Enhancing Privacy in Personalized Differential Privacy under the Shuffle Model

E Chen[1]   Yang Cao[2][⋆]   Gennian Ge[1,3]

2023 年 7 月 20 日

Abstract. Different levels of privacy are required to satisfy users' varying attitudes locally, while a stringent privacy guarantee for the global model is also necessary centrally. Personalized Local Differential Privacy (PLDP) is suitable for preserving users' diverse local privacy, but it only provides a central privacy guarantee equivalent to the worst-case local privacy level. Therefore, achieving strong central privacy while ensuring personalized local privacy with a utility-promising model poses a challenging problem. This paper provides a tighter lower bound than Liu et al., which imposes less restrictive constraints on the parameters and is more robust.

Keywords: Personalized local differential privacy · Shuffle model · Privacy amplification.

1 Introduction

Privacy preservation is a critical concern in the era of big data, where vast amounts of sensitive information are collected and analyzed. Traditional approaches to privacy protection often rely on global privacy guarantees that treat all users' data equally. However, in many scenarios, users have different privacy requirements and attitudes towards data sharing. For instance, individuals

[⋆] Corresponding author

may have varying preferences regarding the sensitivity of their personal information, and some may be more willing to disclose certain aspects of their data for improved services or personalized recommendations.

To address the challenge of accommodating different privacy levels and preferences, the concept of Personalized Local Differential Privacy (**PLDP**) has emerged. PLDP aims to protect users' varying local privacy requirements while still providing a central privacy guarantee for the global model. Several recent works have explored PLDP by perturbing gradients with heterogeneous parameters, effectively preserving both local gradients and the global model (Chen et al. 2016; Li et al. 2020; Shen, Xia & Yu 2021; Yang, Wang & Wang 2021).

While PLDP offers personalized privacy protection, it is limited in terms of providing a central privacy guarantee that is equivalent to the weakest local privacy. This limitation motivates the exploration of alternative approaches that can simultaneously ensure strong central privacy and personalized local privacy guarantees. One promising solution is the shuffle model, which has been shown to amplify central privacy by randomly permuting data points after local perturbations (Bittau et al. 2017). The shuffle model offers the potential to achieve both strong central and local privacy guarantees while preserving utility.

However, existing studies on the shuffle model have primarily focused on scenarios where local privacy requirements are assumed to be uniform, referred to as **Uni-Shuffle** (Erlingsson et al. 2019; Balle et al. 2019; Girgis et al. 2021; Feldman, McMillan & Talwar 2022). These works have made significant contributions to privacy preservation under the shuffle model, but they do not fully address the need for personalized local privacy guarantees. To the best of our knowledge, there is currently a limited number of works that provide both strong central privacy for the global model and personalized local privacy guarantees, while also achieving strong utility of the global model. The best results in this regard have been presented by Liu et al. (2022), which is called **APES** for short; however, there is still room for improvement in their bounds, particularly in handling extreme values and sensitivity analysis.

In this paper, we aim to bridge this gap and propose a novel approach that combines the benefits of the shuffle model with personalized local privacy guarantees. By considering the varying local privacy requirements of different users, we strive to provide customized privacy protection while maintaining a high level of utility for the global model. To achieve this, we provide a enhanced lower bound of

APES (**E-APES** for short) with less restrictive parameter constraints, ensuring the robustness of our approach. Overall, this work advances the field of personalized local differential privacy by introducing a novel approach that addresses the challenges of privacy amplification, utility preservation, and individualized privacy requirements (see Table 1). Our contributions provide a valuable framework for designing privacy-preserving algorithms under the shuffle model with personalized privacy guarantees.

Table 1. Comparative analysis of related studies: ✔indicates protected, while ✘ indicates unprotected.

Methods	Personalization	Adaptive process		Robustness
		Local	Central	
PLDP	✔	✔	Weak	✔
Uni-Shuffle	✘	✔	Medium	✘
APES	✔	✔	Medium	✘
E-APES	✔	✔	Strong	✔

In conclusion, our contributions can be summarized as follows:

1. We propose a novel approach that combines personalized local differential privacy (PLDP) with the shuffle model to achieve strong central privacy for the global model and personalized local privacy guarantees simultaneously. This approach addresses the challenge of balancing privacy and utility in a personalized setting.
2. We present a comprehensive analysis of the privacy and utility trade-offs in the proposed approach. We derive tighter bounds on the privacy guarantees compared to existing works, taking into account the varying local privacy requirements of users. Our analysis also considers the sensitivity to extreme values and provides robustness measures.
3. Furthermore, we have performed extensive comparative evaluations with existing methods, including the approach proposed by Liu et al. (2022). Our evaluations clearly show that our approach outperforms these existing methods in terms of achieving stronger central privacy, personalized local privacy, and preserving the utility of the global model. This comparative analysis

further highlights the superiority and practical significance of our proposed approach in the context of personalized local differential privacy under the shuffle model.

2 Preliminaries

In this section, we provide an overview of the privacy definition, shuffle model, and various properties of differential privacy. These concepts serve as the foundation for the proposed methods and play a crucial role in ensuring the privacy-preserving nature of our approach. We discuss the fundamental principles of privacy, the shuffle model as a powerful privacy mechanism, and explore important properties of differential privacy that guide the development of our methods. This comprehensive understanding sets the stage for the subsequent discussions and analyses in this paper.

2.1 Central and local differential privacy

Definition 1. *(**Differential privacy**) A randomized algorithm M is called (ϵ, δ)-differential privacy, denoted as (ϵ, δ)-DP, if for all $\mathcal{S} \subseteq Range(M)$ and for all neighboring databases D_0, D_1 (D_0 can be obtained form D_1 by replacing one record):*

$$\mathbb{P}(M(D_0) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}(M(D_1) \in \mathcal{S}) + \delta. \tag{1}$$

Definition 2. *(**Local differential privacy**) A randomized algorithm $\mathcal{R} : \mathcal{D} \to \mathcal{S}$ is called (ϵ, δ)-LDP if for all pairs $x, x' \in \mathcal{D}$, $\mathcal{R}(x)$ and $\mathcal{R}(x')$ satisfies*

$$\mathbb{P}(\mathcal{R}(x) \in \mathcal{S}) \leq e^{\epsilon}\mathbb{P}(\mathcal{R}(x') \in \mathcal{S}) + \delta. \tag{2}$$

In local differential privacy (LDP), each data contributor applies a local randomization mechanism to perturb their own data before sharing it with a central aggregator. The randomization introduces noise to the data, making it difficult to infer the exact contribution of any specific individual. By perturbing the data locally, LDP ensures that the privacy of each data contributor is protected even when the aggregated data is analyzed.

2.2 Privacy protection in the shuffle model

In the shuffle model, individuals' outputs from local randomizers are released through a trustworthy shuffler. This additional randomization, known as amplification by shuffling, has been shown to enhance privacy defenses against attackers who do not have access to the original local results. Extensive research has been devoted to the shuffle model's privacy boundary, with numerous articles (Erlingsson et al. 2019; Balle et al. 2019; Girgis et al. 2021; Feldman, McMillan and Talwar 2022) exploring this aspect. Among existing works, the best results in terms of differential privacy are provided by Feldman, McMillan and Talwar (2022). Liu et al. (2022) proposed privacy amplification frameworks via shuffle model for personalized private adaptive process (APES) and yielded a tighter privacy bound compared with unified privacy (see Figure 1). In this paper, we not only extend the local privacy parameter from ϵ_i to ϵ_i, δ_i, but also provide a more precise privacy bound. Therefore, we refer to this framework as enhanced APES (E-APES). This improvement allows for a finer control over the trade-off between privacy and utility in personalized private federated learning with the shuffle model.

