PLDP-IDS: Personalized Local Differential Privacy for Infinite Data Stream

Leilei Du [1], Yikun Hu [1], Xu Zhou [1], Yang Cao [2], Peng Cheng [3], Wangze Ni [4], Kenli Li [1]
[1]*Hunan University, Changsha, China;* [2]*Institute of Science Tokyo, Tokyo, Japan;*
[3]*Tongji University, Shanghai, China;* [4]*Zhejiang University, China*
leileidu@hnu.edu.cn; yikunhu@hnu.edu.cn; zhxu@hnu.edu.cn; cao@c.titech.ac.jp;
cspcheng@tongji.edu.cn; niwangze@zju.edu.cn; lkl@hnu.edu.cn

***Abstract*—Streaming data collection is crucial for real-time data analysis, such as event monitoring. However, directly publishing this data can lead to privacy leakage. Local Differential Privacy (LDP) has become a standard technique for protecting individual privacy while maintaining high accuracy in data collection. Nevertheless, most existing LDP research on data streams typically assumes a uniform privacy protection level, which often result in suboptimal utility. In this paper, we address this limitation by introducing PLDP-IDS, a novel personalized LDP framework for infinite data streams. We also propose two population division methods to tailor privacy protection. To further enhance data utility, we propose two additional methods that minimize errors through optimal window-size sampling and improve the accuracy of highly protected data by leveraging information from data with lower privacy levels. We validate the efficiency and effectiveness of PLDP-IDS by conducting extensive experiments on both real-world and synthetic datasets. Experimental results demonstrate that our approaches achieves significantly higher utility accuracy than traditional non-personalized methods, reducing the average error by 43.18% on real datasets and 41.88% on synthetic datasets.**

***Index Terms*—differential privacy, data stream, personalized local differential privacy.**

I. Introduction

With the widespread adoption of smart devices and high-quality wireless networks, users can easily connect to online services. They continuously generate and transmit data streams to service platforms, which in turn collect and analyze these data in real time to deliver more personalized and efficient services.

However, directly collecting streaming data introduces significant privacy risks, leading many users to hesitate before engaging with such platforms. For instance, an HIV-positive person may refuse to participate in a related investigation due to privacy concerns [1]. To address this issue, Differential Privacy (DP) [2] is proposed, which protects individual privacy through a trusted third party. To further mitigate privacy risks arising from reliance on such a third party, Local Differential Privacy (LDP) has been developed as a more decentralized and privacy-preserving solution.

Recently, w-event privacy based on LDP (w-event LDP) has been introduced for private stream data collection and analysis [3]. This approach effectively protects related events within a window size of w (also referred to as an event block or a window). However, in real-world scenarios, users often generate data streams with diverse temporal patterns and varying window lengths [4]. For example, most entertainers prefer not to disclose their locations, while many street artists willingly reveal theirs to attract more attention. Consequently, adopting a uniform w-event window may lead to overprotection for users with smaller event blocks, thereby degrading the accuracy of data estimation.

We illustrate a privacy-preserving example of an online car-hailing scenario in Figure 1.

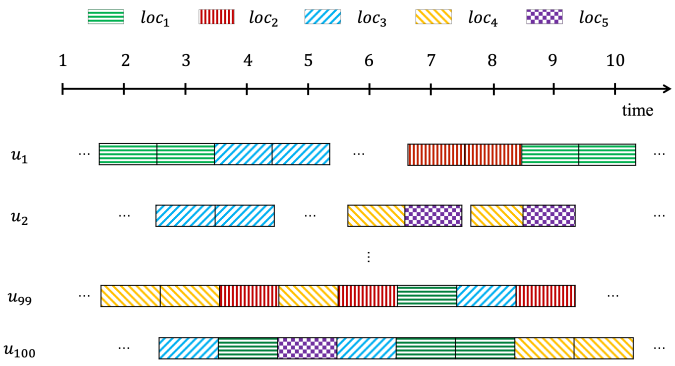


Fig. 1: An example for personalized w-event

Example 1. *Consider a scenario with $n = 100$ drivers $U = \{u_1, \ldots, u_{100}\}$, driving among 5 locations, $\{loc_1, \ldots, loc_5\}$. As shown in Figure 1, these 100 drivers provide their respective stream data. For users u_1 through u_{98}, the largest size of related events is 4 (the window size is no larger than 4). For u_{99} and u_{100}, the largest size of related events is 8. To protect each user's privacy under traditional w-event privacy, we would set the event window size w to 8, fully utilize the privacy budget to maintain high utility while satisfying 8-event privacy. Assuming the total privacy budget $\epsilon = 1$, the upper bound of absolute error for this example under 8-event privacy is:*

$$AE_u = \sqrt{Var_{FO}\left[\frac{\epsilon}{w}, N\right]} = \sqrt{n \times \frac{d - 2 + e^{\epsilon/w}}{(e^{\epsilon/w} - 1)^2}} \approx 152.68.$$

However, it's unnecessary for the first 98 users to have a window size of 8, as this would only achieve 8-event privacy for them. By dividing the 100 users into two groups — group 1 consisting of the first 98 users, and group 2 consisting of the remaining two users — we can create two sub-mechanisms

with distinct privacy levels: w_1-event privacy (with $w_1 = 4$) for group 1 and w_2-event privacy (with $w_2 = 8$ for group 2). With group 1 containing $n_1 = 98$ users and group 2 containing $n_2 = 2$ users, the upper bound of absolute error becomes:

$$AE_{\boldsymbol{u}} = \sqrt{Var_{FO}\left[\frac{\epsilon}{w_1}, N_1\right] + Var_{FO}\left[\frac{\epsilon}{w_2}, N_2\right]} \approx 75.3,$$

which is significantly lower than the error under uniform 8-event privacy.

In this paper, we conduct a detail analysis of error characteristics introduced by Personalized Local Differential Privacy (PLDP) [5] mechanisms in streaming data collection. Based on the theoretical foundation of PLDP, we propose two baseline methods, PLDP Population Distribution (PLPD) and PLDP Population Absorption (PLPA), to improve data utility under personalized privacy settings. However, these two methods do not fully exploit the heterogeneity of personalized window sizes, nor do they leverage the information published under lower privacy protection (i.e., higher utility) to enhance the estimation of high-privacy data. To overcome these limitations, we further design PLDP Population Distribution Plus (PLPD$^+$) and PLDP Population Absorption Plus (PLPA$^+$), which choose the best window size with the lowest error and improve the estimation of high-privacy data using low-privacy data, achieving superior overall utility.

Contributions. We summarize our contributions as follows:

- We formally define the problem of Personalized Private Steaming Data Estimation in Local Setting, which focuses on enable statistical analysis of personalized data streams while ensuring personalized privacy protection (i.e., w-event PLDP) in Section III.
- We propose two effective population division-based methods PLDP Population Distribution (PLPD) and PLDP Population Absorption (PLPA) under PLDP in Section IV.
- We further introduce two enhanced methods PLDP Population Distribution Plus (PLPD$^+$) and PLDP Population Absorption Plus (PLPA$^+$) in Section V.
- We evaluate the proposed methods on both real and synthetic datasets to demonstrate their efficiency and effectiveness in Section VI.

II. Related Work

A. Local Differential Privacy on Streaming Data

Differential Privacy (DP) [2] provides a rigorous privacy framework but relies on a trusted third party to perturb data before aggregation, which limits its practicality in decentralized settings. Local Differential Privacy (LDP) [6] addresses this issue by allowing each user to randomize their data locally, removing the need for centralized trust. LDP has been widely adopted by companies such as Microsoft, Apple, and Google. Streaming data estimation under LDP can be categorized into event-level, user-level, and w-event LDP, depending on the granularity and temporal scope of privacy protection.

Event-level LDP (event-LDP) protects each individual record independently. Representative works such as RAPPOR [7] and ToPL [8] enable privacy-preserving data collection and aggregation under the local model. However, event-LDP focuses solely on single events and neglects correlations among events within a user's data stream.

Recent advances in user-level LDP have focused on addressing temporal correlations in time series and improving long-term utility [9]–[12]. These studies propose mechanisms for continual data collection, adaptive budget allocation, and pattern extraction, enabling user-level privacy preservation over time. However, existing approaches are limited to finite or constrained settings and still face challenges such as cumulative budget consumption, rigid structural assumptions, and limited poor adaptability to highly dynamic or unsegmented data streams.

w-event privacy is first proposed by Kellaris et al. [13] under DP settings, which provides privacy protection for event sequences occurring within a window of length w. Ren et al. [3] extend w-event privacy to local settings, and propose LDP-IDS, a framework for infinite streaming data collection and analysis under the w-event LDP model. They introduce two budget allocation methods and two population allocation methods to bridge the gap between event-level and user-level LDP, improving estimation accuracy. Despite these advancements, existing methods still fail to accommodate personalized event window sizes, limiting their flexibility in real-world applications.

B. Personalized Differential Privacy

Personalized Differential Privacy (PDP), also known as Heterogeneous Differential Privacy (HDP) [14], can be divided into item-grained and user-grained privacy models. In the item-grained setting, PDP assigns different privacy levels to individual attributes or features within a dataset, while in the user-grained setting, each user specifies a personalized privacy budget, enabling heterogeneous privacy guarantees across users.

Item-grained Privacy. Kotsogiannis et al. [15] observe that records vary in sensitivity and propose One-Sided Differential Privacy (OSDP), which distinguishes between sensitive and non-sensitive data to improve utility. However, protecting the sensitivity state itself requires perturbing non-sensitive data, thereby reducing estimation accuracy. Kifer and Machanavajjhala [16] introduce the Pufferfish framework, enabling customizable privacy protection across attributes and relationships. Building on this, He et al. [17] propose Blowfish, which relaxes protection for less sensitive attributes to enhance practicality, and has been extended to domains such as graph privacy [18] and interactive data exploration [19]. Despite their effectiveness [20]–[23], both frameworks rely on manual policy specification and complex modeling, limiting their scalability and adoption in practical systems. Song et al. [24] observe that attributes vary in sensitivity and propose Personalized Randomized Response (PRR), a perturbation framework that assigns attribute-specific weights to enhance

TABLE I: Summary for related work.

Categories	Model Types	Methods	Infinite & correlated	Trust-free	Personalized
LDP on streaming data	event-level privacy	RAPPOR [7]	×	✓	×
		ToPL [8]	×	✓	×
	user-level privacy	CGM [9]	✓	✓	×
		DDRM [10]	✓	✓	×
		StaSwitch [11]	✓	✓	×
		PrivShape [12]	✓	✓	×
	w-event LDP	LDP-IDS [3]	✓	✓	×
Personalized DP	item-grained privacy	OSDP [15]	×	×	✓
		Pufferfish [16], [20]–[23]	✓	×	✓
		Blowfish [17]–[19]	✓	×	✓
		PRR [24]	×	✓	✓
	user-grained privacy	SM [26]	×	×	✓
		AdaPDP [27]	×	×	✓
		ADPM [28]	×	×	✓
		PCE [29]	×	✓	✓
		HDG-COE [30]	×	✓	✓
		PLDP-MRQ [5]	×	✓	✓
w-event under PDP		PBD & PBA [4]	×	✓	✓
Out methods			✓	✓	✓

data utility under personalized protection. Compared with traditional Randomized Response (RR) [25], PRR achieves higher statistical accuracy; however, it focuses only on attribute sensitivity, overlooking individual differences.

User-grained Privacy. Jorgensen et al. [26] introduce Personalized Differential Privacy (PDP), allowing users to specify individual privacy requirements, and propose the Sample Mechanism (SM) to minimize utility loss. Building on this idea, Niu et al. [27] develop Adaptive Personalized Differential Privacy (AdaPDP), which adaptively selects noise-generation strategies based on query type, data distribution, and privacy preferences to maximize utility across multiple rounds. However, its iterative sampling incurs high computational cost and depends heavily on model training, limiting real-world applicability. More recently, Chaudhuri et al. [28] propose an Affine Differentially Private Mean Estimator (ADPM) for heterogeneous DP, achieving minimax-optimal estimation under heterogeneous privacy constraints. However, it assume all users' data are independent and identically distributed (i.i.d.) samples of the same distribution. Chen et al. [29] introduce PLDP, a personalized local differential privacy model that allows each user to define a safe region and privacy level, along with lightweight protocols such as PCE and PSDA to improve count estimation accuracy. Li et al. [30] propose HDG-COE, which enables users to specify personalized indistinguishable areas for protection, achieving high utility, though it incurs considerable time complexity in high-dimensional settings. He et al. [5] develop PLDP-MRQ, a framework for personalized local differential privacy over multi-dimensional range queries, combining personalized random rotation with hierarchical aggregation for improved accuracy. Despite these advancements, existing user-grained approaches remain limited in capturing evolving user behavior and cannot maintain privacy guarantees under continuous temporal correlations.

III. Problem Settings

In this section, we first introduce key conceptions related to data streams and personalized local differential privacy. Next, we outline the main assumptions of our model. We then present our new privacy definition: $\boldsymbol{w}$-Event $\boldsymbol{\epsilon}$-Personalized Local Differential Privacy ($(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP). Finally, we provide the problem definition: Personalized Private Steaming Data Estimation in Local Setting (PPSDELS). Table II summarizes the notions used in this paper.

TABLE II: Notations.

Notations	Description
u_i	the i-th user
n	the size of U
$\boldsymbol{w}$	the privacy window size of U
w_i	u_i's privacy window size
$\boldsymbol{\epsilon}$	the privacy budget set of U
ϵ_i	u_i's privacy budget
$\tilde{\boldsymbol{\epsilon}}$	the unique list of $\boldsymbol{\epsilon}$
$\tilde{\epsilon}_k$	the k-the value in $\tilde{\boldsymbol{\epsilon}}$
$\boldsymbol{g}$	the frequency list of users' privacy budget requirement
g_k	the frequency of users requiring $\tilde{\epsilon}_k$ (i.e., $g_k = \boldsymbol{g}[k]$)
d	the domain size of the attribute
m	the number of unique privacy budget requirements
$\boldsymbol{f}$	the real statistical histogram at any time slot
$\boldsymbol{f}_t$	the real statistical histogram at time slot t
$\hat{\boldsymbol{f}}_t$	the estimation statistical histogram at time slot t
$\boldsymbol{h}$	the obfuscated statistical frequency at any time slot
$\boldsymbol{h}_t$	the obfuscated statistic frequency at time slot t
ω_j	the j-th value in the domain of A
g_k	the frequency of requiring $\tilde{\epsilon}_k$ among all users
$f_j, \boldsymbol{f}[j]$	the frequency that ω_j occurs
$\hat{f}_j, \hat{\boldsymbol{f}}[j]$	the estimation value of f_j
$h_j, \boldsymbol{h}[j]$	the obfuscated frequency that ω_j occurs

A. Data Streams

Let $D \in \mathcal{D}$ denote a database with a single attribute and n items. Each item x_i corresponds to the data of user u_i. Let d represent the domain size of the attribute, and let the domain be defined as $\Omega = \{\omega_1, \ldots, \omega_d\}$. Then D can be represent as an $n \times d$ binary matrix, where the value $b_{i,j} \in \{0, 1\}$ at the i-th row and j-th column indicates whether the data value of u_i is ω_j ($b_{i,j} = 1$) or not ($b_{i,j} = 0$).

Definition 1 (Data Stream [13])**.** Let $D_t \in \mathcal{D}$ be a database at t-th time slot. The infinite sequence $S = \langle D_1, D_2, \ldots \rangle$ is called a data stream, where $S[t]$ denotes the t-th element of S (i.e., $S[t] = D_t$).

