; the total communication round T and the local SGD step τ; the client-level sampling probability λ.

// Initialization

foreach *client* $C_i \in C$ **do** *in parallel*
  $\{q_{i,j}\}_{j\in[|D_i|]} \leftarrow$ (pre-computation of sampling probabilities for all records)

for $t \in [T]$ **do**
  // Client-level Poisson sampling with the uniform sampling probability λ
  $\check{C}^t \leftarrow$ (a random subset drawn from $[M]$)
  foreach *client* $C_i \in \check{C}^t$ **do** *in parallel*
    for $r \in [\tau]$ **do**
      // Record-level Poisson sampling with non-uniform sampling probabilities $\{q_{i,j}\}_{j\in[|D_i|]}$
      $S^r \leftarrow$ (a random mini-batch drawn from D_i)
      // Differentially private SGD
  // The central server averages the collected noisy model updates and obtains the updated global model parameters

Chapter 3: Methodology

II. Privacy Analysis Key Results

Previous Work:

$$\rho_{PoiSG}(\alpha, q) \leq \frac{1}{\alpha - 1} \log \left\{ (1-q)^{\alpha-1}(\alpha q - q + 1) + \sum_{\ell=2}^{\alpha} \binom{\alpha}{\ell} (1-q)^{\alpha-\ell} q^{\ell} e^{(\ell-1)\rho(\ell)} \right\}$$

When we have untrusted clients or third parties, the accumulative ρ could be written as:

$$\rho_{i,j}^{\tau,\lambda}(\alpha, q_{i,j}) \leq \frac{1}{\alpha - 1} \ln \left\{ 1 - \lambda + \lambda e^{(\alpha-1)\rho_{i,j}^{\tau}(\alpha, q_{i,j})} \right\}$$

When we have honest-but-curious server, the accumulative ρ could be written as :

$$\rho_{i,j}^{\tau}(\alpha, q_{i,j}) \leq \frac{\tau}{\alpha - 1} \ln \left\{ (1-q_{i,j})^{\alpha-1}(\alpha q_{i,j} - q_{i,j} + 1) + \sum_{\ell=2}^{\alpha} \binom{\alpha}{\ell} (1-q_{i,j})^{\alpha-\ell} q_{i,j}^{\ell} e^{(\ell-1)\rho_G(\ell)} \right\}$$

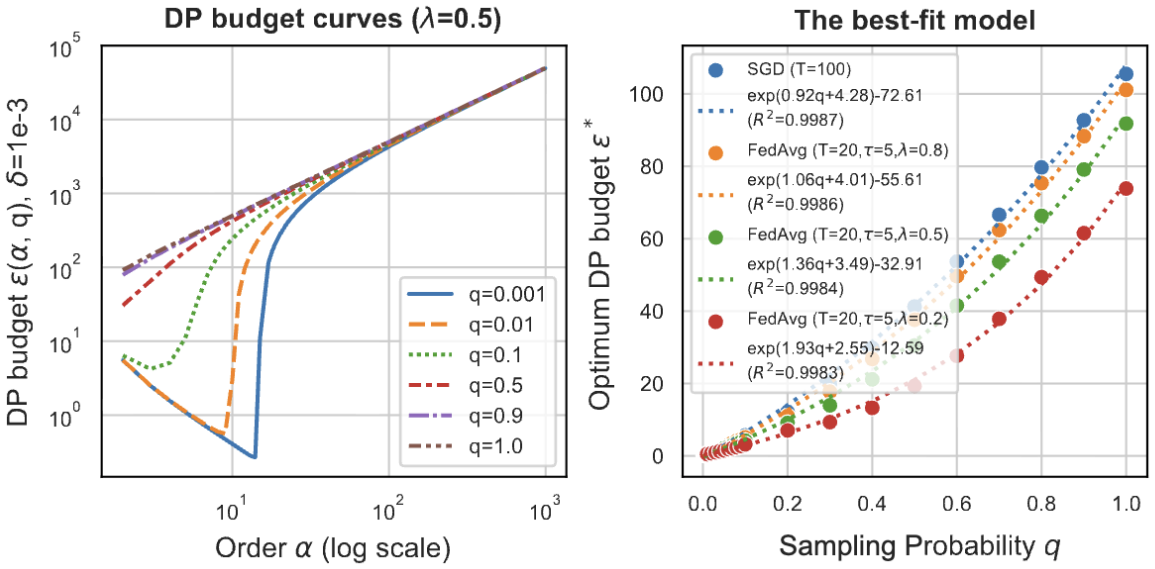
For any $\delta \in (0,1)$, the random algorithm $\mathcal{A}_{FL}(\mathcal{D})$ satisfies $(\hat{\varepsilon}_{i,j}^{*}, \delta)$-DP w.r.t. a specific record $d_{i,j} \in \mathcal{D}$, where