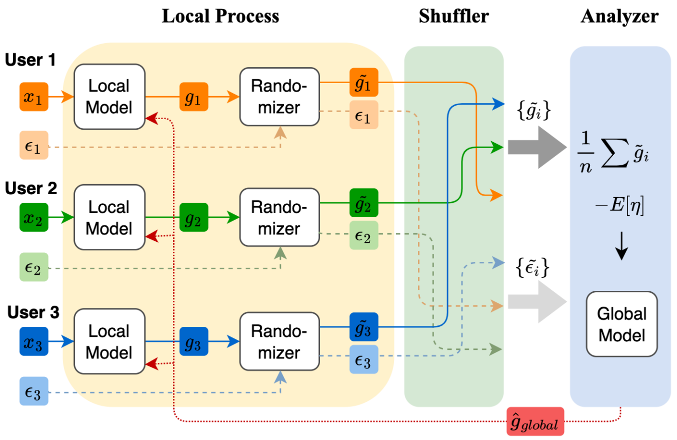

Fig. 1. Pure personalized LDP under the shuffle model. g_i is locally trained by user data x_i, then privacy parameters ϵ_i and g_i are shuffled separately.

2.3 Privacy tools

In the definition of differential privacy, the key is to measure the impact of individual data changes on the distribution of outputs. Therefore, it can be viewed as a hypothesis testing problem for a given distribution. Dong et al. (2022) proposed a powerful tool called f-DP based on hypothesis testing to handle this problem. To facilitate readability, we list the relevant properties of hypothesis testing. For two neighbouring databases D_0, D_1, let U and V denote the probability distributions of $M(D_0)$ and $M(D_1)$, respectively. We consider a rejection rule $0 \leq \phi \leq 1$, with type I and type II error rates defined as

$$\alpha_\phi = \mathbb{E}_U[\phi], \quad \beta_\phi = 1 - \mathbb{E}_V[\phi]. \tag{3}$$

It is well-known that $\alpha_\phi + \beta_\phi \geq 1 - TV(U, V)$, where $TV(U, V)$ is the supremum of $|U(A) - V(A)|$ over all measurable sets A. To characterize the fine-grained trade-off between the two errors, the following definition is necessary.
For any two probability distributions U and V on the same space Ω, the trade-off function $T(U, V) : [0, 1] \to [0, 1]$ is defined as

$$T(U, V)(\alpha) = \inf\{\beta_\phi : \alpha_\phi \leq \alpha\}, \tag{4}$$

where the infimum is taken over all measurable rejection rules ϕ, and $\alpha_\phi = \mathbb{E}_U(\phi)$ and $\beta_\phi = 1 - \mathbb{E}_V(\phi)$.

Definition 3. *(***Functional differential privacy***, f-DP) Let f be a trade-off function, a mechanism M is said to be f-differentially private if*

$$T(M(D_0), M(D_1)) \geq f, \tag{5}$$

for all neighboring data sets D_0 and D_1.

Fact 1 *(ϵ, δ)-DP is equivalent to $f_{\epsilon,\delta}$-DP, where*

$$f_{\epsilon,\delta} = \max\{0, 1 - \delta - e^{\epsilon}\alpha, e^{-\epsilon}(1 - \delta - \epsilon)\}. \tag{6}$$

Fact 2 *(μ-GDP) A f-DP mechanism is called μ-GDP if f can be obtained by $f = T(N(0, 1), N(\mu, 1)) = \Phi(\Phi^{-1}(1 - \alpha) - \mu)$. Then a mechanism is μ-GDP if and only if it is $(\epsilon, \delta(\epsilon))$-DP for all $\epsilon \geq 0$, where*

$$\delta(\epsilon) = \Phi(-\frac{\epsilon}{\mu} + \frac{\mu}{2}) - e^{\epsilon}\Phi(-\frac{\epsilon}{\mu} - \frac{\mu}{2}),$$

where $\Phi(\cdot)$ is cumulative distribution function of standard normal distribution $N(0,1)$. In particular, if a mechanism is μ-GDP, then it is $k\mu$-GDP for groups of size k. And the n-fold composition of μ_i-GDP mechanisms is $\sqrt{\mu_1^2+\cdots+\mu_n^2}$-GDP.

Fact 3 *f-DP holds the post-processing property, that is, if a mechanism M is f-DP, then its post-processing $Proc \circ M$ is also f-DP.*

The following two figures are taken from Dong et al. (2022), Figure 2 illustrates the relationship between f-DP and traditional DP from the perspective of hypothesis testing. It provides a visual representation of how the choice of parameter μ in μ-GDP relates to the strength of privacy protection. Figure 3 demonstrates that the bound provided by f-DP is nearly lossless and significantly outperforms the bounds given by traditional (ϵ,δ)-DP.

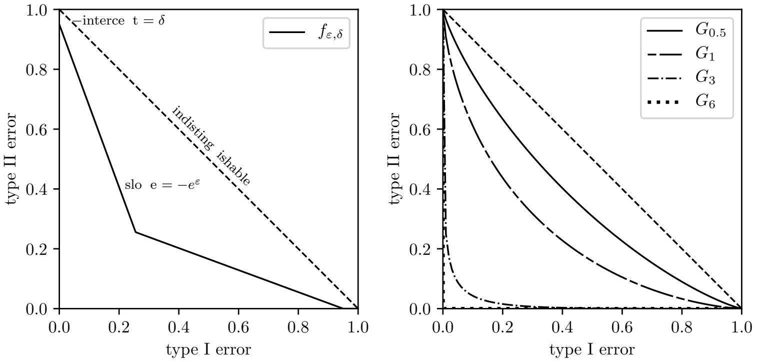

Fig. 2. The connection between traditional differential privacy (DP) and f-DP can be illustrated as follows. On the left, the function $f_{\epsilon,\delta}$ is a piecewise linear function that is symmetric about the line $y=x$. It has slopes of $-e^{\pm\epsilon}$ and intercepts of $1-\delta$. On the right, the trade-off functions of Gaussian distributions with unit variance and varying means are shown. The line $y=1-x$ represents the absence of privacy leakage.

3 E-APES framework

Definition 4. *For a domain $\mathcal{D}$, let $\mathcal{R}^{(i)} : \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \times \mathcal{S}^{(i-1)} \times \mathcal{D} \to \mathcal{S}^{(i)}$ for $i \in [n]$, where $\mathcal{S}^{(i)}$ is the range space of $\mathcal{R}^{(i)}$ be a sequence of algorithms such*

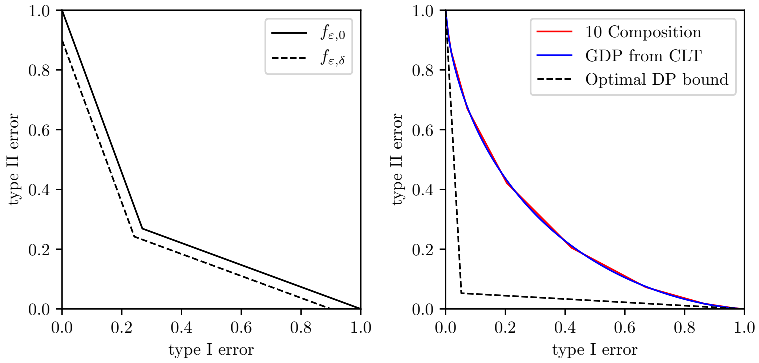

Fig. 3. Optimal privacy bounds based on f-DP and (ϵ, δ)-DP. Left: Applying tensoring with $f_{0,\delta}$ compresses the graph towards the origin, with a compression factor of $1 - \delta$. Right: The graph shows the result of a 10-fold composition of $(1/\sqrt{10}, 0)$-DP mechanisms. The dashed curve corresponds to "$\epsilon = 2.89; \delta = 0.001$". These values are determined by setting $\delta = 0.001$ and finding the smallest ϵ such that the composition is (ϵ, δ)-DP. It is worth noting that while the central limit theorem provides an almost perfect approximation of the true trade-off curve, the approximation achieved through (ϵ, δ)-DP is substantially looser.