Let $S_{\tau,t}$ ($1 \leq \tau \leq t$) be a sub-stream of S from time slot τ to time slot t, with length $t - \tau + 1$. Specifically, when $\tau = 1$, we abbreviate $S_{\tau,t}$ as S_t, which also represents a stream prefix of S composed of tuples arriving on or before time t.

Definition 2 (w-Neighboring Stream Prefixes [13], [31])**.** Let w be a positive integer. Two stream prefixes S_t, S'_t are w-neighboring (i.e., $S_t \sim_w S'_t$), if:

1) for each $S_t[\tau]$, $S'_t[\tau]$ such that $\tau \leq t$ and $S_t[\tau] \neq S'_t[\tau]$, it holds that $S_t[\tau]$ and $S'_t[\tau]$ are neighboring [13] in centralized DP, and
2) for each $S_t[\tau_1]$, $S_t[\tau_2]$, $S'_t[\tau_1]$, $S'_t[\tau_2]$ with $\tau_1 \leq \tau_2$, $S_t[\tau_1] \neq S'_t[\tau_1]$ and $S_t[\tau_2] \neq S'_t[\tau_2]$, it holds that $\tau_2 - \tau_1 + 1 \leq w$.

Definition 3 (Data Stream Frequency Estimation [4])**.** Let $Q : \mathcal{D} \to \mathbb{R}^d$ be a frequency query. Let $Q(S[t]) = Q(D_t) = \boldsymbol{f}_t$ be the frequency vector of all $\omega \in \Omega(A)$ to be published at time slot t, where $\boldsymbol{f}_t[j]$ represents the frequency of ω_j in D_t. The infinite data frequency series $\langle \boldsymbol{f}_1, \boldsymbol{f}_2, \ldots \rangle$ is called a data stream frequency estimation.

B. $\boldsymbol{w}$-Event $\boldsymbol{\epsilon}$-Personalized Local Differential Privacy

Let $\mathcal{U}$ be the domain of users and $\mathcal{E}$ be the domain of privacy budgets. Let $U \subseteq \mathcal{U}$ be the user set with n users, specifically, $U = \{u_1, \ldots, u_n\}$. Let $\boldsymbol{\epsilon} : \mathcal{U} \to \mathcal{E}$ be the privacy budget requirement function, where $\boldsymbol{\epsilon}(u_i) = \epsilon_i$ indicates that u_i requires ϵ_i privacy budget. We define $\tilde{\boldsymbol{\epsilon}} = \text{Unique}(\boldsymbol{\epsilon})$ as the unique privacy budget list with size $m = |\tilde{\boldsymbol{\epsilon}}|$. Let $\mathcal{D}_k$ be the database domain protected by $\tilde{\boldsymbol{\epsilon}}_k$.

Definition 4 (Budget Group and Budget Group Data)**.** Given a unique privacy budget list $\tilde{\boldsymbol{\epsilon}}$ with $|\tilde{\boldsymbol{\epsilon}}| = m$ and a user set $U \subseteq \mathcal{U}$ with n users, where each user u_i holds a privacy budget $\epsilon_i \in \tilde{\boldsymbol{\epsilon}}$, U can be divided into m groups $\{U_1, U_2, \ldots U_m\}$ based on distinct privacy budgets. The pair $\boldsymbol{BG}_j = (\epsilon_j, U_j)$ is called the j-th Budget Group of U. Let Y_j be the output of each $u_i \in U_j$ achieving ϵ_j-Local Differential Privacy on the data domain Ω. The tuple $\boldsymbol{BGD}_j = (\epsilon_j, U_j, Y_j)$ is called the j-th Budget Group Data of U.

Definition 5 ($\boldsymbol{\epsilon}$-Personalized Local Differential Privacy, $\boldsymbol{\epsilon}$-PLDP [5])**.** A mechanism $\mathcal{M}$ satisfies $\boldsymbol{\epsilon}$-PLDP if and only if, for any user u with possible input values $x, x' \in D_k \subseteq \mathcal{D}_k$,

$$\forall y \in \text{Range}(\mathcal{M}) : \Pr[\mathcal{M}(x) = y] \leq e^{\boldsymbol{\epsilon}(u)} \Pr[\mathcal{M}(x') = y],$$

where $\text{Range}(\mathcal{M})$ denotes the set of all possible outputs of $\mathcal{M}$.

Definition 6 ($\boldsymbol{w}$-Event $\boldsymbol{\epsilon}$-Personalized Local Differential Privacy, $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP)**.** Let $\mathcal{M}$ be a mechanism that takes a stream prefix of arbitrary τ size as inputs. Let $\mathcal{O}$ be the set of all possible outputs of $\mathcal{M}$. For any universe set of users $U = \{u_1, u_2, \ldots, u_{|U|}\}$, $\mathcal{M}$ satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP if $\forall w_i \in \boldsymbol{w}$, $\forall S_\tau, S'_\tau$ satisfying $S_\tau \sim_{w_i} S'_\tau$ and $\forall O \subseteq \mathcal{O}$, it holds that

$$\Pr[\mathcal{M}(S_\tau) \in O] \leq e^{\boldsymbol{\epsilon}(u_i)} \Pr[\mathcal{M}(S'_\tau) \in O],$$

where u_i requires w_i-event ϵ_i-LDP privacy. When the output set is discrete, i.e. $O = \{y_1, y_2, \ldots y_{|O|}\}$, the condition above can also be rewritten as

$$\Pr[\mathcal{M}(S_\tau) = y] \leq e^{\boldsymbol{\epsilon}(u_i)} \Pr[\mathcal{M}(S'_\tau) = y].$$

The pair (w_i, ϵ_i) is denoted as u_i's *privacy requirement* [4]. When $w_i \equiv 1$, $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP collapses to $\boldsymbol{\epsilon}$-PLDP [5]. Additionally, when (w_i, ϵ_i) is a constant pair $\big($i.e., $(w, \epsilon)\big)$, it collapses to w-Event ϵ-LDP [3].

C. Privacy Definition

Definition 7. ($\boldsymbol{w}$-Event $\boldsymbol{\epsilon}$-Personalized Local Differential Privacy, $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP). Let $\mathcal{M}$ be a mechanism that takes a stream prefix S_t of arbitrary size as input. Given a universe of n users $U = \{u_1, u_2, \ldots, u_n\}$, $\mathcal{M}$ is $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP if $\forall w_i \in \boldsymbol{w}$ and $\forall S_t, S'_t$ satisfying $S_t \sim_{w_i} S'_t$, it holds that

$$\forall Y \in Range(M), \Pr[M(S_t) = Y] \leq e^{\epsilon_i} \Pr[M(S'_t) = Y],$$

where $u_i \in U$ requires w_i-event privacy and ϵ_i denotes u_i's privacy budget requirement within w_i continuous events.

Similar to the formulation in CDP setting [4], we define the pair (w_i, ϵ_i) as the *privacy requirement* of u_i. Specifically, when $w_i = 1$, this reduces to $\boldsymbol{\epsilon}$-Personalized Local Differential Privacy ($\boldsymbol{\epsilon}$-PLDP) [5], [29]. Moreover, when (w_i, ϵ_i) is fixed for all users (i.e., (w, ϵ)), it corresponds to w-Event Privacy [3], [13].

D. Problem Definition

Given a data stream S, the server aims to obtain the data stream frequency estimation, denoted as $\langle \boldsymbol{f}_1, \boldsymbol{f}_2, \ldots \rangle$. However, to protect user privacy in local setting, the server can only receive the obfuscated stream from each u_i and subsequently publishes the obfuscated data stream frequency estimation, denoted as $\left\langle \hat{\boldsymbol{f}}_1, \hat{\boldsymbol{f}}_2, \ldots \right\rangle$. We now define the problem as follows.

Definition 8. (PPSDELS Problem). Given any positive integer $T \in \mathbb{N}^+$, a user set $U = \{u_1, u_2, ..., u_n\}$, where each u_i holds a privacy requirement pair (w_i, ϵ_i) and a series of data $x_{i,t}$ for $t \in [T]$, all the $x_{i,t}$ at time slot t form D_t, and all the D_t form a data stream prefix $S_T = \langle D_1, D_2, \ldots, D_T \rangle$. PPSDELS is to publish an obfuscated stream prefix frequency estimation $\left\langle \hat{\boldsymbol{f}}_1, \hat{\boldsymbol{f}}_2, \ldots, \hat{\boldsymbol{f}}_T \right\rangle$ of S_T achieving $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP with the error between $\hat{\boldsymbol{f}}$ and $\boldsymbol{f}$ minimized, namely:

$$\min \sum_{t \in [T]} \|\hat{\boldsymbol{f}}_t - \boldsymbol{f}_t\|_2^2$$

$$s.t. \sum_{\tau = \min(t - w_i + 1, 1)}^{t} \epsilon_{i,\tau} \leq \epsilon_i, \quad \forall u_i \in U$$

where $\epsilon_{i,\tau}$ denotes the privacy budget at time slot τ.

IV. Population Division-Based Approaches

Many studies [3], [25], [32] have shown that partitioning users into groups and using the entire privacy budget within each group achieves higher utility compared to dividing the budget. In this section, we first encapsulate frequency estimation under PLDP as the Personalized Frequency Oracle (PFO) protocol. We then analyze the errors introduced by population division in PFO. Finally, we propose two population division-based approaches: PLDP Population Distribution (PLPD) and PLDP Population Absorption (PLPA), which achieve accurate estimation by dynamically recycling population while satisfying PLDP constraints.

A. Personalized Frequency Oracle

Similar to Frequency Oracle (FO) [33] in the LDP model, frequency estimation under PLDP can also be encapsulated into a standard protocol. Here, we refer to this as the Personalized Frequency Oracle (PFO).

PFO consists of three functions: the ***P**erturbation* function PFO.P, the ***A**ggregate* function PFO.A and the ***E**stimation* function PFO.E. The function PFO.P randomizes the raw data into obfuscated data using the PLDP algorithm. PFO.A calculates the aggregate statistic for different parts of the obfuscated data. PFO.E is responsible for estimating the distribution of raw data.

There are many implementations of PFO, such as Personalized Hybrid-Dimensional Grids (P-HDG) [5] and Personalized Low-Dimensional Hierarchy-Interval Optimization (P-HIO) [5]. Here, we illustrate another implementation of PFO called Generalized Personalized Random Response (GPRR). In GPRR, the output domain is equal to the input domain. Let p_i be the probability that u_i reports real data (i.e., $y_i = x_i$), then the probability of reporting any fake data (i.e., $y_i \in \Omega \backslash \{x_i\}$) is $q_i = \frac{1-p_i}{d-1}$. Specifically, we have:

$$\textbf{PFO.P}: \forall y_i \in \Omega, \Pr[M(x_i) = y_i] = \begin{cases} p_i = \dfrac{e^{\epsilon_i}}{e^{\epsilon_i} + d - 1}, & \text{if } y_i = x_i, \\ q_i = \dfrac{1}{e^{\epsilon_i} + d - 1}, & \text{otherwise.} \end{cases}$$

After receiving the obfuscated reports $\boldsymbol{y}_k \subseteq \boldsymbol{Y}$ from all users across different groups divided by distinct $\tilde{\epsilon}_k \in \tilde{\boldsymbol{\epsilon}}$, the server computes the obfuscated statistic of each value $\omega_j \in \Omega$ for every group G_k as follow:

$$\textbf{PFO.A}: \forall k \in [m], j \in [d], h_{k,j} = \frac{\text{Count}\left(y_{k,j}\right)}{n \cdot g_k},$$

The obfuscated statistic of group G_k is denoted as $\boldsymbol{h}_k =< h_{k,1}, h_{k,2}, \ldots, h_{k,d} >$. Subsequently, the server estimates the global frequency distribution as:

$$\textbf{PFO.E}: \forall \hat{f}_j \in \hat{\boldsymbol{f}}, \hat{f}_j = \sum_{k=1}^{m} \alpha_k \cdot \frac{h_{k,j} - \tilde{q}_k}{\tilde{p}_k - \tilde{q}_k}, \tag{1}$$

where $\alpha_k = \frac{1/\operatorname{Var}[\hat{f}_{k,j}]}{\sum_{k'=1}^{m} 1/\operatorname{Var}[\hat{f}_{k',j}]}$ is the weight of group G_k in the final obfuscated estimation, and $\hat{f}_j^{(k)}$ denotes the estimation of ω_j in Group G_k. And $\operatorname{Var}\left[\hat{f}_{k,j}\right]$ represents the variance under ϵ_k-LDP [5].

However, when implementing ϵ_k-LDP using the Generalized Randomized Response (GRR) mechanism, the variance is given by $\operatorname{Var}\left[\hat{f}_j^{(k)}\right] = \frac{\tilde{q}_k(1-\tilde{q}_k)}{n g_k(\tilde{p}_k-\tilde{q}_k)^2} + \frac{1-\tilde{p}_k-\tilde{q}_k}{n g_k(\tilde{p}_k-\tilde{q}_k)} \cdot f_j$, which depends on the true frequency f_j. Since f_j is unknown, this formulation is infeasible for practical computation. To address this issue, we modify the weight definition as $\alpha_k = \frac{1/\overline{\operatorname{Var}}[\hat{f}_{k,j}]}{\sum_{k'=1}^{m} 1/\overline{\operatorname{Var}}[\hat{f}_{k',j}]}$, where the average variance is given by $\overline{\operatorname{Var}}\left[\hat{f}_{k,j}\right] = \frac{\tilde{q}_k(1-\tilde{q}_k)}{n g_k(\tilde{p}_k-\tilde{q}_k)^2} + \frac{1-\tilde{p}_k-\tilde{q}_k}{n g_k(\tilde{p}_k-\tilde{q}_k)} \cdot \frac{1}{d}$. Here, $\tilde{p}_k$ and $\tilde{q}_k$ represent the probabilities that users with $\tilde{\epsilon}_k$ report their true data and fake data, respectively. The weight parameter α_k is therefore determined solely by g_k and $\tilde{q}_k$.

Based on the modified weights, we can calculate the total variance in PLDP as:

$$Var\left[\hat{\boldsymbol{f}}\right] = 1/\sum_{k=1}^{m} n/\left(d\lambda_k + \mu_k\right), \tag{2}$$

where $\lambda_k = \frac{\tilde{q}_k(1-\tilde{q}_k)}{g_k(\tilde{p}_k-\tilde{q}_k)^2}$ and $\mu_k = \frac{1-\tilde{p}_k-\tilde{q}_k}{g_k(\tilde{p}_k-\tilde{q}_k)}$. For more details of the modified weights and the calculation process for the total variance, please refer to Appendix IX-B our report [38]. We abbreviate the total variance in Equation (2) as $\mathrm{V}_{PLDP}(\tilde{\boldsymbol{\epsilon}}, \boldsymbol{g}, n, d)$, where $\boldsymbol{g}$ is the count list of $\tilde{\boldsymbol{\epsilon}}$ among the n users.

B. Private Strategy Determination

The core idea of population division methods is to adaptively adjust the participating population at each time slot. This process can be regarded as a form of *strategy determination*. Specifically, the method compares the private dissimilarity *dis* with the reporting error *err* at the current time slot. If *dis* exceeds *err*, the mechanism performs a new estimation using a new portion of population. Otherwise, it reuses the previous estimation as an approximation, without consuming additional population resources.

Dissimilarity. The definition of *dis* in $\boldsymbol{\epsilon}$-PLDP is similar to that in the standard LDP, which is defined as

$$dis = \frac{1}{d}\sum_{j=1}^{d}\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2 - \frac{1}{d}\sum_{j=1}^{d} Var\left[\hat{\boldsymbol{f}}_{s,t}[j]\right]. \tag{3}$$

Further details about the computation and analysis of *dis* are provided in Appendix IX-C of our report [38]. In contrast, the definition of *err* in PLDP is more complex and differs substantially from that in $\boldsymbol{\epsilon}$-PLDP. Therefore, we need to redefine these variables accordingly.