$$\hat{\varepsilon}_{i,j}^{*} \triangleq \min_{\alpha} \left(\rho_{FL}(\alpha, q_{i,j}) + \frac{\ln(1/\delta)}{\alpha - 1} \right)$$

Note that: (1) for untrusted clients or third parties, $\rho_{FL}(\alpha, q_{i,j}) \triangleq T\rho_{i,j}^{\tau,\lambda}(\alpha, q_{i,j})$; (2) for the honest-but-curious server, $\rho_{FL}(\alpha, q_{i,j}) \triangleq \lambda T \rho_{i,j}^{\tau}(\alpha, q_{i,j})$.

Chapter 3: Methodology

III. Simulation-CurveFitting (SCF)



$$\hat{\varepsilon}^* \approx f(q) \triangleq e^{a \cdot q + b} + c$$

R^2: coefficient of determination

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Algorithm 3: The Simulation-CurveFitting (SCF) strategy

input : The noise multiplier σ, the gradient clipping bound L, and the target DP parameter δ.

output : The sampling probability estimator

// Initialize two candidate lists of α, q

$\mathbb{A} \leftarrow$ a candidate list of RDP order $\alpha \in (1, \infty)$

$\Pi \leftarrow$ a candidate list of sampling probability $q \in [0, 1]$

foreach $q \in \Pi$ **do**

  // Numerical simulation analysis of PoiSGM with sampling probability q

  $\rho_{FL}(\alpha, q) \leftarrow$ (the RDP budget curve w.r.t. order $\alpha \in \mathbb{A}$ calculated based on Theorem 1)

  $\varepsilon(\alpha, \delta, q) = \rho_{FL}(\alpha, q) + \frac{\log 1/\delta}{\alpha - 1} \leftarrow$ (the DP budget curve w.r.t. order $\alpha \in \mathbb{A}$ calculated based on Lemma 1)

  $\varepsilon^*(\delta, q) = \min_{\alpha \in \mathbb{A}} \varepsilon(\alpha, \delta, q) \leftarrow$ (the optimum DP budget w.r.t. sampling probability q)

// Curve fitting

$f(q) \leftarrow$ (the best-fit mathematical model to the generated observations $\{(q, \varepsilon^*)\}_{q \in \Pi}$)

// The sampling probability estimator

$$F = \begin{cases} f^{-1}(\varepsilon), & 0 < \varepsilon < \varepsilon^*(1.0) \\ 1.0, & \varepsilon \geq \varepsilon^*(1.0) \end{cases}$$