that $\mathcal{R}^{(i)}(z_{1:i-1}, \cdot)$ is an (ϵ_i, δ_i)-DP local randomizer for all values of auxiliary inputs $z_{1:i-1} \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \mathcal{S}^{(i-1)}$. Let $\mathcal{A}_R : \mathcal{D} \to \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \times \mathcal{S}^{(n)}$ be the algorithm that given a dataset $x_{1:n} \in \mathcal{D}^n$, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_i)$ for $i \in [n]$ and outputs $z_{1:n}$. We say $\mathcal{A}_R(\mathcal{D})$ is an (ϵ_i, δ_i)-DP adaptive process. Similarly, if we first sample a permutation π uniformly at random, then sequentially computes $z_i = \mathcal{R}^{(i)}(z_{1:i-1}, x_{\pi_i})$ for $i \in [n]$ and outputs $z_{1:n}$, we say this process is shuffled (ϵ_0, δ_0)-DP adaptive and denote it by $\mathcal{A}_{R,S}(\mathcal{D})$.

The (ϵ_i, δ_i)-differentially private mechanism can be dominated by the following hypothesis testing problem.

Lemma 1 (KOV15). *Let $\mathcal{R}^{(i)} : \mathcal{D} \to \mathcal{S}$ be an (ϵ_i, δ_i)-DP local randomizer, and $x_0, x_1 \in D$, then there exists two quaternary random variables $\tilde{X}_0$ and $\tilde{X}_1$, such that $\mathcal{R}^{(i)}(x_0)$ and $\mathcal{R}^{(i)}(x_1)$ can be viewed as post-processing of $\tilde{X}_0$ and $\tilde{X}_1$,* *respectively. In details,*

$$P(\tilde{X}_0 = x) = \begin{cases} \delta_i & \text{if } x = A, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \text{if } x = 0, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \text{if } x = 1, \\ 0 & \text{if } x = B, \end{cases}$$

and

$$P(\tilde{X}_1 = x) = \begin{cases} 0 & \text{if } x = A, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \text{if } x = 0, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \text{if } x = 1, \\ \delta_i & \text{if } x = B. \end{cases}$$

Lemma 2. *Given an (ϵ_i, δ_i)-DP adaptive process, then in the i-th step, local randomizer $\mathcal{R}^{(i)}: \mathcal{D} \rightarrow \mathcal{S}$ and for any $n+1$ inputs $x_1^0, x_1^1, x_2, \cdots, x_n \in \mathcal{D}$, there exists distributions $\mathcal{Q}_1^0, \mathcal{Q}_1^1, \mathcal{Q}_1, \mathcal{Q}_2, \cdots, \mathcal{Q}_n$ such that*

$$\mathcal{R}^{(i)}(x_1^0) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{Q}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^1 + \delta_i \mathcal{Q}_1, \tag{7}$$

$$\mathcal{R}^{(i)}(x_1^1) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{Q}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^1 + \delta_i \mathcal{Q}_1. \tag{8}$$

$$\forall x_i \in \{x_2, \cdots, x_n\}, \mathcal{R}(x_i) = \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{Q}_1^1 + (1 - \frac{2(1-\delta_i)}{1+e^{\epsilon_i}})\mathcal{Q}^i \tag{9}$$

Proof. For inputs $X_0 = \{x_1^0, x_2, \ldots, x_n\}$ and $X_1 = \{x_1^1, x_2, \ldots, x_n\}$, $\mathcal{R}^{(i)}$ satisfies the constraints of Lemma 1, so there exists an (ϵ_i, δ_i)-DP local randomizer $\mathcal{R}' : \mathcal{D} \rightarrow \mathcal{Z}$ for the i-th output, and post-processing function $proc(\cdot)$ such that $proc(\mathcal{R'}^{(i)}(x)) = \mathcal{R}^{(i)}(x)$, and

$$P(\mathcal{R'}^{(i)}(x_1^0) = z) = \begin{cases} 0 & \text{if } z = A, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \text{if } z = 0, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \text{if } z = 1, \\ \delta_i & \text{if } z = B. \end{cases}$$

$$P(\mathcal{R'}^{(i)}(x_1^1) = z) = \begin{cases} \delta_i & \text{if } z = A, \\ \frac{1-\delta_i}{1+e^{\epsilon_i}} & \text{if } z = 0, \\ \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}} & \text{if } z = 1, \\ 0 & \text{if } z = B. \end{cases}$$

Let $L = \{z \in \mathcal{Z} | \mathbb{P}(\mathcal{R}'(x_1^0 = z)) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}$ and $\mathbb{P}(\mathcal{R}'(x_1^1 = z)) = \frac{1-\delta_i}{1+e^{\epsilon_i}}\}$, $U = \{z \in \mathcal{Z} | \mathbb{P}(\mathcal{R}'(x_1^1 = z)) = \frac{1-\delta_i}{1+e^{\epsilon_i}}$ and $\mathbb{P}(\mathcal{R}'(x_1^1 = z)) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\}$. Let $M = \mathcal{Z}\{L \bigcup U\}$ and $p = \sum_{z \in \mathcal{L}} p_z = \sum_{z \in \mathcal{U}} p_z$. Since conditioned on the output lying in $\mathcal{L}$, the distribution of $\mathcal{R}'(x_1^0)$ and $\mathcal{R}'(x_1^1)$ are same. Let $\mathcal{W}_1^0 = \mathcal{R}'(x_1^0)|L = \mathcal{R}'(x_1^1)|L$, $\mathcal{W}_1^1 = \mathcal{R}'(x_1^0)|U = \mathcal{R}'(x_1^1)|U$ and $\mathcal{W}_1 = \mathcal{R}'(x_1^0)|M = \mathcal{R}'(x_1^1)|M$. Then

$$\mathcal{R}'(x_1^0) = \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{W}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{W}_1^1 + \delta_i \mathcal{W}_1,$$

$$\mathcal{R}'(x_1^1) = \frac{(1-\delta_i)}{1+e^{\epsilon_i}}\mathcal{W}_1^0 + \frac{(1-\delta_i)e^{\epsilon_i}}{1+e^{\epsilon_i}}\mathcal{W}_1^1 + \delta_i \mathcal{W}_1.$$

Further, for all $x_i \in \{x_2, \cdots, x_n\}$,

$$\mathcal{R}'(x_i) \geq \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{W}_1^0 + \frac{1-\delta_i}{1+e^{\epsilon_i}}\mathcal{W}_1^1 + (1 - \frac{2(1-\delta_i)}{1+e^{\epsilon_i}})\mathcal{W}_i.$$

Letting $\mathcal{Q}_1^0 = proc(\mathcal{W}_1^0)$, $\mathcal{Q}_1^1 = proc(\mathcal{W}_1^1)$, $\mathcal{Q}_1 = proc(\mathcal{W}_1)$ and for all $i \in \{2, \cdots, n\}$, $\mathcal{Q}_i = proc(\mathcal{W}_i)$. The proof is completed.