Reporting Errors. Similar to population division methods under LDP [32], population division under PLDP can also be decomposed into two components: the *variance due to PLDP perturbation* and the *variance due to sampling*. The variance for PLDP perturbation, denoted as $\mathrm{V}_{PLDP}(\tilde{\boldsymbol{\epsilon}}, \boldsymbol{g}, n, d)$, has been shown in Equation (2).

Variance for Sampling. Given a sample of users U with size z, let X_j denote the number of occurrences of $\omega_j \in \Omega$ within this sample. Then X_j follows a Hypergeometric Distribution [34], and its variance can be expressed as

$$\operatorname{Var}[X_j] = z \cdot f_j \cdot (1 - f_j) \cdot \frac{n-z}{n-1}.$$

Accordingly, the variance of estimated frequency for Ω_j is:

$$\operatorname{Var}\left[\hat{f}_j\right] = \frac{1}{z^2}\operatorname{Var}[X_j] = \frac{f_j \cdot (1-f_j)}{z} \cdot \frac{n-z}{n-1}.$$

Thus, the variance of the entire frequency estimation is given by:

$$\operatorname{Var}\left[\hat{\boldsymbol{f}}\right] = \sum_{j=1}^{d}\frac{f_j \cdot (1-f_j)}{z} \cdot \frac{n-z}{n-1} = \frac{n-z}{z(n-1)}\left(1 - \sum_{j=1}^{d} f_j^2\right). \tag{4}$$

When the domain size d is large, $\sum_{j=1}^{d} f_j^2 \approx 0$, and the upper bound of Equation (4) becomes:

$$\operatorname{Var}^*\left[\hat{\boldsymbol{f}}\right] = \frac{n-z}{z(n-1)}. \tag{5}$$

We abbreviate Equation (5) as $\mathrm{V}_{smpl}(n, z)$.

Reporting Errors for population division under PLDP. The total variance combining both sampling and perturbation effects is expressed as:

$$\begin{aligned} &\mathrm{V}(\tilde{\boldsymbol{\epsilon}}, \boldsymbol{g}, n, z, d) \\ =&\mathrm{V}_{smpl}(n, z) + \mathrm{V}_{PLDP}(\tilde{\boldsymbol{\epsilon}}, \boldsymbol{g}, z, d) \\ =&\frac{n-z}{z(n-1)} + 1/\sum_{k=1}^{m} z/\left(d\lambda_k + \mu_k\right), \end{aligned} \tag{6}$$

We defined the reporting error as the average variance across all values in the output domain, given by:

$$\begin{aligned} &err(\tilde{\boldsymbol{\epsilon}}, \boldsymbol{g}, n, z, d) \\ =&\frac{1}{d}\left(\mathrm{V}_{smpl}\left(n, z\right) + \mathrm{V}_{PLDP}(\tilde{\boldsymbol{\epsilon}}, \boldsymbol{g}, z, d)\right) \\ =&\frac{n-z}{dz(n-1)} + 1/\sum_{k=1}^{m} dz/\left(d\lambda_k + \mu_k\right), \end{aligned} \tag{7}$$

From Equation (7), we observe that, for fixed privacy requirements, a larger sampling size results in a smaller reporting error.

After defining the dissimilarity and reporting error, we now propose two methods: PLDP Population Distribution (PLPD) and PLDP Population Absorption (PLPA), which follow the main idea described above. Both methods share the same private dissimilarity calculation module ($\mathcal{M}_{s,t}$), which consumes a fixed-size population $U_{s,t}$ at each time slot. However, they differ in the private publication module ($\mathcal{M}_{r,t}$), where each method adopts a distinct dynamic allocation strategy for the publication population $U_{r,t}$ across the steam. We refer to $U_{s,t}$ as the *dissimilarity population*, and $U_{r,t}$ as the *publication population*.

C. PLDP Population Basic Solutions

To address the problem under personalized local differential privacy settings, we propose two basic PLDP population solutions: PLDP Population Distribution (PLPD) and PLDP Population Absorption (PLPA).

Both PLPD and PLPA are built upon LDP Budget Distribution (LBD) and LDP Budget Absorption (LBA), respectively, as described in Reference [3]. At the initialization stage, both methods adopt a uniform population sampling size defined as $z = \min_{i\in[n]} \left\lfloor \frac{n}{w_i} \right\rfloor = \left\lfloor \frac{n}{w_{max}} \right\rfloor$, and compute the dissimilarity *dis* and error *err* following the same principles as in PLPD and PLPA, while applying PFO for each estimation. Due to space limitations, detailed descriptions are deterred to Appendix IX-A of our report [38].

D. Analysis

Time Complexity Analysis. Let d denote the value domain size, m the number of budget groups, and n the total number of users, satisfying $m \leq n$. Let z_{min} be the chosen sampling size at each time slot, with $z_{min} \leq n/2$. We analyze the time complexity of PLPD and PLPA as follows.

Theorem IV.1. *The time complexity of both PLPD and PLPA is* $O(n + d \cdot m)$.

Proof. Please refer to the detailed proof of Theorem IV.1 in Appendix IX-E1 of our report [38]. □

Privacy Analysis. When the window size is fixed (i.e., $w = w_{max}$), each user in both PLPD and PLPA participates at most once within this window. Furthermore, every estimation is released through a PFO mechanism that satisfies $\boldsymbol{\epsilon}$-PLDP. Therefore, the following theorem holds for the privacy guarantee.

Theorem IV.2. *PLPD and PLPA satisfy* $(\boldsymbol{w}, \boldsymbol{\epsilon})$*-EPLDP.*

Proof. Please refer to the detailed proof of Theorem IV.2 in Appendix IX-F1 of our report [38]. □

Utility Analysis. For simplicity, given the maximum window size w_{max}, we assume that at most $s < w_{max}$ new publications occur at time slots $\rho_1, \rho_2, \ldots \rho_s$, without any budget absorption from past time slots preceding the current window. In addition, there exists an equal number of skipped / nullified publications corresponding to these new publications. Let $\tilde{\boldsymbol{n}}$ denote the user count list, where $\tilde{\boldsymbol{n}}(k)$ represents the number of users requiring privacy budget list $\tilde{\boldsymbol{\epsilon}}(k)$. Based on these settings, we provide the following two theorems for PLPD and PLPA, respectively.

Theorem IV.3. *The error upper bound of per time slot in PLPD is* $\frac{1}{d^3}\left(\frac{2w_{max}}{n-1} - \frac{1}{n-1} + 2w_{max}A\right)^2 + \frac{4(2^s-1)}{s} \cdot \left(\frac{1}{n-1} + A\right) - \frac{1}{n-1}$, *where* $A = \frac{d-1}{\min(\tilde{\boldsymbol{n}})} \cdot \frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}$.

Proof. The detailed proof of Theorem IV.3 is provided in Appendix IX-G1 of our report [38]. □

Theorem IV.4. *The error upper bound of per time slot in PLPA is* $\frac{1}{d^3}\left(\frac{2w_{max}}{n-1} - \frac{1}{n-1} + 2w_{max}A\right)^2 + BC \cdot err_{nlf} - \frac{(B+1)C}{n-1} + \left(\frac{s}{n-1} + s \cdot A\right) \cdot \left(2 - \frac{1}{2^{B+1}} + \frac{2}{B}\right)$, *where* $A = \frac{d-1}{\min(\tilde{\boldsymbol{n}})} \cdot \frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}$, $B = \frac{w_{max}-s}{2s}$ *and* $C = \frac{s}{w_{max}}$.

Proof. The detailed proof of Theorem IV.4 is provided in Appendix IX-G1 of our report [38]. □

V. Enhanced Approaches

PLPD and PLPA address the population assignment by unifying the window size to the maximum value, i.e., $w_{max} = \max(w_1, w_2, \ldots, w_n)$. However, these two methods face two main issues: (1) the use of maximum window size results in a minimum sampling size at each time slot, leading to large sampling errors; (2) in personalized local different privacy settings, data with higher privacy protection (i.e., smaller ϵ) can reduce the accuracy of the overall estimation.

To address the first issue, we propose Optimal Population Selection (OPS), which selects an optimal sampling size to minimize the reporting error while maintaining privacy levels through budget division. For the second issue, we propose a technique to re-disturb high-ϵ data and combine it with low-ϵ data to enhance the accuracy of the latter.

A. Optimal Population Selection

In personalized local differential privacy settings, the optimal sampling size strategy should take into account both LDP error and sampling error. Thus, we need to determine the optimal sampling size z_{opt} that minimizes the total error. This process is called Optimal Population Selection (OPS), and illustrated in Algorithm 1.

OPS first calculates the unique sampling sizes $\tilde{\boldsymbol{z}}$. Then it iterates over all sampling size in $\tilde{\boldsymbol{z}}$ to find the optimal one that minimizes the reporting error (Lines 3–16). To maintain privacy protection, during the iteration, the privacy budget ϵ_i of users whose sampling size z_i is smaller than the current $\tilde{z}$ are

Algorithm 1: Optimal Population Selection (OPS)

Input: The sampling size requirement list $\boldsymbol{z} = \langle z_1, \ldots, z_n \rangle$; the privacy budget requirement list $\boldsymbol{\epsilon} = \langle \epsilon_1, \ldots \epsilon_n \rangle$; the value domain size d

Output: The optimal sampling size z_{opt}, the optimal error err_{opt}, the trasformed privacy budget vector $\boldsymbol{\epsilon}'_{opt}$

1 Initialize err_{opt} as the upper bound of error value;
2 Set $\tilde{\boldsymbol{z}} \leftarrow \text{Unique}(\boldsymbol{z})$ as the unique list of $\boldsymbol{z}$;
3 **for** $\tilde{z} \in \tilde{\boldsymbol{z}}$ **do**
4 Initialize $\boldsymbol{\epsilon}' \leftarrow \emptyset$;
5 **for** $z_i \in \boldsymbol{z}$ **do**
6 **if** $z_i < \tilde{z}$ **then**
7 add $\epsilon_i / \lceil \tilde{z}/z_i \rceil$ to $\boldsymbol{\epsilon}'$;
8 **else**
9 add ϵ_i to $\boldsymbol{\epsilon}'$;
10 Set $\tilde{\boldsymbol{\epsilon}}' \leftarrow \text{Unique}(\boldsymbol{\epsilon}')$;
11 Count the privacy requirement frequency $\boldsymbol{g}'$ of $\tilde{\boldsymbol{\epsilon}}'$;
12 Calculate $err_{tmp} \leftarrow \mathbb{V}(\tilde{\boldsymbol{\epsilon}}', \boldsymbol{g}', n, \tilde{z}, d)$ according to Equation (6);
13 **if** $err_{tmp} < err_{opt}$ **then**
14 $err_{opt} \leftarrow err_{tmp}$;
15 $z_{opt} \leftarrow \tilde{z}$;
16 $\boldsymbol{\epsilon}'_{opt} \leftarrow \boldsymbol{\epsilon}'$;
17 **return** z_{opt}, err_{opt}, $\boldsymbol{\epsilon}'_{opt}$;

divided into $\lceil \tilde{z}/z_i \rceil$ shares. The privacy budget is then updated as a share of size $\epsilon_i / \left\lceil \frac{\tilde{z}}{z_i} \right\rceil$ (Lines 5–9). OPS calculates the statistic $\boldsymbol{g}'$ of distinct new privacy budgets $\tilde{\boldsymbol{\epsilon}}'$ and records the minimum error err_{opt} with the corresponding optimal sampling size z_{opt} (Lines 10–15).

An example of OPS is provided in Example 2.

Example 2. *Assume there are* 10 *users with sampling sizes* $\boldsymbol{z} = \langle 3, 5, 9, 8, 9, 5, 6, 5, 8, 9 \rangle$ *and personalized privacy budgets* $\boldsymbol{\epsilon} = \langle 0.1, 0.4, 0.4, 0.1, 0.4, 0.4, 0.8, 0.8, 0.8, 0.4 \rangle$ *with* $d = 2$*. Then, the distinct sampling sizes are* $\tilde{\boldsymbol{z}} = [3, 5, 6, 8, 9]$*. OPS iterates over all* $\tilde{z} \in \tilde{\boldsymbol{z}}$*, dividing the privacy budgets with smaller sampling sizes and calculating the corresponding error. Take* $\tilde{z} = 6$ *as an example. Users with* $z = 3$ *and* $z = 5$ *divide their privacy budgets, resulting in* $\boldsymbol{\epsilon}' = [0.05, 0.2, 0.4, 0.1, 0.4, 0.2, 0.8, 0.4, 0.8, 0.4]$ *and* $err = 0.988$*. The calculated errors for all* $\tilde{\boldsymbol{z}}$ *are* $[1.121, 0.596, 0.988, 0.721, 1.124]$*, yielding the optimal sampling size* $z_{opt} = 5$ *with minimum error* $err = 0.596$*.*

B. Utility Improvement

To further enhance utility at each private calculation (e.g., *dis* calculation or new estimation), we propose using obfuscated data with low privacy levels to improve the accuracy of those with higher privacy level. This idea has been explored in some privacy protection studies [5], [35]. Similar enhancement concepts also appear in learning-based frameworks [36], [37]. Here, we extend the re-perturbation method in Reference [5] in GRR settings and introduce the General Personalized Randomized Response (GPRR) mechanism as follows.

Algorithm 2: Personalized Utility Enhancement (PUE)

Input: A privacy budget list $\tilde{\boldsymbol{\epsilon}} = \langle \tilde{\epsilon}_1, \tilde{\epsilon}_2, \ldots \tilde{\epsilon}_m \rangle$; a perturbed data list $\boldsymbol{Y} = \langle Y_1, Y_2, \ldots Y_m \rangle$

Output: The enhanced re-perturbed data $\hat{\boldsymbol{Y}}$

1 Construct ordered budget group data $EBGD = \langle BGD_1, BGD_2, \ldots BGD_m \rangle$ sorted by $\tilde{\boldsymbol{\epsilon}}$ in ascending order;
2 Initialize $\hat{\boldsymbol{Y}} \leftarrow \emptyset$;
3 **for** $i \in [m]$ **do**
4 Initialize $\hat{Y}_i \leftarrow Y_i$;
5 **for** $j \in [i+1, m]$ **do**
6 Get re-pertubed value $Y'_j \leftarrow \text{RP}(Y_j, \tilde{\epsilon}_j, \tilde{\epsilon}_i)$ by Equation 8;
7 $\hat{Y}_i \leftarrow \hat{Y}_i \cup Y'_j$;
8 $\hat{\boldsymbol{Y}} \leftarrow \hat{\boldsymbol{Y}} \cup \hat{Y}_i$
9 **return** $\hat{\boldsymbol{Y}}$;

Theorem V.1. *For any two budget group data* $BGD_L = (\epsilon_L, U_L, Y_L)$ *and* $BGD_H = (\epsilon_H, U_H, Y_H)$ *under GPRR with* $\epsilon_L < \epsilon_H$*, the variance of* BGD_L *can be reduced by*

$$\frac{z_H}{z_L\,(z_L + z_H)} \left(\frac{dq_L(1-q_L)}{(p_L - q_L)^2} + \frac{1 - p_L - q_L}{p_L - q_L} \right)$$

through updating BGD_L *to* $BGD'_L = \big(\epsilon_L, U_L \cup U_H, Y_L \cup Y'_H\big)$*, where* Y'_H *is the re-perturbed output of* Y_H *with probability:*

$$\forall y' \in Y', \Pr\left[\text{RP}(y) = y'\right] = \begin{cases} \beta, & \text{if } y' = y, \\ \gamma, & \text{otherwize}, \end{cases} \tag{8}$$

where $\beta = \frac{dp_L - p_L + p_H - 1}{dp_H - 1}$ *and* $\gamma = \frac{p_H - p_L}{dp_H - 1}$*.*

Proof. Please refer to the detailed proof of Theorem V.1 in Appendix IX-D of our report [38]. □

Based on the re-perturbation method, we propose an enhanced utility approach called Personalized Utility Enhancement (PUE) shown in Algorithm 2. In the PUE process, privacy budgets, populations and obfuscated counts are sorted by the privacy budgets. PUE iterates over all the privacy budget ϵ_j and applies re-perturbation to the obfuscated counts whose privacy budgets are larger than ϵ_i (Line 6).