return F

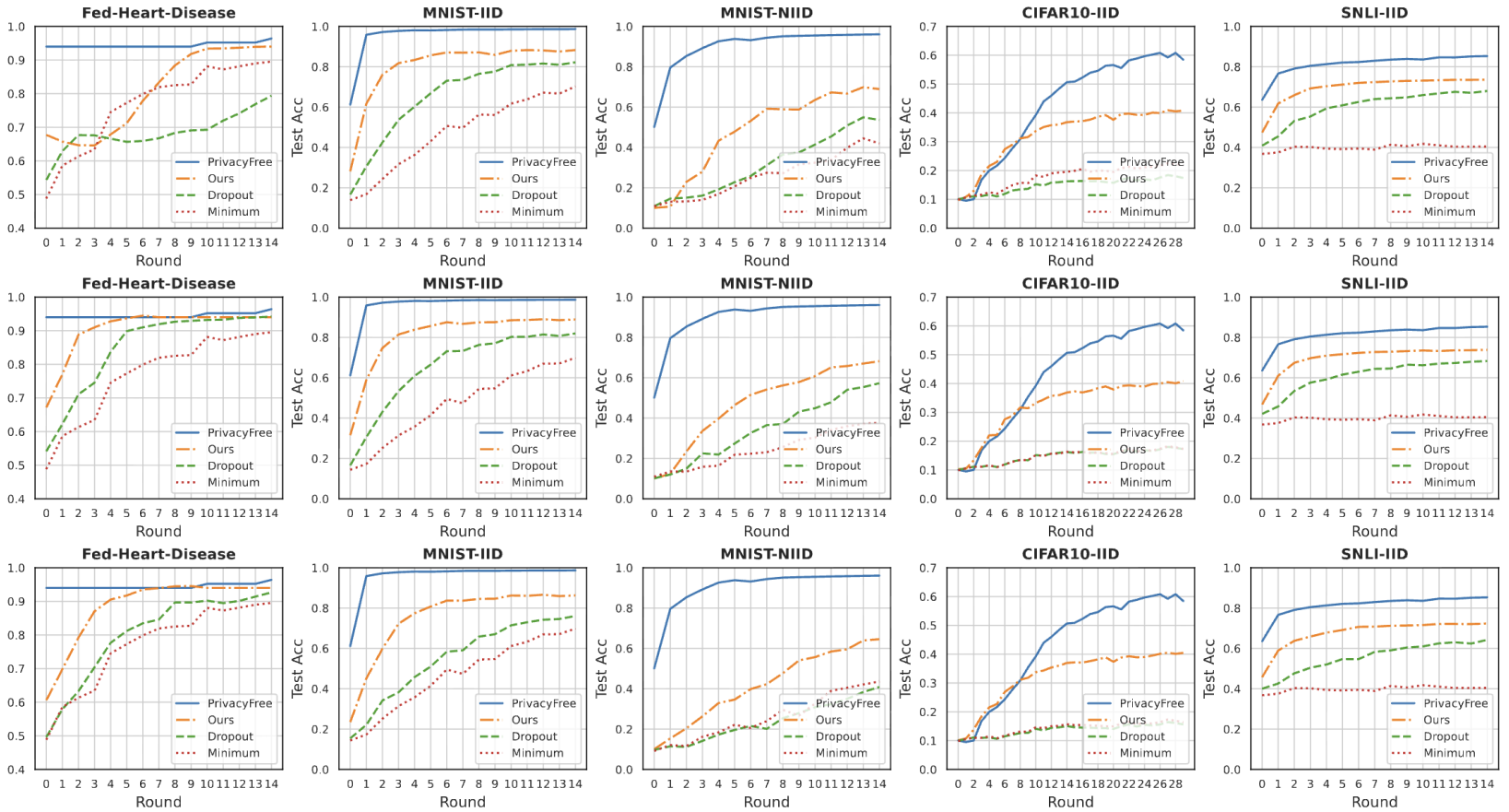
Chapter 4: Experiments

I. Efficient

Priv. Pref. Dist.	Method	Test Acc	Runtime (s)
Group-3: 3 unevenly sized privacy groups (54%-37%-9%) with privacy budgets [1., 2., 3.]	BinarySearch	0.7274	**1.69**
	SCF	0.7240	13.11
Group-100: 100 evenly sized privacy groups with privacy budgets [1., 1.05, ..., 5.95]	BinarySearch	0.8135	52.50
	SCF	0.8134	**13.32**
Individual-1000: per-record privacy budgets drawn from BoundedMixGauss	BinarySearch	0.6858	597.82
	SCF	0.6861	**14.10**

Chapter 4: Experiments

II. Effective



Chapter 5: Conclusion

I. **Novel and meaningful setting:**
   Personalized, record-level differential privacy in federate learning.

II. **Robust and detailed Privacy analysis:**
   Rigorous RDP-based accounting under the proposed setting.

III. **SCF strategy:**
   Efficient estimate sampling probabilities from given privacy budgets.

IV. **Practical Effectiveness:**
   Demonstrated efficiency and strong performance on benchmark datasets

Thanks for your listening