Theorem 1. *For a domain D, if $\mathcal{A}_R(\mathcal{D})$ is an shuffled (ϵ_i, δ_i)-DP adaptive process, then there exists a post-processing function $proc(\cdot)$: $(0, 1, 2) \rightarrow \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \cdots \times \mathcal{S}^{(n)}$, such that*

$$T(\mathcal{A}_{R,S}(X_0), \mathcal{A}_{R,S}(X_1)) = T(proc(P_0), proc(P_1)).$$

Here,

$$P_0 = (\varDelta_0, \varDelta_1, \varDelta_2) + \sum_{i=1}^{n-1} MultiBern(\frac{(1-\delta_i)}{1+e^{\epsilon_i}}, \frac{1-\delta_i}{1+e^{\epsilon_i}}, \delta_i), \tag{10}$$

$$P_1 = (\varDelta_1, \varDelta_0, \varDelta_2) + \sum_{i=1}^{n-1} MultiBern(\frac{(1-\delta_i)}{1+e^{\epsilon_i}}, \frac{1-\delta_i}{1+e^{\epsilon_i}}, \delta_i), \tag{11}$$

$\varDelta_2 \sim Bern(\delta_{\max})$, $\varDelta_0 \sim Bin(1-\varDelta_2, \frac{e^{\epsilon_i}}{1+e^{\epsilon_i}})$, $\varDelta_1 = 1-\varDelta_0-\varDelta_2$ and $MultiBern(\theta_1, \cdots, \theta_d)$ represents a d-dimensional Bernoulli distribution with $\sum_{j=1}^{d} \theta_j = 1$.

Proof. Formally, for each $i \in \{2, \cdots, n\}$, let $p_i = \frac{1-\delta_i}{1+e^{\epsilon_i}}$, we define random variables $Y_{1,i}^0$, $Y_{1,i}^1$ and Y_i as follows:

$$Y_{1,i}^0 = \begin{cases} 0 & w.p. \quad e^{\epsilon_i}\frac{p_i}{2}, \\ 1 & w.p. \quad \frac{p_i}{2}, \\ 2 & w.p. \quad 1 - e^{\epsilon_i}\frac{p_i}{2} - \frac{p_i}{2}. \end{cases} \qquad Y_{1,i}^1 = \begin{cases} 0 & w.p. \quad \frac{p_i}{2}, \\ 1 & w.p. \quad e^{\epsilon_i}\frac{p_i}{2}, \\ 2 & w.p. \quad 1 - e^{\epsilon_i}\frac{p_i}{2} - \frac{p_i}{2}. \end{cases} \tag{12}$$

and

$$Y_i = \begin{cases} 0 & w.p. \quad \frac{p_i}{2}, \\ 1 & w.p. \quad \frac{p_i}{2}, \\ 2 & w.p. \quad 1 - p_i. \end{cases} \tag{13}$$

We consider the case in the t-th iteration. Given a dataset X_b for $b \in \{0,1\}$, we generate n samples from $\{0,1,2\}$ in the following way. Client number one reports a sample from $Y^b_{1,i}$. Clients $i(i = 2, \cdots, n)$ each reports an independent sample from Y_i. We then shuffle the reports randomly. Let ρ_b denote the resulting distribution over $\{0,1,2\}^n$. We then count the total number of 0s and 1s. Note that a vector containing a permutation of the users responses contains no more information than simply the number of 0s and 1s, so we can consider these two representations as equivalent.

We claim that there exists a post-processing function $proc(\cdot)$ such that for y sampled from ρ_b, $proc(y)$ is distributed identically to $\mathcal{A}_S(X_b)$. To see this, let π be a randomly and uniformly chosen permutation of $\{1, \cdots, n\}$. For every $i \in \{1, \cdots, n\}$, given the hidden permutation π, we can generate a sample from $\mathcal{A}_S(X_b)$ by sequentially transforming $proc(y_t)$ to obtain the correct mixture components, then sampling from the corresponding mixture component. Specially, by Lemma 2,

$$z_t = \begin{cases} \mathcal{R}^{(t)}(z_{1:t-1}, x_1^0) & \text{if } y_t = 0; \\ \mathcal{R}^{(t)}(z_{1:t-1}, x_1^1) & \text{if } y_t = 1; \\ \mathcal{R}^{(t)}(z_{1:t-1}, x_{\pi(i)}) & \text{if } y_t = 2. \end{cases} \tag{14}$$

By our assumption, this produces a sample z_t from $\mathcal{R}^{(i)}(x_{\pi(i)})$. It is easy to see that the resulting random variable (z, y) has the property that for input $b \in \{0,1\}$, it marginal distribution over $\mathcal{S}$ is the same as $\mathcal{A}_S(X_b)$ and marginal distribution over $\{0,1,2\}^n$ is ρ_b. The difficulty then lies in the fact that conditioned on a particular instantiation $y = v$, the permutation $\pi|_{y=v}$ is not independent of b. Note that if $v_t = 0$ or 1, the corresponding $\mathcal{Q}_1^{0(t)}(z_{1:t-1})$ or $Q_1^{1(t)}(z_{1:t-1})$, is independent of π. Therefore, in order to do the appropriate post-processing, it suffices to know the permutation π restricted to the set of users who sampled 2, $K = \pi(\{i : y_i = 2\})$. The set K of users who select 2 is independent of b since $Y^0_{1,i}$ and $Y^1_{1,i}$ have the same probability of sampling 2. The probability of being included in K is identical for each $i \in \{2, \cdots, n\}$, and slightly smaller for the first user. Since the sampling of z given y only needs K, we can sample from $z|_{(y,K)=(v,J)}$ without knowing b. This conditional sampling is exactly the post-processing step that we claimed.

We now analyze the divergence between P_0 and P_1, the shuffling step implies that P_0 and P_1 are symmetric. This implies that the divergence between P_0 and P_1 is equal to the divergence between the distribution between the distribution of the counts of $0'$s and $1'$s. The decomposition in equation (9) implies that the divergence between $\mathcal{A}_S(X_0)$ and $\mathcal{A}_S(X_1)$ can be dominated by the divergence of P_0 and P_1, where $\Delta_2 \sim Bern(\delta_{\max})$, $\Delta_0 \sim Bin(1-\Delta_2, \frac{e^{\epsilon_i}}{1+e^{\epsilon_i}})$, $\Delta_1 = 1-\Delta_0-\Delta_2$ and $MultiBern(\theta_1, \cdots, \theta_d)$ represents a d-dimensional Bernoulli distribution with $\sum_{j=1}^{d} \theta_j = 1$.