Example 3. *Assume the value domain is* $\boldsymbol{\omega} = \{0, 1\}$ *and there are* 5 *groups* $G_1, G_2, \ldots G_5$ *with sorted privacy budgets* $\tilde{\boldsymbol{\epsilon}} = \langle 0.2, 0.3, 0.5, 0.6, 0.8 \rangle$*. Assume the obfuscated statistics of groups* G_3*,* G_4 *and* G_5 *are* $\{40, 60\}, \{70, 30\}$ *and* $\{60, 50\}$*, where each group adopts GRR as its base mechanism. Assume the current enhanced group is* G_3 *(*$\tilde{\epsilon}_3 = 0.6$*). Then, PUE re-perturbs all users in* G_4 *and* G_5*. For* $u_1 \in G_4$ *with obfuscated data* $y_1 = 1$*, the re-perturbed data is* 1 *with probability* $\beta_{4,3} = \frac{dp_3 - p_3 + p_4 - 1}{dp_4 - 1} = 0.65$ *and* 0 *with probability* $\gamma_{4,3} = \frac{p_4 - p_3}{dp_4 - 1} = 0.35$*. For* $u_2 \in G_5$ *with* $y_2 = 0$*, the re-perturbed data is* 1 *with probability* $\gamma_{5,3} = \frac{p_5 - p_3 + p_4 - 1}{dp_5 - 1} = 0.18$ *and* 0 *with probability* $\beta_{5,3} = 0.82$*.*

C. PLDP Population Enhanced Solutions

Building upon the OPS and PUE methods, we further propose two enhanced solutions: PLDP Population Distribution Plus (PLPD$^+$) and PLDP Population Absorption Plus

Algorithm 3: PLDP Population Distribution Plus

Input: Available user set U_A, privacy requirement ($\boldsymbol{w}$, $\boldsymbol{\epsilon}$) of all Users U, data domain size d, historical data publication $(\hat{\boldsymbol{f}}_1, \hat{\boldsymbol{f}}_2, \ldots, \hat{\boldsymbol{f}}_{t-1})$

Output: $\hat{\boldsymbol{f}}_t$

1 Calculate the distinct privacy budget $\tilde{\boldsymbol{\epsilon}}$ with the statistic $\boldsymbol{g}$;
// sub-mechanism $\mathcal{M}_{s,t}$
2 Calculate the sampling size $\boldsymbol{z} \leftarrow \left\langle \left\lfloor \frac{n}{2w_1} \right\rfloor, \left\lfloor \frac{n}{2w_2} \right\rfloor, \ldots, \left\lfloor \frac{n}{2w_n} \right\rfloor \right\rangle$;
3 Get optimal population size and error $z_{opt}, err_{opt}, \boldsymbol{\epsilon}_{opt} \leftarrow \text{OPS}(\boldsymbol{z}, \boldsymbol{\epsilon}, d)$ by Algorithm 1;
4 Sample user set $U_{s,t}$, whose data is $D_{s,t}$ and privacy budget list is $\boldsymbol{\epsilon}'_{opt} \subseteq \boldsymbol{\epsilon}_{opt}$, from U_A with the size of z_{opt} and remove $U_{s,t}$ from U_A, i.e., $U_A = U_A \backslash U_{s,t}$;
5 $\tilde{\boldsymbol{\epsilon}}'_{opt} \leftarrow \text{Unique}(\boldsymbol{\epsilon}'_{opt})$;
6 Get report $\boldsymbol{Y}_{s,t} \leftarrow \text{PFO.P}(D_{s,t})$ with privacy budget list $\boldsymbol{\epsilon}'_{opt}$ from users in $U_{s,t}$;
7 Calculate the enhanced report $\hat{\boldsymbol{Y}}_{s,t} \leftarrow \text{PUE}(\tilde{\boldsymbol{\epsilon}}'_{opt}, \boldsymbol{Y}_{s,t})$ by Algorithm 2;
8 Get $\boldsymbol{h}_{s,t,k} \leftarrow \text{PFO.A}(\hat{\boldsymbol{Y}}_{s,t})$ for each group G_k;
9 Estimate $\hat{\boldsymbol{f}}_{s,t} \leftarrow \text{PFO.E}(\boldsymbol{h}_{s,t}, \tilde{\boldsymbol{\epsilon}}'_{opt})$;
10 Calculate $dis \leftarrow \frac{1}{d}\sum_{j=1}^{d}\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2 - \frac{1}{d}\sum_{j=1}^{d}\text{Var}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right]$;
// sub-mechanism $\mathcal{M}_{r,t}$
11 Calculate the optimal window size $w_{opt} \leftarrow n/(2z_{opt})$;
12 Calculate the optimal remaining population size $n_{r,opt} \leftarrow n/2 - \sum_{\tau=t-w_{opt}+1}^{t-1}|U_{r,\tau}|$;
13 Set the optimal number of potential publication users $n_{pp,opt} \leftarrow \lfloor n_{r,opt}/2 \rfloor$;
14 Sample a user set $U_{r,t}$ from U_A with the size of $|U_{r,t}| = n_{pp,opt}$ and privacy budget list $\boldsymbol{\epsilon}''_{opt} \subseteq \boldsymbol{\epsilon}_{opt}$;
15 Calculate the budget group user statistic $\boldsymbol{g}_{pp,opt}$ of $\tilde{\boldsymbol{\epsilon}}''_{opt} = \text{Unique}(\boldsymbol{\epsilon}''_{opt})$ by $U_{r,t}$;
16 Calculate the potential reporting error $err \leftarrow err(\tilde{\boldsymbol{\epsilon}}''_{opt}, \boldsymbol{g}_{pp,opt}, n, n_{pp,opt}, d)$ by Equation (7);
17 **if** $dis > err$ *and* $n_{pp,opt} \geq$ **then**
18 Remove $U_{r,t}$ from U_A, i.e., $U_A \leftarrow U_A \backslash U_{r,t}$;
19 Get report $\boldsymbol{Y}_{r,t} \leftarrow \text{PFO.P}(D_{r,t})$ with privacy budget list $\boldsymbol{\epsilon}''_{opt}$ from users in $U_{r,t}$;
20 Calculate the enhanced report $\hat{\boldsymbol{Y}}_{r,t} \leftarrow \text{PUE}(\tilde{\boldsymbol{\epsilon}}''_{opt}, \boldsymbol{Y}_{r,t})$;
21 Get $\boldsymbol{h}_{r,t,k} \leftarrow \text{PFO.A}(\hat{\boldsymbol{Y}}_{r,t})$ for each group G_k;
22 Calculate $\hat{\boldsymbol{f}}_{r,t} \leftarrow \text{PFO.E}(\boldsymbol{h}_{r,t}, \tilde{\boldsymbol{\epsilon}}''_{opt})$;
23 **else**
24 Set $\hat{\boldsymbol{f}}_{r,t} \leftarrow \hat{\boldsymbol{f}}_{r,t-1}$;
25 **if** $t \geq w_{opt}$ **then**
26 $U_A \leftarrow U_A \cup U_{s,t-w_{opt}+1} \cup U_{r,t-w_{opt}+1}$;
27 **return** $\hat{\boldsymbol{f}}_{r,t}$.

(PLPA$^+$). The key idea behind these methods is to calculate the optimal sampling population while maintaining privacy protection through OPS, and to enhance utility by improving personalized responses across all users using PUE.

PLDP Population Distribution Plus. Algorithm 3 shows the process of PLPD$^+$. In PLPD$^+$, the population is treated as a resource and divided among the slots in each window. It is further split into two subsets, $U_{s,t}$ and $U_{r,t}$, for calculations in two sub-mechanisms: $\mathcal{M}_{s,t}$ and $\mathcal{M}_{r,t}$. $\mathcal{M}_{s,t}$ consumes $U_{s,t}$ to calculate dissimilarity *dis*, which is compared with the current error *err* to decide whether a new obfuscated statistic should be published. $\mathcal{M}_{s,t}$ uses $U_{r,t}$ to publish a new obfuscated data when $dis > err$.

For the process of $\mathcal{M}_{s,t}$, PLPD$^+$ first calculates the population sampling size list $\boldsymbol{z}$ for all users (Line 2). Using these personalized sampling sizes, PLPD$^+$ applies OPS to obtain the optimal size z_{opt} while ensuring privacy protection through budget division (Line 3). Next, it samples $U_{s,t}$ from the available user set U_A (Line 4) and applies the PFO protocol to calculate the dissimilarity (Line 6–10). Unlike PLPD, PLPD$^+$ applies PUE to improve the utility of disturbed data from PFO (Line 7).

For the process of $\mathcal{M}_{r,t}$, PLPD$^+$ calculates the remaining population resource $n_{r,opt}$ for new publication within the window w_{opt} (Line 12). It reserves half of $n_{r,opt}$ (i.e., $n_{pp,opt}$) for the new publication and leaves the other half for future use (Line 13). Based on the reserved population, it calculates the error *err* for the new publication (Line 16). Unlike traditional LDP setting [3], here, PLPD$^+$ samples the population for the new publication before the dissimilarity-error comparison to improve the accuracy of the error calculation without introducing additional privacy leakage (Line 14). Then, it calculates the current error *err* according to Equation (7) (Line 15–16) and compares it with the dissimilarity *dis* (Line 17–24). If $dis > err$, it indicates that the difference between the current and previous estimation is large enough to warrant a new publication. In this case, PLPD$^+$ removes the sampled $U_{r,t}$ from the candidate population (Line 18) and applies enhanced PFO to generate a new estimation $\hat{\boldsymbol{f}}_{r,t}$ (Line 19–22). If $dis \leq err$, PLPD$^+$ discards the sampled $U_{r,t}$ and approximates the current estimation as the one at the previous time slot (Line 24). Finally, PLPD$^+$ recycles the population at the $w_{opt} - 1$ time slot (Line 26).

We give an example for the procedure of PLPD$^+$ in Example 4.

Example 4. *Consider a scenario with* 5 *locations* $\{A, B, C, D, E\}$ *and* 2000 *users* $u_1, u_2, \ldots, u_{2000}$*, each with privacy requirements from* $\{0.2, 0.4, 0.6, 0.8\}$ *and window size requirements from* $\{1, 2, 3, 4\}$*. The distinct privacy budget list is* $\tilde{\boldsymbol{\epsilon}} = \langle 0.2, 0.4, 0.6, 0.8\rangle$ *and corresponding probability lists are* $\tilde{\boldsymbol{q}} = \langle 0.19, 0.18, 0.17, 0.16\rangle$ *and* $\tilde{\boldsymbol{p}} = \langle 0.23, 0.27,\ 0.31, 0.36\rangle$*. Table III(a) shows the user count of distinct window size-privacy budget pair. The corresponding statistic list of* $\tilde{\boldsymbol{\epsilon}}$ *is* $\boldsymbol{g} = \langle 0.1305, 0.3645, 0.244, 0.261\rangle$*. Based on the data above, the distinct sampling size list is* $\tilde{\boldsymbol{z}} = \langle 1000, 500, 333, 250\rangle$*. By executing OPS, the errors for these four sampling sizes are* $\boldsymbol{err} = \langle 0.042, 0.035, 0.0329, 0.0349\rangle$*. Thus, the optimal sampling size is chosen as* $z_{opt} = 333$*. Besides, the privacy budgets for users with window size* $\tilde{w}_4 = 4$ *are divided into* $\lceil \tilde{z}_3/\tilde{z}_4 \rceil = \lceil 333/250 \rceil = 2$ *shares, each having values* $0.1, 0.2, 0.3, 0.4$*, respectively. Table III(b) shows the user counts for the new budget-window sizes.*

At time stamp 1*, the available user set* U_A *is initially set to* U*. PLPD*$^+$ *samples* $z_{opt} = 333$ *users, de-*

noted as $U_{s,1} = \{u_1, u_2, \ldots, u_{333}\}$. *After sampling,* U_A *is updated as* $U_A = U_A \backslash U_{s,1} = \{u_{334}, u_{335}, \ldots, u_{2000}\}$. *Users in* $U_{s,1}$ *report their obfuscated locations using* PFO.P. *Assume the real statistic of* $U_{s,1}$ *is* $\boldsymbol{f}_{s,1} = \langle 0.183, 0.237, 0.174, 0.228, 0.177\rangle$. *After applying* PFO.P, *the perturbed statistic can be* $\tilde{\boldsymbol{f}}_{s,1} =< 0.204, 0.186, 0.189, 0.204, 0.216 >$. *After applying* PUE *and* PFO.A, *the user count of* U_A *increases to* 1356. *The re-perturbed statistic becomes* $\tilde{\boldsymbol{h}}_{s,1} = \langle 0.204, 0.198, 0.190, 0.198, 0.209\rangle$. *Using* PFO.E, *the estimation is calculated as* $\hat{\boldsymbol{f}}_{s,1} = \langle 0.301, 0.194, -0.047, 0.336, 0.215\rangle$ *and normalized by Simplex Projection as* $\langle 0.29, 0.182, 0, 0.324, 0.204\rangle$, *where* $\boldsymbol{\alpha} = \langle 0.015, 0.062, 0.117, 0.207, 0.286, 0.313\rangle$ *for distinct privacy budget list* $\tilde{\boldsymbol{\epsilon}} = \langle 0.1, 0.2, 0.3, 0.4, 0.6, 0.8\rangle$. *The variance sum is* $\sum_{j=1}^{d} \mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,1}[j]\right] = 0.085$. *The dissimilarity is calculated as* $dis = 0.041$. *The potential publication user number is* $n_{pp,opt} = \lfloor \frac{n}{2}/2 \rfloor = 500$. *Thus, we calculate the reporting error as* $err = 0.021$. *Since* $dis > err$, *the system continues to sample* 500 *users from* U_A *for a new publication. Assume the sampling set is* $U_{r,1} = \{u_{334}, u_{335}, \ldots, u_{833}\}$. *These users report their obfuscated locations using* PFO.P. *Assume the real statistic of* $U_{r,1}$ *is* $\boldsymbol{f}_{r,1} = \langle 0.2, 0.206, 0.196, 0.182, 0.216\rangle$, *and the perturbed statistic is* $\tilde{\boldsymbol{f}}_{r,1} = \langle 0.206, 0.19, 0.194, 0.23, 0.18\rangle$. *Assume the obfuscated frequency is* $\boldsymbol{h}_{r,1} = \langle 0.23, 0.19, 0.19, 0.21, 0.18\rangle$. *After re-perturbing, we get* $\tilde{\boldsymbol{h}}_{r,1} = \langle 0.209, 0.196, 0.208, 0.201, 0.187\rangle$. *The final estimation is* $\hat{\boldsymbol{f}}_{r,1} = \langle 0.21, 0.108, 0.322, 0.281, 0.079\rangle$, *where* $\boldsymbol{\alpha} = \langle 0.016, 0.064, 0.12, 0.21, 0.28, 0.313\rangle$ *for distinct privacy budget list* $\tilde{\boldsymbol{\epsilon}} = \langle 0.1, 0.2, 0.3, 0.4, 0.6, 0.8\rangle$. *Because there are no users for historical publications outside the window size* w_{max} *(i.e.,* $t < w_{max}$*),* U_A *remain unchanged.*

At time stamp 2, 333 *users are still sampled from* U_A *for dis calculation. Assume these* 333 *users are* $U_{s,2} = \{u_{834}, u_{835}, \ldots, u_{1166}\}$ *and* $dis = 0.041$. *Then the available user set is updated as* $U_A = U_A \backslash U_{s,2} = \{u_{1167}, u_{1168}, \ldots, u_{2000}\}$. *PLPD*$^+$ *calculates the remaining population size* $n_{r,opt} = 2000/2 - 500 = 500$. *Next, it calculates the potential publication user number* $n_{pp,opt} = \lfloor n_{r,opt}/2 \rfloor = 250$. *It samples* $n_{pp,opt} = 125$ *candidate users for a new publication and calculates the reporting error as* $err = 0.05$. *Since* $dis \leq err$, *it aborts the new publication and approximates the publication as* $\hat{\boldsymbol{f}}_{r,2} = \hat{\boldsymbol{f}}_{r,1}$.

Assume there is a new publications at time stamp 3, *then the sampling population consumption is still* 333, *and the new publication population consumption is* $\lfloor (n/2 - 500 - 0)/2 \rfloor = 250$. *Thus, the remaining population is* $|U_A| = 251$ *after the new publication. Assume* $U_A = \{u_{1750}, u_{1751}, \ldots, u_{2000}\}$. *Since the current times lot* $t = 3 \geq w_{opt}$, *PLPD*$^+$ *recycles the population consumed at time stamp* 1. *Specifically,* $U_A = U_A \cup U_{s,1} \cup U_{r,1} = \{u_1, u_2, \ldots, u_{833}, u_{1750}, u_{1751}, \ldots, u_{2000}\}$, *with a size of* $|U_A| = 1084$.

The process repeats similarly for subsequent time stamps.

TABLE III: An example for the user counts of window size-privacy budget pairs.

(a) The original data counts

ϵ \ w	1	2	3	4
0.2	29	59	92	56
0.4	98	176	273	207
0.6	62	124	188	114
0.8	64	115	202	141

(b) The optimal new data counts

ϵ \ w	1	2	3	4
0.1	0	0	0	56
0.2	29	59	92	207
0.3	0	0	0	114
0.4	98	176	273	141
0.6	62	124	188	0
0.8	64	115	202	0

PLDP Population Absorption Plus. PLPD$^+$ reserves half of the remaining population for future publications, making it suitable for slow, gradual changes in data streams. However, when the data steam encounters occasional but sudden, PLPD$^+$ may fail to provide accurate estimation. To address this limitation, we propose PLDP Population Absorption Plus (PLPA$^+$) as follows.

Algorithm 4: PLDP Population Absorption Plus

Input: Available user set U_A, privacy requirement $(\boldsymbol{w}, \boldsymbol{\epsilon})$ of all Users U, data domain size d, historical data publication $(\hat{\boldsymbol{f}}_1, \hat{\boldsymbol{f}}_2, \ldots, \hat{\boldsymbol{f}}_{t-1})$
Output: $\hat{\boldsymbol{f}}_t$

Calculate the distinct privacy budget $\tilde{\boldsymbol{\epsilon}}$ with the statistic $\boldsymbol{g}$;
// sub-mechanism $\mathcal{M}_{s,t}$
The same as Line 2-10 in Algorithm 3;
// sub-mechanism $\mathcal{M}_{r,t}$
Calculate the optimal window size $w_{opt} \leftarrow n/(2z_{opt})$;
Calculate the time slots to be nullified $t_N = \frac{|U_{r,l}|}{z_{opt}} - 1$;
if $t - l \leq t_N$ **then**
  $\hat{\boldsymbol{f}}_t \leftarrow \hat{\boldsymbol{f}}_{t-1}$;
else
  Calculate time slots that can be absorbed $t_A = t - l - t_N$;
  Set the number of potential publication users $n_{pp,opt} \leftarrow z_{opt} \cdot \min(t_A, w_{opt})$;
  Sample a user set $U_{r,t}$ from U_A with the size of $|U_{r,t}| = n_{pp,opt}$ and privacy budget list $\boldsymbol{\epsilon}''_{opt} \subseteq \boldsymbol{\epsilon}_{opt}$
  Calculate the budget group user statistic $\boldsymbol{g}_{pp,opt}$ of $\tilde{\boldsymbol{\epsilon}}''_{opt} = \mathrm{Unique}(\boldsymbol{\epsilon}''_{opt})$ by $U_{r,t}$.
  Calculate the potential reporting error $err \leftarrow err(\tilde{\boldsymbol{\epsilon}}''_{opt}, \boldsymbol{g}_{pp,opt}, n, n_{pp,opt}, d)$ by Equation (7);
  if $dis > err$ **then**
    Remove $U_{r,t}$ from U_A, i.e., $U_A \leftarrow U_A \backslash U_{r,t}$;
    Get report $\boldsymbol{Y}_{r,t} \leftarrow \mathrm{PFO.P}(D_t)$ with privacy budget list $\boldsymbol{\epsilon}''_{opt}$ from users in $U_{r,t}$;
    Calculate the enhanced report $\hat{\boldsymbol{Y}}_{r,t} \leftarrow \mathrm{PUE}(\tilde{\boldsymbol{\epsilon}}''_{opt}, \boldsymbol{Y}_{r,t})$;
    Get $\boldsymbol{h}_{r,t,k} \leftarrow \mathrm{PFO.A}(\hat{\boldsymbol{Y}}_{r,t})$ for each group G_k;
    Calculate $\hat{\boldsymbol{f}}_{r,t} \leftarrow \mathrm{PFO.E}(\boldsymbol{h}_{r,t}, \tilde{\boldsymbol{\epsilon}}''_{opt})$;
  else
    Set $\hat{\boldsymbol{f}}_{r,t} \leftarrow \hat{\boldsymbol{f}}_{r,t-1}$;
if $t \geq w_{opt}$ **then**
  $U_A \leftarrow U_A \cup U_{s,t-w_{opt}+1} \cup U_{r,t-w_{opt}+1}$;
return $\hat{\boldsymbol{f}}_{r,t}$.

The main idea of PLPA$^+$ is to pre-assign an equal population share for each time slot and absorb the unused population from previous time slots, meanwhile nullifying the population

for future time slots. This approach enhances the utility of the current publication. The process of PLPA$^+$ is shown in Algorithm 4.

PLPA$^+$ also consists of two sub-mechanisms: $\mathcal{M}_{s,t}$ and $\mathcal{M}_{r,t}$, where $\mathcal{M}_{s,t}$ calculates the dissimilarity *dis* and $\mathcal{M}_{r,t}$ publishes new estimations. The mechanism $\mathcal{M}_{s,t}$ in PLPA$^+$ is the same as in Algorithm 3.

For $\mathcal{M}_{r,t}$, PLPA$^+$ calculates the time length t_N nullified by the last time slot t_l (Line 4), and checks whether the current time slot is within this period t_N. If it is, the current publication is approximated by the previous one (Line 6). Otherwise, PLPA$^+$ compares *dis* and *err* to decide whether to publish a new obfuscated estimation (Line 8–20).

Specifically, PLPA$^+$ calculates the time slot length t_A out of the nullified period t_N but not after the current time slot, determining the maximum length that can be absorbed by the current time slot. Then, PLPA$^+$ calculates the population size $n_{pp,opt}$ that can be absorbed (Line 9), samples $n_{pp,opt}$ users $U_{r,t}$, and calculates the corresponding error *err* (Line 10–11). If *dis* $>$ *err*, PLPA$^+$ updates the population candidate set U_A (Line 14) and applies PFO enhanced by PUE to publish a new estimation (Line 15–18). Otherwise, it approximates the current publication as the previous one (Line 20). Finally, PLPA$^+$ recycles the population outside the window size w_{opt} (Line 22).

The process of PLPA$^+$ is illustrated in Example 5.

Example 5. *Consider the data in Example 4. At time slot* 1*, assume PLPA$^+$ consumes population* $\{u_1, u_2 \dots u_{333}\}$ *to calculate the dissimilarity dis. It calculates the nullified time length* $t_N = -1$*. Since* $t - l = 1 > t_N$*, PLPA$^+$ compares dis with err to decide whether to publish a new estimation. Specifically, it calculates* $t_A = 2$ *and sets* $n_{pp,opt} = 333 \times 2 = 666$*. After sampling* 666 *users* $U_{r,1}$*, it calculates the error err. Assume* $U_{r,1} = \{u_{334}, u_{335}, \dots u_{999}\}$ *and dis* $>$ *err. Then PLPA$^+$ publishes a new estimation* $\hat{\boldsymbol{f}}_{r,1}$ *using enhanced* PFO*.*

At time slot 2*, dis calculation still consumes* $z_{opt} = 333$ *users* $U_{s,2} = \{u_{1000}, u_{1001}, \dots u_{1332}\}$*. PLPA$^+$ calculates the nullified time length* $t_N = 666/333 - 1 = 1$*. Since* $t - l = 1 \leq t_N$*, the current estimation* $\hat{\boldsymbol{f}}_{r,2}$ *is approximated as* $\hat{\boldsymbol{f}}_{r,1}$*.*

At time slot 3*, assume there is a new estimation publication. Then it consumes* 333 *users* $U_{s,3} = \{u_{1333}, u_{1334}, \dots, u_{1665}\}$ *for dis calculation. Here,* $t_N = 666/333 - 1 = 1$ *and* $t_A = 3 - 1 - 1 = 1$*. Thus,* $n_{pp,opt}$ *is set to* 333*. After sampling population* $U_{r,3} = \{u_{1666}, u_{1667}, \dots, u_{1998}\}$*, PLPA$^+$ recycles* $U_{s,1}$ *and* $U_{r,1}$*, updating the candidate population set as* $U_A = \{u_1, u_2, \dots, u_{999}, u_{1999}, u_{2000}\}$*.*

The process continues similarly for subsequent time slots.

D. Analysis

Time Complexity Analysis. Let d denote the value domain size, m the number of budget groups, and n the total number of users, satisfying $m \leq n$. Let z_{opt} be the optimal sampling size at each time slot, with $z_{opt} \leq n/2$. We analyze the time complexity of PLPD and PLPA as follows.

Theorem V.2. *The time complexity of both PLPD$^+$ and PLPA$^+$ is* $O(n \cdot m + d \cdot m)$*.*

Proof. Please refer to the detailed proof of Theorem V.2 in Appendix IX-E2 of our report [38]. □

Privacy Analysis. When the optimal window size w_{opt} is determined, each user in both PLPD$^+$ and PLPA$^+$ appears at most once within any w_{opt} size window. For any user u_i with a personalized window size requirement $w_i > w_{opt}$, the privacy budget ϵ_i is divided into $\lceil w_i/w_{opt} \rceil$ portions, ensuring the total privacy budget consumption within a w_i-length window does not exceed ϵ_i. In addition, each estimation is published through the PFO protocol under $\boldsymbol{\epsilon}$-PLDP. Therefore, the following theorem holds for the privacy guarantee of both methods.

Theorem V.3. *PLPD$^+$ and PLPA$^+$ satisfy* $(\boldsymbol{w}, \boldsymbol{\epsilon})$*-EPLDP.*

Proof. Please refer to the detailed proof of Theorem V.3 in Appendix IX-F2 of our report [38]. □

Utility Analysis. For simplicity, let w_{opt} denote the optimal window size from the OPS process. We assume that at most $s_{opt} < w_{opt}$ new publications occur at time slots $\rho_1, \rho_2, \dots \rho_{opt}$, without budget absorption from any past time slots outside the current window. Besides, we assume each new publication corresponds to the same count of skipped / nullified publications. Let $\tilde{\boldsymbol{\epsilon}}_{opt}$ denote the new privacy budget list after applying OPS, and $\tilde{\boldsymbol{n}}_{opt}$ represent the user count list, where $\tilde{\boldsymbol{n}}_{opt}(k)$ indicates the number of users requiring privacy budgets $\tilde{\boldsymbol{\epsilon}}_{opt}(k)$. Let $\hat{n}_{max(\tilde{\boldsymbol{\epsilon}}_{opt})}$ denote the number of users requiring the maximum privacy budget in $\hat{\boldsymbol{\epsilon}}_{opt}$. Under these definitions, we present the following two theorems for PLPD$^+$ and PLPA$^+$, respectively.

Theorem V.4. *The error upper bound of per time slot in PLPD$^+$ is* $\frac{1}{d^3}\left(\frac{2w_{opt}}{n-1} - \frac{1}{n-1} + 2w_{opt}A\right)^2 + \frac{4(2^{s_{opt}}-1)}{s_{opt}} \cdot \left(\frac{1}{n-1} + A\right) - \frac{1}{n-1}$*, where* $A = \frac{d-1}{\hat{n}_{\max(\tilde{\boldsymbol{\epsilon}}_{opt})}} \cdot \frac{2e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})} + d - 2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})} - 1\right)^2}$*.*

Proof. Please refer to the detailed proof of Theorem V.4 in Appendix IX-G2 of our report [38]. □

Theorem V.5. *The error upper bound of per time slot in PLPA$^+$ is* $\frac{1}{d^3}\left(\frac{2w_{opt}}{n-1} - \frac{1}{n-1} + 2w_{opt}A\right)^2 + BC \cdot err_{nlf} - \frac{(B+1)C}{n-1} + \left(\frac{s_{opt}}{n-1} + s_{opt} \cdot A\right) \cdot \left(2 - \frac{1}{2^{B+1}} + \frac{2}{B}\right)$*, where* $A = \frac{d-1}{\hat{n}_{\max(\tilde{\boldsymbol{\epsilon}}_{opt})}} \cdot \frac{2e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})} + d - 2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})} - 1\right)^2}$*,* $B = \frac{w_{opt} - s_{opt}}{2s_{opt}}$ *and* $C = \frac{s_{opt}}{w_{opt}}$*.*

Proof. Please refer to the detailed proof of Theorem V.5 in Appendix IX-G2 of our report [38]. □

VI. Experiments

In this section, we conduct extensive experiments to evaluate the performance of our proposed mechanisms. We compare them against state-of-the-art non-personalized LDP methods for data steams [3]. In addition, to assess the effectiveness of the key components OPS and PUE, we perform an ablation study by selectively removing each module and replacing it with its corresponding baseline mechanism.

TABLE IV: Experimental settings.

Parameters	Values
static privacy budget ϵ	$0.5, 1.0, \mathbf{1.5}, 2.0, 2.5$
static window size w	$40, 80, \mathbf{120}, 160, 200$
personalized privacy budget ϵ_i	$\epsilon, \ldots, 2.0, 2.5$
personalized window size w_i	$40, 80, \ldots, w$
population lower bound n_{min}	10

A. Datasets

Real Datasets. We employ two real-world datasets, namely *Taxi* and *Foursquare*, to evaluate the performance of our PLPD$^+$ and PLPA$^+$.

Taxi. This dataset records real-time trajectories of $10,357$ taxis in Beijing from February 2 to February 8, 2008. After cleaning, $14,859,377$ valid records from $n = 10,269$ taxis are retained, locations are mapped into $d = 5$ areas. We sample the trajectories every minute, resulting in $T = 8,889$ time slots.

Foursquare. This dataset consists of $33,278,683$ Foursquare check-ins from $266,909$ users collected between April 2012 and September 2013. We map all venues into $d = 5$ discrete categories and convert the data into $n = 13,216$ data streams with $T = 7,649$ time slots.

Synthetic Datasets. We generate three synthetic binary data streams based on different probabilistic sequence models to evaluate our mechanisms under controlled conditions. Each stream involves $n = 10,000$ users over $T = 10,000$ time slots. At each time step t, each user's binary value equals 1 with probability p_t (and 0 otherwise), where $(p_1, p_2, \ldots, p_T)$ follows the generation rules defined below.