Lemma 3. *(Berry Esseen) Let $P = (\xi_0, \xi_1, \xi_2) \sim \sum_{i=1}^{m} MultiBern(\frac{p_i}{2}, \frac{p_i}{2}, 1-p_i)$ and $Q \sim N(\mu, \Sigma)$, where $\mu = \mathbb{E}(P)$ and $\Sigma = Var(P)$. Then for the first two components (X_0, X_1), there exists $C > 0$, such that $\|\tilde{P} - \tilde{Q}\|_{TV} \leq \frac{C}{\sqrt{m}}$, where $\tilde{P}$ and $\tilde{Q}$ represent the distribution of (ξ_0, ξ_1) and corresponding normal distribution, respectively.*

In fact, for given n, we can obtain sophisticated bound of $\|\tilde{P}-\tilde{Q}\|_{TV}$ by numerical methods. Without loss of generality, we assume $\epsilon_i = \epsilon_0, \delta_i = \delta_0 = O(1/n)$, then $p_0 = \frac{1-\delta_0}{1+e^{\epsilon_0}}$. For some fixed output $(\xi_0, \xi_1) = (k_0, k_1)$, we approximate by integrating the normal probability density function around that point. Let $G(\cdot)$ be the cumulative distribution function of $\tilde{Q}$ and $h(k_0, k_1) = G(k_0 + 0.5, k_1 + 0.5) - G(k_0 + 0.5, k_1) - G(k_0, k_1 + 0.5) + G(k_0, k_1)$, then

$$\|\tilde{P} - \tilde{Q}\|_{TV} = \sup_{(k_0,k_1)} |\mathbb{P}(\xi_0 = k_0, \xi_1 = k_1) - h(k_0, k_1)|. \tag{15}$$

For example, if we take $\epsilon_i = \epsilon_0 = 0.1, \delta_i = \delta_0 = 1/n$, numerical method gives $\|\tilde{P} - \tilde{Q}\|_{TV} = 0.0206$, which is nearly $O(1/n)$.

It is famous that for type I error and type II error between P and Q, we have

$$\alpha + \beta \geq 1 - \|P - Q\|_{TV}. \tag{16}$$

The next step is to focus on the privacy preservation of normal distribution.

Theorem 2. *Let $p_i = \frac{2(1-\delta_i)}{1+e^{\epsilon_i}}$, if $\bar{\mu} = \sum_{i=1}^{n-1} (\frac{p_i}{2}, \frac{p_i}{2})'$ and $\mu_0 = (1, 0)' + \bar{\mu}$, $\mu_1 = (0, 1)' + \bar{\mu}$, then $T(N(\mu_0, \Sigma), N(\mu_1, \Sigma)) = \Phi(\Phi^{-1}(1-\alpha) - \frac{2}{\sqrt{\sum_{i=1}^{n-1} p_i}})$, where*

$$\boldsymbol{\Sigma} = \sum_{i=1}^{n-1} \begin{pmatrix} \frac{p_i}{2}(1-\frac{p_i}{2}) & -\frac{p_i^2}{4} \\ -\frac{p_i^2}{4} & \frac{p_i}{2}(1-\frac{p_i}{2}) \end{pmatrix}.$$

Proof. Since

$$T(N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}), N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})) = \Phi(\Phi^{-1}(1-\alpha) - \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)}),$$

according to the property of normal distribution, the key is to calculate $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$. Let $v_1 = \sum_{i=1}^{n-1} p_i, v_2 = \sum_{i=1}^{n-1} p_i^2$, then

$$\boldsymbol{\Sigma} = \begin{pmatrix} \frac{v_1}{2} - \frac{v_2}{4} & -\frac{v_2}{4} \\ -\frac{v_2}{4} & \frac{v_1}{2} - \frac{v_2}{4} \end{pmatrix},$$

and

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \frac{2v_1 - v_2}{v_1^2 - v_1 v_2} & \frac{v_2}{v_1^2 - v_1 v_2} \\ \frac{v_2}{v_1^2 - v_1 v_2} & \frac{2v_1 - v_2}{v_1^2 - v_1 v_2} \end{pmatrix}.$$

By simple calculation, we can obtain that

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = (-1,1)\boldsymbol{\Sigma}^{-1}(-1,1)' = \frac{4}{\sum_{i=1}^{n-1} p_i}.$$

Substituting $\mu = \sqrt{\frac{4}{\sum_{i=1}^{n-1} p_i}}$ yields the proof.

Lemma 4 (DRS22). *Suppose $T(P,R) \geq f, T(Q,R) \geq g$, then $T(P,R) \geq f \circ g = g(1 - f(\alpha))$.*

Theorem 3. *Assume ρ_0 and ρ_1 are defined in equation (10) and (11), then there exists $C > 0$, such that $T(\rho_0, \rho_1) \geq \delta_{\max}(1-\alpha) + (1-\delta_{\max})(G_\mu(\alpha + \frac{C}{\sqrt{n-1}}) - \frac{C}{\sqrt{n-1}})$, where $G_\mu(\alpha) = \Phi(\Phi^{-1}(1-\alpha) - \mu), \mu = \sqrt{\frac{2}{\sum_{i=1}^{n-1} \frac{1-\delta_i}{1+e^{\epsilon_i}}}}$.*

Proof. According to the definition of $(\Delta_0, \Delta_1, \Delta_2)$,

$$(\Delta_0, \Delta_1, \Delta_2) = \begin{cases} (0,0,1) & w.p. \quad \delta_{\max}; \\ (1,0,0) & w.p. \quad \frac{(1-\delta_{\max})e^{\epsilon_{\max}}}{1+e^{\epsilon_{\max}}}; \\ (0,1,0) & w.p. \quad \frac{(1-\delta_{\max})}{1+e^{\epsilon_{\max}}}. \end{cases} \tag{17}$$

When $\Delta_2 = 1$, ρ_0 and ρ_1 are indistinguishable, which indicates that $T(\rho_0, \rho_1)|_{\Delta_2 = 1} = 1 - \alpha$. Assume $P \sim N(\mu_0, \Sigma)$, $Q \sim N(\mu_1, \Sigma)$, where μ_0, μ_1, Σ are same as that in Theorem 2, then

$$T(\rho_0, \rho_1) = \frac{(1-\delta_{\max})e^{\epsilon_{\max}}}{1+e^{\epsilon_{\max}}} T(P,Q) + \frac{(1-\delta_{\max})}{1+e^{\epsilon_{\max}}} T(Q,P) + \delta_{\max}(1-\alpha). \tag{18}$$

Due to the symmetry of normal distribution, we can obtain that $T(P,Q) = T(Q,P)$. Let $\rho_0' = (1,0,0)' + \sum_{i=1}^{n-1} MultiBern(p_i/2, p_i/2, 1-p_i)$ and $\rho_1' = (0,1,0)' + \sum_{i=1}^{n-1} MultiBern(p_i/2, p_i/2, 1-p_i)$, combined with Theorem 2,

$$T(\rho_0,\rho_1) = \delta_{\max}(1-\alpha) + (1-\delta_{\max})T(\rho_0',\rho_1'). \tag{19}$$

Let $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n-1}\frac{1-\delta_i}{1+e^{\epsilon}i+1}}}$, according to equation (16),

$$T(\rho_0',P) \geq 1-\alpha-\|\rho_0'-P\|_{TV},$$

$$T(\rho_1',Q) \geq 1-\alpha-\|\rho_1'-Q\|_{TV},$$

then based on Lemma 4,

$$T(\rho_0',Q) \geq \Phi(\Phi^{-1}(1-\alpha-\|\rho_0'-P\|_{TV})-\mu) = F(\alpha).$$

Reuse Lemma Lemma 4, we can obtain that

$$T(\rho_0',\rho_1') \geq 1-(1-F(\alpha))-\|\rho_1'-Q\|_{TV} = \Phi(\Phi^{-1}(1-\alpha-\|\rho_0'-P\|_{TV})-\mu)-\|\rho_1'-Q\|_{TV}.$$

Lemma 3 shows that there exists $C > 0$, such that $\|\rho_1'-Q\|_{TV} \leq \frac{C}{\sqrt{n-1}}$ and $\|\rho_0'-P\|_{TV} \leq \frac{C}{\sqrt{n-1}}$. Hence

$$T(\rho_0',\rho_1') \geq G_\mu(\alpha+\frac{C}{\sqrt{n-1}})-\frac{C}{\sqrt{n-1}}.$$

Then

$$T(\rho_0,\rho_1) \geq \delta_{\max}(1-\alpha)+(1-\delta_{\max})(G_\mu(\alpha+\frac{C}{\sqrt{n-1}})-\frac{C}{\sqrt{n-1}}).$$

4 Application and experiments

4.1 Privacy analysis of personalized private stochastic gradient descent in the shuffle model

The personalized private stochastic gradient descent (personalized-SGD) is a method that combines personalized differential privacy with stochastic gradient descent optimization for model training and parameter updates while ensuring privacy protection.

Traditional gradient descent algorithms typically use raw training data for optimizing model parameters. However, in the context of personalized differential privacy, privacy of individual users must be protected, and direct use of raw data for parameter updates is not feasible.

The key idea of personalized differential privacy is to introduce personalized parameters into the differentially private mechanism to flexibly adjust the level of privacy protection. For the gradient descent algorithm, personalized differential privacy can be achieved by introducing noise during gradient computation.

Algorithm 1 Shuffled noisy SGD for (ϵ_0, δ_0)-LDP adaptive process

```
Input: Dataset $X = (x_1, \dots, x_n)$, loss function $\mathcal{L}(\boldsymbol{\theta}, x)$, initial point $\boldsymbol{\theta}_0$, learning rate $\eta$, number of epochs $T$, privacy budget $\mathcal{S} = \{\epsilon_1, \delta_1, \cdots, \epsilon_n, \delta_n\}$, batch size $m$ and gradient norm bound $C$.
Output: $\hat{\boldsymbol{\theta}}_{0,m}$
Split $[n]$ into $m$ disjoint subsets $S_1, \cdots, S_m$ with equal size $n/m$
Choose arbitrary initial point $\hat{\boldsymbol{\theta}}_{0,m}$
Choose a random permutation $\pi$: $[m] \rightarrow [m]$
for each $t \in [T]$ do
    $\tilde{\boldsymbol{\theta}}_0 = \hat{\boldsymbol{\theta}}_{0,m}$
    for each $i \in [m]$ do
        $\sigma = \frac{2C}{m} \frac{\sqrt{2\log(1.25/\delta_{\pi(i)})}}{\epsilon_{\pi(i)}}$
        $\boldsymbol{b}_i \sim N(0, \sigma^2 \boldsymbol{I}_d)$
        for each $j \in S_{\pi(i)}$ do
            Compute gradient:
            $\boldsymbol{g}_i^j = \nabla \ell(\tilde{\boldsymbol{\theta}}_{i-1}, x_j)$
            Clip to norm $C$:
            $\tilde{\boldsymbol{g}}_i^j = \boldsymbol{g}_i^j / \max(1, \|\boldsymbol{g}_i^j\|_2 / C)$
        end for
        $\tilde{\boldsymbol{\theta}}_i = \tilde{\boldsymbol{\theta}}_{i-1} - \eta(\frac{1}{m}\sum_j \tilde{\boldsymbol{g}}_i^j + \boldsymbol{b}_i)$
    end for
    $\hat{\boldsymbol{\theta}}_{t,m} = \tilde{\boldsymbol{\theta}}_m$
end for
return $\hat{\boldsymbol{\theta}}_{T,m}$
```

Theorem 4. *Algorithm 1 approximately satisfies ss-GDP with probability.*

Dataset and implementation The MNIST dataset (LeCun et al. 1998) for handwritten digit recognition consists of $60,000$ training images and $10,000$ test images. Each sample in the dataset represents a 28×28 vector generated from handwritten images, where the independent variable corresponds to the input vector, and the dependent variable represents the digit label ranging from 0 to 9. In our experiments, we consider a scenario with m clients, where each client has n/m samples. For simplicity, we train a simple classifier using a feed-forward neural network with ReLU activation units and a softmax output layer with 10 classes, corresponding to the 10 possible digits. The model is trained using cross-entropy loss and an initial PCA input layer with 60 components. At each step of the shuffled SGD, we choose at one client at random without replacement. The parameters of experimental setup is listed in Table 2.

Baselines The EoN method proposed by Liu et al. (2023) demonstrates significant superiority in both theoretical analysis and numerical results on privacy amplification compared with existing work [18–21]. To the best of our knowledge, EoN method is the first to analyze personalized differential privacy in the shuffle model. Therefore, we consider it as the baseline for comparison.

Parameter Selection As a result, our approach achieves an accuracy of 96.78% on the test dataset after approximately 50 epochs. This result is consistent with the findings of a vanilla neural network [22] trained on the same MNIST dataset. By employing this methodology, we can effectively train a simple classifier that achieves high accuracy in recognizing handwritten digits from the MNIST dataset.

Table 2. Experiment setting for the shuffled SGD on the MNIST dataset

Parameters/Setting	**Value**	**Explanation**
C	2	Clipping bound
δ_0	10^{-5}	With prob. $1 - \delta_0$ satisfies DP
ϵ_0	$[0.05, 1]$	Privacy budget
η	0.05	Step size of the gradient
m	100	The number of clients
n	$60,000$	Total number of training samples
T	50	The number of epochs

Privacy Amplification Effect We provide numerical evaluations for privacy amplification effect under fixed personalized LDP settings in Table 4.1. Given local privacy budget $\epsilon^\ell \in [0.01, 2]$, we set δ for shuffling to $\frac{1}{n}$. For the purpose of comparison, we examine privacy amplification for a fixed ϵ^ℓ while varying n from 10^3 to 10^6. To avoid misunderstandings, we repeat the first 10^3 parameters. Considering that γ in Lemma 3 is $O(1/n)$ and can be negligible in numerical analysis, our focus lies in measuring G_μ.

To keep it concise, we will use the abbreviation [LZXCL23] to refer to the work by Liu et al. (2023). The numerical results demonstrate the following results: (i) Our bound is robust for extreme values, as one budget should not significantly impact others, while each ϵ_i should not be close to zero to prevent loose bounds in [LZXCL23]. However, it is natural to encounter user responses that contain no information, resulting in $\epsilon_i = 0$. (ii) As the sample size n increases, the amplification effect also increases proportionally to the square root of n. (iii) Our privacy bounds significantly outperform [LZXCL23] in all current scenarios eliminate the constraints on ϵ^ℓ. (iv) The privacy bound of Uniform 2 and Gauss 2 are nearly same, which indicates that the distribution is not the primary factor influencing the bounds.

Table 3. Distributions of personalized LDP budgets ϵ^ℓ. $U(a, b)$ represents uniform distribution ranging from a to b, $N(\mu, \sigma^2)$ represents normal distribution with mean μ and standard deviation σ. The clip range $[a, b]$ indicates that values outside the range $[a, b]$ will be replaced with the nearest boundary value a or b.

Name	Distribution of $\epsilon^\ell = (\epsilon_1^\ell, \epsilon_2^\ell, \cdots, \epsilon_n^\ell)$	Clip range
Uniform 1	$U(0.05, 1)$	[0.05,1]
Uniform 2	$U(0.01, 2)$	[0.01,2]
Gauss 1	$N(0.5, 0.2)$	[0.05,1]
Gauss 2	$N(1, 0.2)$	[0.01,2]
Gauss 3	$N(1, 0.5)$	[0.01,2]

Utility of E-APES We evaluate the utility of E-APES with ϵ^ℓ drawing from Table 4.1.