TLNS. In this dataset, the probability evolves dynamically as $p_t = p_{t-1} + \mathcal{N}(0, Q)$, where $\mathcal{N}(0, Q)$ denotes Gaussian noise with variance Q and standard deviation $\sqrt{Q} = 0.0025$. The initial probability is $p_0 = 0.05$, and all p_t values are clipped to the range $[0, 1]$ (i.e., set $p_t = 0$ if $p_t < 0$ and $p_t = 1$ if $p_t > 1$) to ensure validity.

Sin. In the Sin dataset, the probability follows a sinusoidal trend, defined as $p_t = A\sin(\omega t) + h$, where $A = 0.05$, $\omega = 0.01$, and $h = 0.075$.

Log. In the Log dataset, the probability follows a logistic growth pattern, defined as $p_t = \dfrac{A}{1 + e^{-bt}}$, where $A = 0.25$ and $b = 0.01$.

B. Experiment Setup

Compared Algorithms. We compare the enhanced personalized methods, PLPD$^+$ and PLPA$^+$, with their basic counterparts: PLPD and PLPA. Additionally, we also compare the above four personalized solutions with two non-personalized methods: LDP Budget Distribution (LBD) and LDP Budget Absorption (LBA) [3]. Each experiment is repeated ten times using different random seeds ranging from 0 to 9, and we report the average results.

Parameter Settings. In non-personalized settings (i.e., LBD and LBA), ϵ and w represent the privacy budget and window size, respectively. We vary ϵ from 0.5 to 1.5 and w from 40 to 200. Following the Reference [4], we set the lower bound of all users' personalized privacy budgets to ϵ and the upper bound of their personalized window sizes to w.

Table IV summaries all experimental parameters, where default values are highlighted in bold. All experiments are conducted in Java on an Intel(R) Xeon(R) Silver 4210R CPU @ 2.4GHz with 128 RAM.

Performance Metrics. We evaluate each method in terms of running time and data utility. Data utility is measured as the *Average Mean Square Error* (*AMSE*) and the *Average Jensen-Shannon Divergence* [39] (*AJSD*, $\bar{D}_{JS}$), defined as follows.

$$\textit{AMSE} = \frac{1}{T}\sum_{\tau=1}^{T} \textit{MSE}_\tau = \frac{1}{dT}\sum_{\tau=1}^{T}\sum_{j=1}^{d}\left(\hat{f}_{j,\tau} - f_{j,\tau}\right)^2.$$

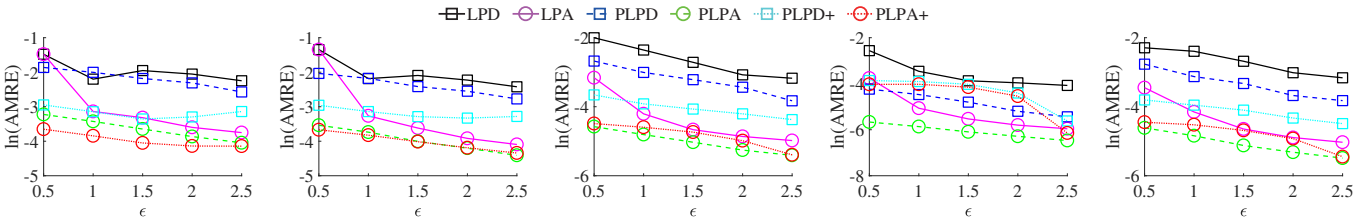
$$\begin{aligned}\bar{D}_{JS}(\hat{\boldsymbol{f}}\|\boldsymbol{f}) &= \frac{1}{T}\sum_{\tau=1}^{T} D_{JS}(\hat{\boldsymbol{f}}\|\boldsymbol{f}) = \frac{1}{T}\sum_{\tau=1}^{T}\left(\frac{1}{2}D_{KL}(\hat{\boldsymbol{f}}\|\boldsymbol{v}) + \frac{1}{2}D_{KL}(\boldsymbol{f}\|\boldsymbol{v})\right)\\ &= \frac{1}{2T}\sum_{\tau=1}^{T}\sum_{j=1}^{d}\left(\hat{\boldsymbol{f}}_{j,\tau}\log\left(\frac{\hat{\boldsymbol{f}}_{j,\tau}}{\boldsymbol{v}_{j,\tau}}\right) + \boldsymbol{f}_{j,\tau}\log\left(\frac{\boldsymbol{f}_{j,\tau}}{\boldsymbol{v}_{j,\tau}}\right)\right),\end{aligned}$$

where $\boldsymbol{v}_{j,\tau} = \frac{1}{2}\left(\hat{\boldsymbol{f}}_{j,\tau} + \boldsymbol{f}_{j,\tau}\right)$.

C. Overall Utility Analysis

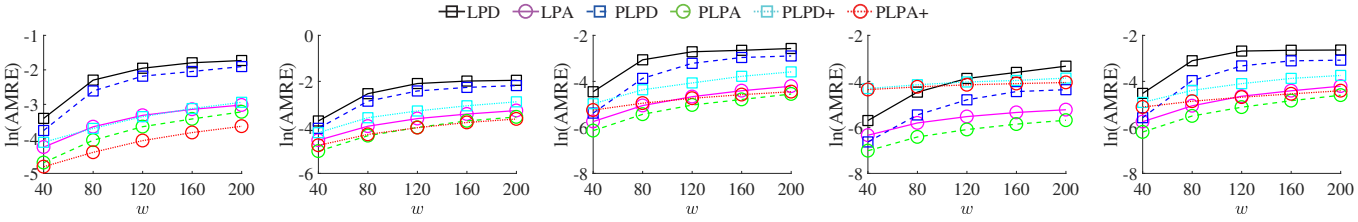
Figure 2 presents the publication accuracy of all methods as the privacy budget ϵ varies. In general, the logarithmic error $\ln(\textit{AMSE})$ of all methods decreases as ϵ increases, reflecting the trade off between privacy and data utility. The absorption-based methods (i.e., LBA, PLPA, PLPA$^+$) consistently outperform their corresponding distribution-based counterparts (i.e., LBD, PLPD, PLPD$^+$). This is because the distribution-based methods distribute population in an exponentially decaying way in the window, leading to quite larger estimation error. More over, our personalized mechanisms outperform the corresponding non-personalized ones in most datasets as ϵ increases, demonstrating the benefits of personalized methods. Additionally, our PLPA$^+$ yields the best overall performance on real datasets, with an average error 47.7% lower than LBA, while our PLPA performs best on synthetic datasets, reducing the average error by 47.27% compared with LBA. This difference arises because when the data dimension d is small, the advantages of OPS and PUE become less pronounced, leading PLPA to outperform PLPA$^+$ in low-dimensional synthetic datasets.

Figure 3 shows the publication accuracy as the window size w increases. Overall, $\ln(\textit{AMSE})$ increases with larger w, since a wider window reduces the number of users contributing to each time slot. Similar to the previous trend, absorption-based methods (i.e., LBA, PLPA, PLPA$^+$) still outperform distribution-based ones (i.e., LBD, PLPD, PLPD$^+$) for the same reason—exponentially decaying population assignment results in higher estimation error. More over, our personalized mechanisms outperform their non-personalized counterparts, validating the benefit of personalization. Consistent with the results on ϵ, our PLPA$^+$ achieves the best accuracy on real datasets, with an average error 38.65% lower than LBA, whereas our PLPA excels on synthetic datasets, reducing the average error by 36.5% compared with LBA.



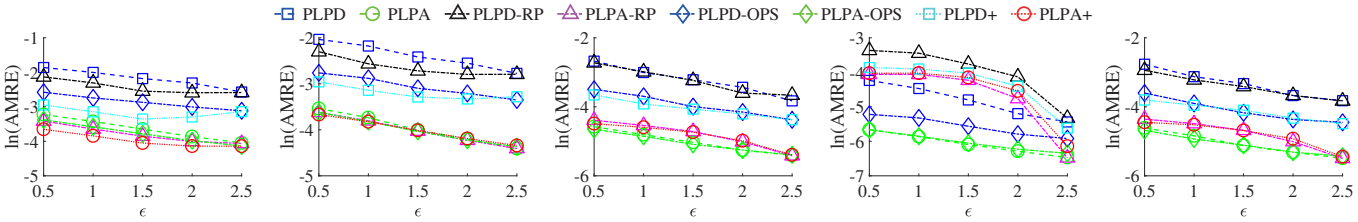
(a) Taxi, $w = 120$ (b) Foursquare, $w = 120$ (c) TLNS, $w = 120$ (d) Sin, $w = 120$ (e) Log, $w = 120$

Fig. 2: $AMSE$ with different ϵ.



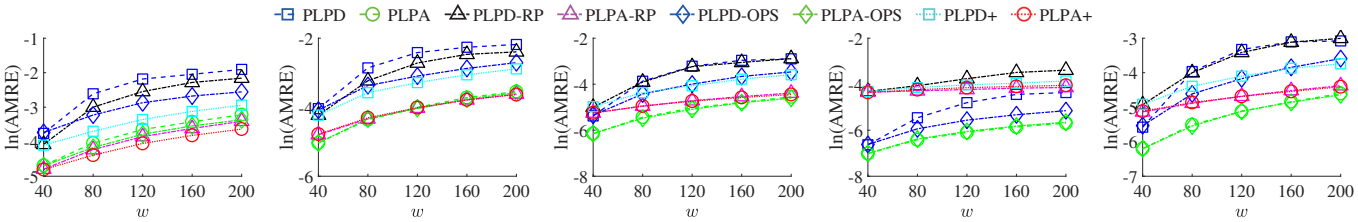
(a) Taxi, $\epsilon = 1.5$ (b) Foursquare, $\epsilon = 1.5$ (c) TLNS, $\epsilon = 1.5$ (d) Sin, $\epsilon = 1.5$ (e) Log, $\epsilon = 1.5$

Fig. 3: $AMSE$ with different w.



(a) Taxi (b) Foursquare, $w = 120$ (c) TLNS, $w = 120$ (d) Sin, $w = 120$ (e) Log, $w = 120$

Fig. 4: The ablation study of $AMSE$ with different ϵ.



(a) Taxi, $\epsilon = 1.5$ (b) Foursquare, $\epsilon = 1.5$ (c) TLNS, $\epsilon = 1.5$ (d) Sin, $\epsilon = 1.5$ (e) Log, $\epsilon = 1.5$

Fig. 5: The ablation study of $AMSE$ with different w.

D. Ablation Study

To evaluate the contributions of our sub-modules OPS and PUE, we conduct an ablation study by selectively removing them. We construct four ablated variants: PLBD-RP, PLBA-RP, PLBD-OPS and PLBA-OPS. Specifically, PLBD-RP and PLBA-RP are the enhanced population methods without OPS, while PLBD-OPS and PLBA-OPS are the enhanced population methods without PUE.

Figure 4 and Figure 5 compare the accuracy of these ablated methods with our complete solutions as ϵ and w vary, respectively. As shown, all methods exhibit similar trends: the error decreases with larger ϵ and increase with larger w. Besides, the inclusion of OPS yields substantial accuracy gains over the basic population methods in most cases and maintains comparable performance otherwise. Similarly, PUE also improves accuracy, especially on real datasets where heterogeneity among users is more evident. When the data dimension decreases to $d = 2$, however, the advantage of PUE diminishes or even reverses—particularly in datasets such as *Sin*, where the underlying data exhibit non-stationary variation rates.

VII. Conclusion

In this paper, we propose two basic methods PLPD and PLPA for Personalized Private Steaming Data Estimation in Local Setting. To further improve the utility, we propose two enhanced methods, PLPD$^+$ and PLPA$^+$. We evaluate these methods against recent non-personalized approaches, to demonstrate their efficiency and effectiveness.

VIII. AI-Generated Content Acknowledgement

No content of this paper is generated by Generative AI tools and technologies, such as ChatGPT.

References

[1] M. Feijoo-Cid, A. Arreciado Marañón, A. Huertas, A. Rivero-Santana, C. Cesar, V. Fink, M. I. Fernández-Cano, and O. Sued, "Exploring the decision-making process of people living with hiv enrolled in antiretroviral clinical trials: A qualitative study of decisions guided by trust and emotions," *Health Care Analysis*, vol. 31, no. 3, pp. 135–155, 2023.

[2] C. Dwork, "Differential privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, vol. 4052, pp. 1–12, Springer, 2006.

[3] X. Ren, L. Shi, W. Yu, S. Yang, C. Zhao, and Z. Xu, "LDP-IDS: local differential privacy for infinite data streams," in *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022* (Z. Ives, A. Bonifati, and A. E. Abbadi, eds.), pp. 1064–1077, ACM, 2022.

[4] L. Du, P. Cheng, L. Chen, H. T. Shen, X. Lin, and W. Xi, "Infinite stream estimation under personalized w-event privacy," *Proc. VLDB Endow.*, vol. 18, no. 6, pp. 1905 – 1918, 2025.

[5] Y. He, M. Wang, X. Deng, P. Yang, Q. Xue, and L. T. Yang, "Personalized local differential privacy for multi-dimensional range queries over mobile user data," *IEEE Transactions on Mobile Computing*, 2025.

[6] R. Bassily and A. D. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015* (R. A. Servedio and R. Rubinfeld, eds.), pp. 127–135, ACM, 2015.

[7] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014* (G. Ahn, M. Yung, and N. Li, eds.), pp. 1054–1067, ACM, 2014.

[8] T. Wang, J. Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, and S. Jha, "Continuous release of data streams under both centralized and local differential privacy," in *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pp. 1237–1253, 2021.

[9] E. Bao, Y. Yang, X. Xiao, and B. Ding, "CGM: an enhanced mechanism for streaming data collection with local differential privacy," *Proc. VLDB Endow.*, vol. 14, no. 11, pp. 2258–2270, 2021.

[10] Q. Xue, Q. Ye, H. Hu, Y. Zhu, and J. Wang, "DDRM: A continual frequency estimation mechanism with local differential privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6784–6797, 2023.

[11] Q. Ye, H. Hu, K. Huang, M. H. Au, and Q. Xue, "Stateful switch: Optimized time series release with local differential privacy," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications, New York City, NY, USA, May 17-20, 2023*, pp. 1–10, 2023.

[12] Y. Mao, Q. Ye, H. Hu, Q. Wang, and K. Huang, "Privshape: Extracting shapes in time series under user-level local differential privacy," in *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, pp. 1739–1751, IEEE, 2024.

[13] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proc. VLDB Endow.*, vol. 7, no. 12, pp. 1155–1166, 2014.

[14] M. Alaggan, S. Gambs, and A. Kermarrec, "Heterogeneous differential privacy," *J. Priv. Confidentiality*, vol. 7, no. 2, 2016.

[15] I. Kotsogiannis, S. Doudalis, S. Haney, A. Machanavajjhala, and S. Mehrotra, "One-sided differential privacy," in *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*, pp. 493–504, IEEE, 2020.

[16] D. Kifer and A. Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions," *ACM Trans. Database Syst.*, vol. 39, no. 1, pp. 3:1–3:36, 2014.

[17] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: tuning privacy-utility trade-offs using policies," in *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014* (C. E. Dyreson, F. Li, and M. T. Özsu, eds.), pp. 1447–1458, ACM, 2014.

[18] J. Liu, K. Knopf, Y. Tan, B. Ding, and X. He, "Catch a blowfish alive: A demonstration of policy-aware differential privacy for interactive data exploration," *Proc. VLDB Endow.*, vol. 14, no. 12, pp. 2859–2862, 2021.

[19] T. Edwards, B. I. P. Rubinstein, Z. Zhang, and S. Zhou, "A graph symmetrization bound on channel information leakage under blowfish privacy," *IEEE Trans. Inf. Theory*, vol. 68, no. 1, pp. 538–548, 2022.

[20] T. Nuradha and Z. Goldfeld, "Pufferfish privacy: An information-theoretic study," *IEEE Trans. Inf. Theory*, vol. 69, no. 11, pp. 7336–7356, 2023.

[21] C. Pierquin, A. Bellet, M. Tommasi, and M. Boussard, "Rényi pufferfish privacy: General additive noise mechanisms and privacy amplification by iteration via shift reduction lemmas," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, OpenReview.net, 2024.

[22] T. Nuradha, Z. Goldfeld, and M. M. Wilde, "Quantum pufferfish privacy: A flexible privacy framework for quantum systems," *IEEE Trans. Inf. Theory*, vol. 70, no. 8, pp. 5731–5762, 2024.

[23] N. Ding, "Approximation of pufferfish privacy for gaussian priors," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 5630–5640, 2024.

[24] H. Song, T. Luo, X. Wang, and J. Li, "Multiple sensitive values-oriented personalized privacy preservation based on randomized response," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2209–2224, 2020.

[25] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017* (E. Kirda and T. Ristenpart, eds.), pp. 729–745, USENIX Association, 2017.

[26] Z. Jorgensen, T. Yu, and G. Cormode, "Conservative or liberal? personalized differential privacy," in *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015* (J. Gehrke, W. Lehner, K. Shim, S. K. Cha, and G. M. Lohman, eds.), pp. 1023–1034, IEEE Computer Society, 2015.

[27] B. Niu, Y. Chen, B. Wang, Z. Wang, F. Li, and J. Cao, "Adapdp: Adaptive personalized differential privacy," in *40th IEEE Conference on Computer Communications, INFOCOM 2021, Vancouver, BC, Canada, May 10-13, 2021*, pp. 1–10, IEEE, 2021.

[28] S. Chaudhuri, K. Miagkov, and T. A. Courtade, "Mean estimation under heterogeneous privacy demands," *IEEE Trans. Inf. Theory*, vol. 71, no. 2, pp. 1362–1375, 2025.

[29] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pp. 289–300, IEEE Computer Society, 2016.

[30] X. Li, H. Yan, Z. Cheng, W. Sun, and H. Li, "Protecting regression models with personalized local differential privacy," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 2, pp. 960–974, 2023.

[31] T. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 3, pp. 26:1–26:24, 2011.

[32] T. Wang, N. Li, and S. Jha, "Locally differentially private heavy hitter identification," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 2, pp. 982–993, 2021.

[33] T. Wang, M. Lopuhaä-Zwakenberg, Z. Li, B. Skoric, and N. Li, "Locally differentially private frequency estimation with consistency," in *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*, The Internet Society, 2020.

[34] C. D. Kemp and A. W. Kemp, "Generalized hypergeometric distributions," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 18, no. 2, pp. 202–211, 1956.

[35] J. Liu, J. Lou, L. Xiong, J. Liu, and X. Meng, "Projected federated averaging with heterogeneous differential privacy," *Proc. VLDB Endow.*, vol. 15, no. 4, pp. 828–840, 2021.

[36] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 37–45, IEEE Computer Society, 2015.

[37] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, 2018.

[38] L. Du, Y. Hu, X. Zhou, Y. Cao, P. Cheng, W. Ni, and K. Li, "Pldp-ids: Personalized local differential privacy for infinite data stream [technical report]." http://cspcheng.github.io/pdf/WEventPLDP.pdf, 2025.

Algorithm 5: PLDP Population Distribution

Input: Available user set U_A, privacy requirement ($\boldsymbol{w}$, $\boldsymbol{\epsilon}$) of all Users U, dataset D_t data domain size d, historical data publication ($\hat{\boldsymbol{f}}_1, \hat{\boldsymbol{f}}_2, \ldots, \hat{\boldsymbol{f}}_{t\text{-}1}$)

Output: $\hat{\boldsymbol{f}}_t$

1 Calculate the distinct privacy budget $\tilde{\boldsymbol{\epsilon}}$ with the statistic $\boldsymbol{g}$;
2 Get the maximal window size $w_{max} = \max(\boldsymbol{w})$;
// sub-mechanism $M_{s,t}$
3 Calculate the minimal sampling size $z_{min} \leftarrow \lfloor n/(2w_{max}) \rfloor$;
4 Sample users $U_{s,t}$ from U_A with the size of z_{min} and remove $U_{s,t}$ from U_A, i.e., $U_A = U_A \backslash U_{s,t}$;
5 Get report $\boldsymbol{Y}_t \leftarrow$ PFO.P(D_t) with privacy budget list $\boldsymbol{\epsilon}$ from users in $U_{s,t}$;
6 Get the combination $\boldsymbol{h}_{s,t} \leftarrow$ PFO.A($\boldsymbol{Y}_t$);
7 Estimate $\hat{\boldsymbol{f}}_{s,t} \leftarrow$ PFO.E($\boldsymbol{h}_{s,t}, \boldsymbol{\epsilon}$);
8 Calculate $dis \leftarrow \frac{1}{d}\sum_{j=1}^{d}\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_l[j]\right)^2 - \frac{1}{d}\sum_{j=1}^{d}\text{Var}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right]$;
// sub-mechanism $M_{r,t}$
9 Calculate the minimal remaining population size $n_{r,min} = n/2 - \sum_{\tau=t-w_{max}+1}^{t-1}|U_{r,\tau}|$;
10 Set the minimal number of potential publication users $n_{pp,min} = \lfloor n_{r,min}/2 \rfloor$;
11 Calculate the potential reporting error $err \leftarrow err(\hat{\boldsymbol{\epsilon}}, \boldsymbol{g}, n, n_{pp,min}, d)$ by Equation (7);
12 **if** $dis > err$ *and* $n_{pp,min} \geq n_{min}$ **then**
13 Sample a user set $U_{r,t}$ from U_A with the size of $|U_{r,t}| = n_{pp,min}$ and remove $U_{r,t}$ from U_A, i.e., $U_A \leftarrow U_A \backslash U_{r,t}$;
14 $\boldsymbol{h}_{r,t} \leftarrow$ Users in $U_{r,t}$ report via an PFO.P with privacy budget list $\boldsymbol{\epsilon}$;
15 Calculate $\hat{\boldsymbol{f}}_{r,t} \leftarrow$ PFO.E($\boldsymbol{h}_{r,t}, \boldsymbol{\epsilon}$);
16 **else**
17 Set $\hat{\boldsymbol{f}}_{r,t} \leftarrow \hat{\boldsymbol{f}}_{r,t\text{-}1}$;
18 **if** $t \geq w_{max}$ **then**
19 $U_A \leftarrow U_A \cup U_{s,t-w_{max}+1} \cup U_{r,t-w_{max}+1}$;
20 **return** $\hat{\boldsymbol{f}}_{r,t}$.

[39] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, 1991.

IX. Appendix

A. Process of PLPD and PLPA

We introduce the process of PLPD and PLPA for subsection IV-C as follows.

PLDP Population Distribution. PLPD takes an available user set U_A, the privacy requirement ($\boldsymbol{w}, \boldsymbol{\epsilon}$) of all users, the value domain d and historical publications as input. PLPD starts with a total population ($U_A = U$) and executes three steps: (1) calculates a fixed population per time slot; (2) distributes publication population in an exponential decreasing fashion; (3) recycles the population spent in time slots falling outside the activate window. Algorithm 5 shows the details of PLPD.

PLPD first gets different privacy budgets' statistic which is used for PFO. It finds the max window size w_{max} and calculate the minimal population average share z_{min} which ensures the total population spent within any window is not large than the total population U. PLPD samples z_{min} size of population $U_{s,t}$ (Line 4) and executes PFO using $U_{s,t}$ (Line 5 and Line 7) to get the estimation frequency of current data. It then uses these estimation to calculate the private dissimilarity (Line 8). Rather than directly calculating new reporting by spending half of the remaining population, PLPD first calculates the new reporting error without any population spent (Lines 9–11). It judges whether the dissimilarity is larger than the error. If so, it spends the population and reports new estimation (Lines 14–15). Otherwise, it sets the current estimation as the before estimation. Finally, PLPD recycles the population that spent for private dissimilarity calculation and new estimation at time slot $t - w_{max} + 1$. We give an example for the procedure of PLPD in Example 6.

Example 6. *Consider a scenario with* 5 *two-by-two connected locations* $\{A, B, C, D, E\}$ *and* 1000 *users* $u_1, u_2, \ldots, u_{1000}$. *These users have privacy requirements in domain* $\{0.2, 0.4, 0.6, 0.8\}$ *and window size requirements in domain* $\{1, 2, 3, 4\}$. *From this, we can determine that the input and output domain size is* $d = 5$. *The distinct privacy budget vector is* $\tilde{\boldsymbol{\epsilon}} = \langle 0.2, 0.4, 0.6, 0.8\rangle$, *with probability vectors* $\tilde{\boldsymbol{q}} = \langle 0.19, 0.18, 0.17, 0.16\rangle$ *and* $\tilde{\boldsymbol{p}} = \langle 0.23, 0.27, 0.31, 0.36\rangle$. *The corresponding statistic vector of* $\tilde{\boldsymbol{\epsilon}}$ *is* $\boldsymbol{g} = \langle 0.35, 0.25, 0.25, 0.15\rangle$. *The maximum window size is* $w_{max} = 4$, *and the minimum sampling size is* $z_{min} = \lfloor 1000/(2 \times 4) \rfloor = 125$.

At time slot 1, *the available user set* U_A *is set to* U. *It samples* 125 *users. Assume these* 125 *users are* $U_{s,1} = \{u_1, u_2, \ldots, u_{125}\}$. *Then* U_A *is updated as* $U_A = U_A \backslash U_{s,1} = \{u_{126}, u_{127}, \ldots, u_{1000}\}$. *Users in* $U_{s,1}$ *report their obfuscated locations using PFO.P. Assume the obfuscated frequency is* $\boldsymbol{h}_{s,1} = \langle 0.22, 0.2, 0.18, 0.22, 0.18\rangle$. *We can then obtain the estimation by Equation* (1) *as* $\hat{\boldsymbol{f}}_{s,1} = \langle 0.396, 0.2, 0.004, 0.396, 0.004\rangle$ *with variance sum as* $\sum_{j=1}^{d}\text{Var}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right] = 0.606$. *Next, we calculate the dissimilarity* $dis = 0.185$. *The potential publication user number* $n_{pp,min} = \left\lfloor \frac{n}{2}/2 \right\rfloor = 250$. *Thus, we can calculate* $err = 0.061$. *Since* $dis > err$, *the system continues to sample* 250 *users from* U_A *for a new publication. Assume the sampling set is* $U_{r,1} = \{u_{126}, u_{127}, \ldots, u_{375}\}$. *These users report their obfuscated locations using PFO.P. Assume the obfuscated frequency is* $\boldsymbol{h}_{r,1} = \langle 0.23, 0.19, 0.19, 0.21, 0.18\rangle$. *We can then get the estimation as* $\hat{\boldsymbol{f}}_{r,1} = \langle 0.494, 0.102, 0.102, 3.00, 0.04\rangle$. *Because there are no users for historical publications out of window size* w_{max} *(i.e.,* $t < w_{max}$*),* U_A *remain unchanged.*

At time slot 2, *it still samples* 125 *users from* U_A *for dis calculation. Assume these* 125 *users are* $U_{s,2} = \{u_{376}, u_{377}, \ldots, u_{500}\}$. *Then the available user set is updated as* $U_A = U_A \backslash U_{s,2} = \{u_{501}, u_{502}, \ldots, u_{1000}\}$. *The system still calculates the minimum remaining population size* $n_{r,min} = 1000/2 - 250 = 250$. *Then, it calculates the potential publication user number* $n_{pp,min} = \lfloor n_{r,min}/2 \rfloor = 125$ *and err. Assume* $dis > err$, *it continues to sample* $n_{pp,min} = 125$ *users for a new publication. Assume the sampling user set is* $U_{r,2} = \{u_{501}, u_{502}, \ldots, u_{625}\}$. *Then, the available user set is*

updated as $U_A = U_A \backslash U_{r,2} = \{u_{626}, u_{627}, \ldots, u_{1000}\}$. *It then reports a new publication* $\hat{\boldsymbol{f}}_{r,2}$.

Assume there are no new publications from time slot 3 *(i.e., dis* $\leq$ *err and* $\hat{\boldsymbol{f}}_{r,3} = \hat{\boldsymbol{f}}_{r,2}$*). The system still consumes* $z_{min} = 125$ *population for dissimilarity calculation. Assume the consumed population set is* $U_{r,3} = \{u_{626}, u_{627}, \ldots, u_{750}\}$. *At time slot* 4*, the available user set is* $U_A = \{u_{751}, u_{752}, \ldots, u_{1000}\}$. *The system samples* $z_{min} = 125$ *users for dissimilarity calculation. Assume the sampling user set is* $U_{s,4} = \{u_{751}, u_{752}, \ldots, u_{875}\}$. *Then* U_A *is updated as* $U_A = \{u_{876}, u_{877}, \ldots, u_{1000}\}$. *Assume dis* $>$ *err. The system calculates the minimum remaining population size* $n_{r,min} = 1000/2 - (250 + 125) = 125$ *and samples* $n_{pp,min} = \lfloor n_{pp,min}/2 \rfloor = 62$ *users for new publication. Assume these* 62 *users are* $U_{r,4} = \{u_{876}, u_{877}, \ldots, u_{937}\}$. *It then reports a new publication* $\hat{\boldsymbol{f}}_{r,4}$. *Additionally, it recycles the population consumption from time slot* 1*, specifically,* $U_A = U_A \cup U_{s,1} \cup U_{r,1} = \{u_1, u_2, \ldots, u_{375}, u_{938}, u_{939}, \ldots, u_{1000}\}$.

The process repeats in a similar fashion for subsequent time slots.

PLDP Population Absorption. PLPD performs well for steams with frequent but gradual changes. However, it becomes less effective when dealing with steams that changes that change infrequently but dramatically. To address this scenario, we propose PLDP Population Absorption (PLPA).

Algorithm 6 shows the process of PLPA. In PLPA, like PLPD, it calculates the statistics $\boldsymbol{g}$ of distinct privacy budget $\tilde{\boldsymbol{\epsilon}}$ and the maximum window size w_{max} among all users. The process of dissimilarity calculation remains the same as in PLPD. However, the population allocation method is different. In PLPA, it first pre-allocates to $\lfloor n/(2w_{max}) \rfloor$ population at each time slot. This population value, called unit population, is the smallest allocation and cannot be further divided. Each unit population can be nullified by predecessors or absorbed by successors within the w_{max} window. However, the average population consumption across each w_{max} period is not allowed to exceed the unit population.

Especially, for error calculation, the system first determines the current nullified t_N, which represents the number of consecutive time slots—from $l+1$ to $l+t_N$—whose populations are nullified by the last new publication time slot l. These t_N population resources are marked as null. PLPA checks whether the current time slot t is before $l + t_N$ (Line 5). If so, the population at time slot t is remains nullified and needs to be skipped (Line 6). Otherwise, it calculates the number of time slot t_A that can be absorbed from $l+t_N+1$ to t (Line 8) and absorbs them to increase the current population (Line 9). The error calculation (Line 10) and dissimilarity-error comparison process (Lines 11–16) follow the same approach as in PLPD. Finally, PLPA recycles the historical population at time slot $t - w_{max} + 1$.

We give an example for PLPA as follows.