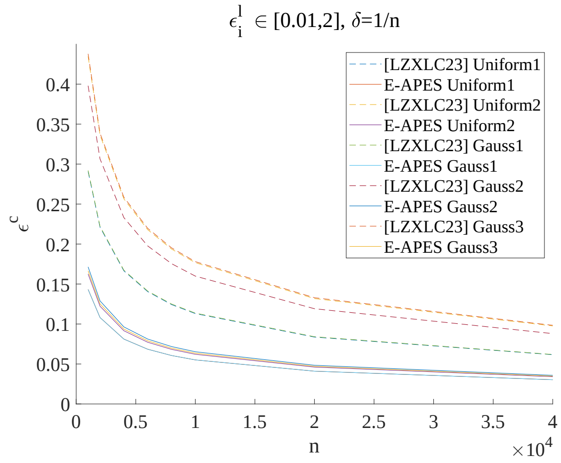

Fig. 4. Privacy bound of personalized LDP for different set of budgets

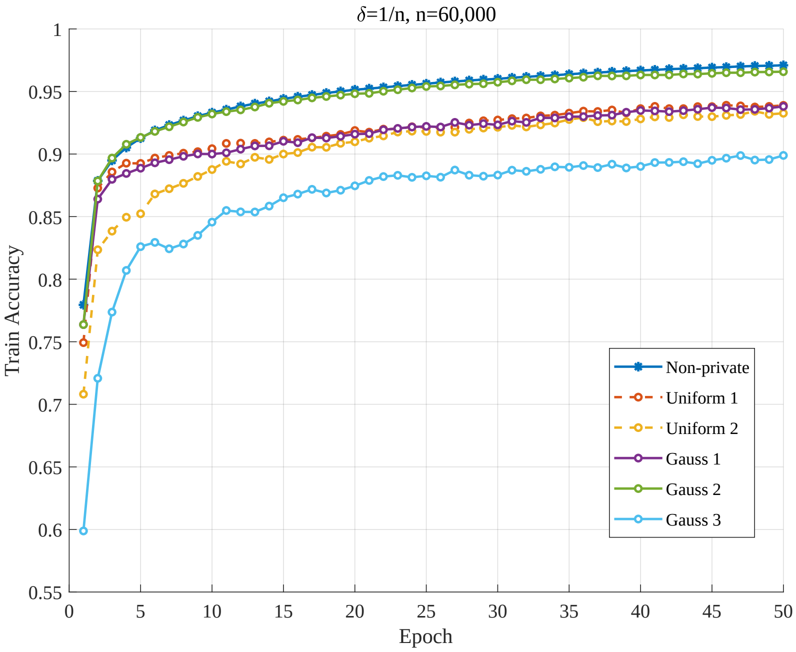

Fig. 5. Comparison of Training Set Accuracy with Different Noise Distributions

4.2 Application to average and count functions

Average function In this section, we apply E-APES to the average function on the synthetic data. According to [16], we randomly divide the users into three groups: conservative, moderate, and liberal. The fraction of three groups are determined by f_c, f_m, f_l. As is reported in [17], the default values in this experiment are $f_c = 0.54, f_m = 0.37, f_l = 0.09$. For convenience, the privacy preferences for the users in conservative, moderate and liberal groups are ϵ_C, ϵ_M and ϵ_l, respectively. In the LDP case, the privacy preference of users in the liberal group is fixed at $\epsilon_L = 1$, while the default values of ϵ_C and ϵ_M are set to 0.1 and 0.5, respectively. However, due to the presence of a shuffler, the privacy of users can be further enhanced, allowing for smaller noise to be added in LDP. According to the definition of local differential privacy and Theorem 3, we can obtain privacy bound of Algorithm 2.

Corollary 1. *Algorithm 2 approximately preserves μ-GDP for each user, where* $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n-1} \frac{1}{1+e^{\epsilon_i}}}}$.

Next, we simulate the accuracy for different set of privacy protection. To facilitate comparison, we set $f_l = 0.09$ as a fixed value and vary f_c from 0.01 to 0.5 with $f_m = 1 - f_l - f_c$. Additionally, we generate $n = 10,000$ privacy budgets for users based on the privacy preferences rule. We assume that each sample is drawn from a normal distribution $N(50, \sigma^2)$, and then the samples are clipped into the range $[20, 80]$. We repeat this procedure for a total of $1,000$ times to give a confidence interval. According to Theorem 1, privacy parameter μ under the shuffle model can be obtained for varying ϵ_c. Figure 7 shows that

Algorithm 2 Shuffled noisy average satisfies personalized-LDP

```
Input: Dataset $X = (x_1, \dots, x_n) \in \mathbb{R}^n$, privacy budget $\mathcal{S} = \{\epsilon_1, \cdots, \epsilon_n\}$ for each user.
Output: $z \in \mathbb{N}$
for each $i \in [n]$ do
    $y_i \leftarrow x_i + Lap(\Delta f / \epsilon_i)$
end for
Choose a random permutation $\pi$: $[n] \rightarrow [n]$
$z = \frac{1}{n} \sum_{i=1}^{n} y_{\pi(i)}$
return $z$
```

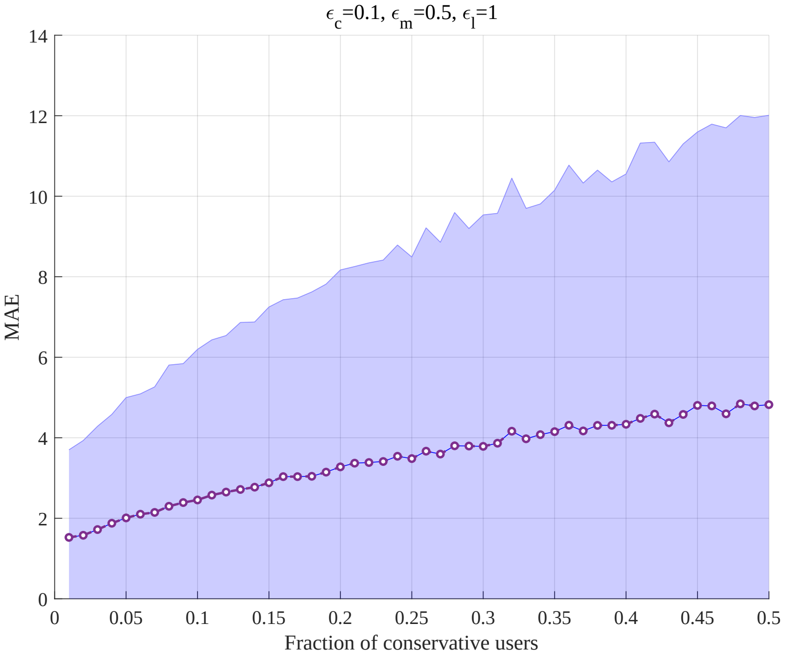

Fig. 6. Impact of f_c

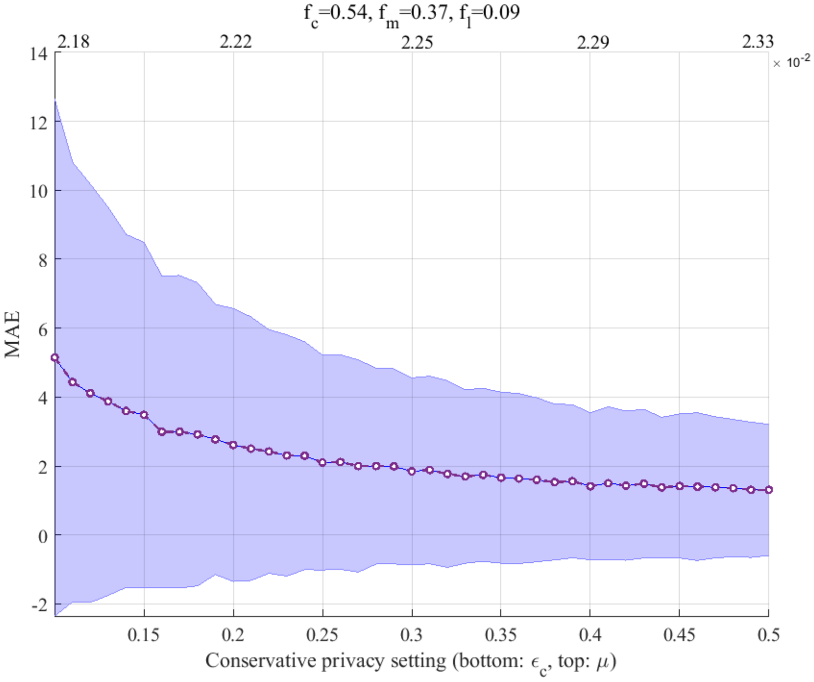

Fig. 7. Impact of ϵ_c (with $\epsilon_m = 0.5$)

Frequency estimation In order to obtain the dataset, a total of 10,000 records are generated for counting. Each record is encoded as a binary attribute. The proportion of records with a value of 1 is determined by a density parameter c, which ranges from 0 to 1 (with a default value of $c = 0.3$).

Corollary 2. *Algorithm 3 approximately preserves μ-GDP for each user, where* $\mu = \sqrt{\frac{2}{\sum_{i=1}^{n-1} \frac{1}{1+e^{\epsilon_i}}}}$.

The proof of Corollary 2 is the as Corollary 1. The direct calculation shows that z is an unbiased estimator of c, that is, $\mathbb{E}(z) = c$. Similar with average function, we take the same setting of personalized privacy budgets.

Algorithm 3 Shuffled personalized-LDP frequency estimation

```
Input: Dataset $X = (x_1, \dots, x_n) \in \{0,1\}^n$, privacy budget $\mathcal{S} = \{\epsilon_1, \cdots, \epsilon_n\}$ for each user.
Output: $z \in \mathbb{N}$
for each $i \in [n]$ do
    if $x_i = 1$ then
        $y_i \leftarrow Ber(\frac{e^{\epsilon_i}}{1+e^{\epsilon_i}})$
    else
        $y_i \leftarrow Ber(\frac{1}{1+e^{\epsilon_i}})$
    end if
end for
Choose a random permutation $\pi$: $[n] \to [n]$
$A = \sum_{i=1}^{n} y_{\pi(i)}$
$B = \sum_{i=1}^{n} \frac{1}{1+e^{\epsilon_{\pi(i)}}}$
$z = \frac{A-B}{n-2B}$
return $z$
```

References

1. Balle B, Barthe G, Gaboardi M. (2018). *Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences.* In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 6280-6290.
2. Bittau A, Erlingsson Ú, Maniatis P, Mironov I, Raghunathan A, Lie D, Rudominer M, Kode U, Tinnes J, Seefeld B (2017). *Prochlo: Strong Privacy for Analytics*

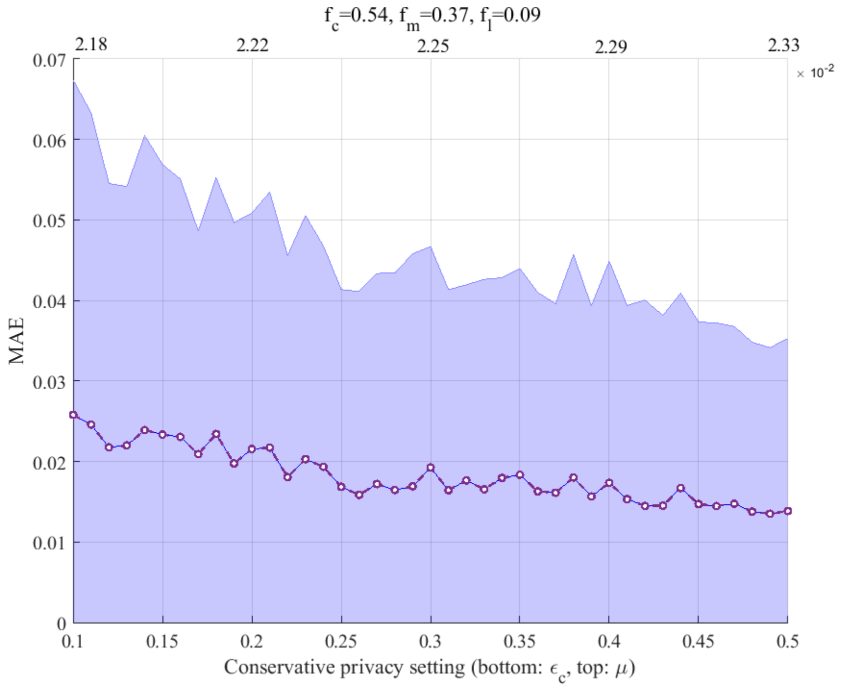
f_c=0.54, f_m=0.37, f_l=0.09
2.18
2.22
2.25
2.29
2.33
× 10^-2
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
MAE
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
Conservative privacy setting (bottom: ϵ_c, top: μ)

Fig. 8.

in the Crowd. In: Proceedings of the 26th Symposium on Operating Systems Principles, 441-459.

3. Chen R, Li H, Qin A K, Kasiviswanathan S P, Jin H (2016). *Private spatial data aggregation in the local setting.* In: IEEE 32nd International Conference on Data Engineering, 289-300.
4. Cheu A, Smith A, Ullman J, Zeber D, Zhilyaev M (2019). *Distributed differential privacy via shuffling.* Advances in Cryptology-EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 1(38): 375-403.
5. Erlingsson *Ú*, Feldman V, Mironov I, Raghunathan A, Talwar K, Thakurta A (2019). *Amplification by shuffling: from local to central differential privacy via anonymity.* In: Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, 2468-2479.
6. Feldman V, McMillan A and Talwar K (2022). *Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling.* In: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS), 954 – 964.
7. Girgis A M, Data D, Diggavi S, Suresh A T, Kairouz P (2021). *On the rényi differential privacy of the shuffle model.* In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2321-2341.
8. Li T, Sahu A K, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020). *Federated optimization in heterogeneous networks.* In: Proceedings of Machine Learning and Systems, 2: 429 – 450.
9. Murtagh J and Vadhan S (2016). *The complexity of computing the optimal composition of differential privacy.* In: Theory of Cryptography Conference, Springer, 157-175.
10. Shen Z, Xia Z, Yu P (2021). *Pldp: Personalized local differential privacy for multidimensional data aggregation.* Security and Communication Networks, 2021(4): 1-13.
11. Yang G, Wang S, Wang H (2021). *Federated learning with personalized local differential privacy.* In: 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), 484-489.
12. On the dependence of the Berry – Esseen bound on dimension.
13. A multivariate Berry – Esseen theorem with explicit constants.
14. Stronger Privacy Amplication by shuffling for Renyi and Approximate differential privacy
15. Gaussian differential privacy
16. Conservative or Liberal?Personalized differential privacy.
17. Privacy and rationality in individual decision making.
18. The privacy blanket of the shuffle model

19. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling
20. On the renyi differential privacy of the shuffle model
21. Rappor: randomized aggregatable privacy-preserving ordinal response.
22. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11): 1998.