Example 7. *Assume the users are the same in Example 6. At time slot* 1*, PLPA spends* $U_{s,1} = \{u_1, u_2, \ldots, u_{125}\}$ *in calculating the dissimilarity* $dis = 0.185$. *It then gets the user set* $|U_{r,l}| = 0$ *consumption for publication at the last time* $l = 0$ *and calculates the nullified* $t_N = -1$. *Because* $t - l = 1 > t_N$*, PLPA calculates* $t_A = t - l - t_N = 2$*, which mean the new publication can consume two shares of populations. It calculate the two share population quantity* $n_{pp,min} = \lfloor 1000/(2 \times 4) \rfloor \times 2 = 250$ *and the potential reporting error* $err = 0.03$. *Because dis* $>$ *err, PLPA samples a user set* $|U_{r,1}|$ *of size* 250 *from* U_A. *Assume* $|U_{r,1}| = \{u_{126}, u_{127}, \ldots, u_{375}\}$. *PLPA reports a new publication consuming* $|U_{r,1}|$.

At time slot 1*, assume the sampling* 125 *users for dissimilarity calculation is* $U_{s,2} = \{u_{376}, u_{377}, \ldots, u_{500}\}$. *Thus the available user set is updated as* $U_A = U_A \backslash U_{s,2} = \{u_{500}, u_{501}, \ldots, u_{1000}\}$. *The nullified* $t_N = \frac{250}{1000/(2\times4)} = 2$*, which is larger than* $t - l = 1$. *Thus, PLPA approximates the current publication* $\hat{\boldsymbol{f}}_2$ *as the last publication* $\hat{\boldsymbol{f}}_1$.

The process repeats in a similar way for subsequent time slots.

Algorithm 6: PLDP Population Absorption

Input: Available user set U_A, privacy requirement ($\boldsymbol{w}$, $\boldsymbol{\epsilon}$) of all Users U, data domain size d, historical data publication ($\hat{\boldsymbol{f}}_1, \hat{\boldsymbol{f}}_2, \ldots, \hat{\boldsymbol{f}}_{t\text{-}1}$)
Output: $\hat{\boldsymbol{f}}_t$

1 Calculate the distinct privacy budget $\tilde{\boldsymbol{\epsilon}}$ with the statistic $\boldsymbol{g}$;
2 Get the maximal window size $w_{max} = \max(\boldsymbol{w})$;
// sub-mechanism $M_{s,t}$
3 The same as Line 3-8 in Algorithm 5;
// sub-mechanism $M_{r,t}$
4 Calculate the nullified $t_N = \frac{|U_{r,l}|}{\lfloor n/(2w_{max}) \rfloor} - 1$;
5 **if** $t - l < t_N$ **then**
6 $\hat{\boldsymbol{f}}_t \leftarrow \hat{\boldsymbol{f}}_{t\text{-}1}$;
7 **else**
8 Calculate time slots that can be absorbed $t_A = t - l - t_N$;
9 Set the number of potential publication users $n_{pp,min} \leftarrow \lfloor n/(2w_{max}) \rfloor \cdot \min(t_A, w_{max})$;
10 Calculate the potential reporting error *err* by Equation (7);
11 **if** *dis* $>$ *err* **then**
12 Sample a user set $U_{r,t}$ from U_A with the size of $|U_{r,t}| = n_{pp,min}$ and remove $U_{r,t}$ from U_A, i.e., $U_A \leftarrow U_A \backslash U_{r,t}$;
13 $\boldsymbol{h}_{r,t} \leftarrow$ Users in $U_{r,t}$ report via an PFO.P with privacy budget list $\boldsymbol{\epsilon}$;
14 Calculate $\hat{\boldsymbol{f}}_{r,t} \leftarrow$ PFO.E($\boldsymbol{h}_{r,t}, \boldsymbol{\epsilon}$);
15 **else**
16 Set $\hat{\boldsymbol{f}}_{r,t} \leftarrow \hat{\boldsymbol{f}}_{r,t\text{-}1}$;
17 **if** $t \geq w_{max}$ **then**
18 $U_A \leftarrow U_A \cup U_{s,t-w_{max}+1} \cup U_{r,t-w_{max}+1}$;
19 **return** $\hat{\boldsymbol{f}}_{r,t}$.

B. More Details for Modified Personalized Frequency Oracle Theorems

Theorem IX.1. *GPRR is* Φ*-PLDP and* $\hat{f}_j = \sum_{k=1}^{m} \alpha_k \cdot \frac{h_{k,j} - \tilde{q}_k}{\tilde{p}_k - \tilde{q}_k}$ *is an unbiased aggregation of* f_j *for all* $\omega_j \in \Omega$.

Proof. For any two input $x_i, x_i' \in \Omega$ of u_i with the output $y_i \in \Omega$, we have

$$\frac{\Pr[M(x_i) = y_i]}{\Pr[M(x_i') = y_i]} \leq \frac{p_i}{q_i} = e^{\epsilon_i}. \tag{9}$$

Thus, GPRR is Φ-PLDP.

For any expectation $\hat{f}_j \in \hat{\boldsymbol{f}}$ of $f_j \in \boldsymbol{f}$, we have

$$\begin{aligned}
\mathbb{E}\left[\hat{f}_j\right] &= \mathbb{E}\left[\sum_{k=1}^{m} \alpha_k \cdot \frac{h_{k,j} - \tilde{q}_k}{\tilde{p}_k - \tilde{q}_k}\right] \\
&= \mathbb{E}\left[\sum_{k=1}^{m} \alpha_k \cdot \frac{f_j \cdot \tilde{p}_k + (1 - f_j)\tilde{q}_k - \tilde{q}_k}{\tilde{p}_k - \tilde{q}_k}\right] \\
&= \mathbb{E}\left[f_j \cdot \sum_{k=1}^{m} \alpha_k\right] \\
&= f_j.
\end{aligned}$$

Thus, $\hat{f}_j$ is an unbiased estimation of f_j. □

Theorem IX.2. *For any $\boldsymbol{\Phi}$-PLDP mechanism with $\boldsymbol{\Phi} : U \to \mathcal{E}$ independent of Ω, and any $\tilde{p}_k \in \tilde{\boldsymbol{p}}, \tilde{q}_k \in \tilde{\boldsymbol{q}}$ with $\frac{1}{d} < \tilde{p}_k \leq 1$ and $\tilde{q}_k = \frac{1-\tilde{p}_k}{d-1}$, the variance of the j-th frequency estimation $\hat{f}_j \in \hat{\boldsymbol{f}}$ in Equation* (1) *is*

$$\text{Var}\left[\hat{f}_j\right] = \begin{cases} V_{A,j} = \frac{1}{\sum_{k=1}^{m} 1/(\lambda_k + \mu_k \cdot f_j)}, & \text{if } \alpha_k = \frac{1/\text{Var}\left[\hat{f}_{k,j}\right]}{\sum_{k'=1}^{m} 1/\text{Var}\left[\hat{f}_{k',j}\right]}, \\ V_{B,j} = \frac{\sum_{k=1}^{m} \frac{(\lambda_k + \mu_k/d)}{(\lambda_k + \mu_k \cdot f_j)^2}}{\left(\sum_{k=1}^{m} 1/(\lambda_k + \mu_k/d)\right)^2}, & \text{if } \alpha_k = \frac{1/\text{Vara}\left[\hat{f}_{k,j}\right]}{\sum_{k'=1}^{m} 1/\text{Vara}\left[\hat{f}_{k',j}\right]}; \end{cases} \tag{10}$$

Besides, the upper bound of $\sum_{j=1}^{d} V_{A,j}$ and the lower bound of $\sum_{j=1}^{d} V_{B,j}$ are both

$$\text{Var}\left[\hat{\boldsymbol{f}}\right] = 1/\sum_{k=1}^{m} n/\left(d\lambda_k + \mu_k\right), \tag{11}$$

where $\lambda_k = \frac{\tilde{q}_k(1-\tilde{q}_k)}{g_k(\tilde{p}_k - \tilde{q}_k)^2}$ and $\mu_k = \frac{1-\tilde{p}_k-\tilde{q}_k}{g_k(\tilde{p}_k-\tilde{q}_k)}$.

Proof. For any random variable $\hat{h}_j \in \boldsymbol{h}$, it is the average of n independent random variables drawn from the Bernoulli distribution. More specifically, $f_j g_k n$ of these random variables are drawn from the Bernoulli distribution with parameter $\tilde{p}_k$. Besides, $(1 - g_k) f_j n$ of these variables are drawn from the Bernoulli distribution with parameter $\tilde{q}_k$. Thus, if setting $\alpha_k = \frac{1/\text{Var}\left[\hat{f}_{k,j}\right]}{\sum_{k'=1}^{m} 1/\text{Var}\left[\hat{f}_{k',j}\right]}$, then

$$\begin{aligned}
\text{Var}\left[\hat{f}_j\right] &= \sum_{k=1}^{m} \alpha_k^2 \cdot \text{Var}\left[\hat{f}_{k,j}\right] \\
&= \sum_{k=1}^{m} \left(\frac{1/\text{Var}\left[\hat{f}_{k,j}\right]}{\sum_{i=1}^{m} 1/\text{Var}\left[\hat{f}_{i,j}\right]}\right)^2 \cdot \text{Var}\left[\hat{f}_{k,j}\right] \\
&= \frac{\sum_{k=1}^{m} 1/\text{Var}\left[\hat{f}_{k,j}\right]}{\left(\sum_{i=1}^{m} 1/\text{Var}\left[\hat{f}_{i,j}\right]\right)^2} \\
&= \frac{1}{\sum_{k=1}^{m} 1/\text{Var}\left[\hat{f}_{k,j}\right]} \\
&= 1/\sum_{k=1}^{m} 1/\left(\lambda_k + \mu_k \cdot f_j\right);
\end{aligned} \tag{12}$$

If setting $\alpha_k = \frac{1/\overline{\text{Var}}\left[\hat{f}_{k,j}\right]}{\sum_{k'=1}^{m} 1/\overline{\text{Var}}\left[\hat{f}_{k',j}\right]}$, then

$$\begin{aligned}
\text{Var}\left[\hat{f}_j\right] &= \sum_{k=1}^{m} \alpha_k^2 \cdot \text{Var}\left[\hat{f}_{k,j}\right] \\
&= \sum_{k=1}^{m} \left(\frac{1/\overline{\text{Var}}\left[\hat{f}_{k,j}\right]}{\sum_{i=1}^{m} 1/\overline{\text{Var}}\left[\hat{f}_{i,j}\right]}\right)^2 \cdot \text{Var}\left[\hat{f}_{k,j}\right] \\
&= \frac{\sum_{k=1}^{m} \overline{\text{Var}}\left[\hat{f}_{k,j}\right] / \left(\text{Var}\left[\hat{f}_{k,j}\right]\right)^2}{\left(\sum_{i=1}^{m} 1/\overline{\text{Var}}\left[\hat{f}_{i,j}\right]\right)^2} \\
&= \frac{\sum_{k=1}^{m} \left(\lambda_k + \mu_k/d\right) / \left(\lambda_k + \mu_k \cdot f_j\right)^2}{\left(\sum_{k=1}^{m} 1/\left(\lambda_k + \mu_k/d\right)\right)^2}
\end{aligned} \tag{13}$$

For Equation (12), let $\phi(x) = \text{Var}[x] = \frac{1}{\sum_k \frac{1}{\lambda_k + \mu_k x}}$, then we have:

$$\sum_{j=1}^{d} V_{A,j} = \sum_{j=1}^{d} \phi\left(\hat{f}_j\right) = \frac{1}{m} \boldsymbol{H}\left(\lambda_1 + \mu_1 f_j, \lambda_2 + \mu_2 f_j, \ldots, \lambda_m + \mu_m f_j\right),$$

where $\boldsymbol{H}$ is the harmonic average. □

From Equation (10) and (11), we can see the upper bound of $\sum_{j=1}^{d} V_{A,j}$ is independent of f_j.

Because $\lambda_k + \mu_k f_j > 0$, $\sum_{j=1}^{d} V_{A,j}$ is a concave function. As $\sum_j f_j = 1$, according to Jensen's inequality, we have:

$$\sum_{j=1}^{d} V_{A,j} = \sum_{j=1}^{d} \phi(\hat{f}_j) \leq d \cdot \phi\left(\frac{1}{d}\right) = d \cdot \frac{1}{\sum_{k=1}^{m} \frac{1}{\lambda_k + \mu_k/d}} = \frac{1}{\sum_{k=1}^{m} \frac{1}{d\lambda_k + \mu_k}}.$$

The equal sign holds when $f_1 = f_2 = \cdots = f_d = \frac{1}{d}$.

For Equation (13), let $\psi(x) = \sum_{k=1}^{m} \frac{\lambda_k + \mu_k/d}{\left(\lambda_k + \mu_k f_j\right)^2}$, then we have:

$$\sum_{j=1}^{d} V_{B,j} = \sum_{j=1}^{d} \psi(\hat{f}_j) = \frac{1}{\left(\sum_{k=1}^{m} \frac{1}{\lambda_k + \mu_k/d}\right)^2} \sum_{j=1}^{d} \sum_{k=1}^{m} \frac{\lambda_k + \mu_k/d}{\left(\lambda_k + \mu_k f_j\right)^2}.$$

Because $\psi(x)$ is a decreasing convex function, according to Jensen's inequality, we have

$$\sum_{j=1}^{d} V_{B,j} = \sum_{j=1}^{d} \psi(f_j) \geq d \cdot \psi\left(\frac{1}{d}\right) = d \cdot \frac{1}{\sum_{k=1}^{m} \frac{1}{\lambda_k + \mu_k/d}} = \frac{1}{\sum_{k=1}^{m} \frac{1}{d\lambda_k + \mu_k}}.$$

The equal sign holds when $f_1 = f_2 = \cdots = f_d = \frac{1}{d}$.

C. More Details for Dissimilarity

Private Dissimilarity. The ideal method of strategy determination is using non-private dissimilarity *dis** for comparison. The expression of non-private dissimilarity is defined as the mean square error between the

$$dis^* = \frac{1}{d} \sum_{j=1}^{d} \left(\boldsymbol{f}_t[j] - \hat{\boldsymbol{f}}_l[j]\right)^2, \tag{14}$$

where $\hat{\boldsymbol{f}}_l$ represents the previous frequency publication that consumes population resources. However, *dis** as an intermediate result leads to privacy leakage. To overcome this limitation, we use a private dissimilarity *dis* instead. The design of *dis* has two requirements: (1) *dis* needs to be private, meaning that given a privacy requirement Φ for *dis*

calculation, it needs to achieve Φ-PLDP; (2) *dis* needs to be an unbiased estimation of the non-private dissimilarity dis^*, ensuring accurate results.

Theorem IX.3. *Let $\hat{\boldsymbol{f}}_{s,t}$ denote the unbiased frequency estimation of $\boldsymbol{f}_t$ from an **Φ**-PLDP frequency estimation mechanism $M_{s,t}$ at current time slot t. Then the following dissimilarity measure*

$$dis = \frac{1}{d}\sum_{j=1}^{d}\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2 - \frac{1}{d}\sum_{j=1}^{d}\mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right] \tag{15}$$

*is **Φ**-PLDP and an unbiased estimation of dis^* in Equation* (14).

Proof. We first prove (1) dis in Equation (15) is Φ-PLDP (*Privacy Proof*) and then prove (2) dis in Equation (15) is an unbiased estimation of dis^* in Equation (14) (*Unbiasedness Proof*).

Privacy Proof. The calculation of dis is only related to $\{y_1, \ldots, y_n\}$ of all users, which is published utilizing GRR. Thus, According to Theorem IX.1, we get dis is Φ-PLDP.

Unbiasedness Proof. Since $\hat{\boldsymbol{f}}_{s,t}$ is an unbiased estimation of $\boldsymbol{f}_t$, for any $\hat{\boldsymbol{f}}_{s,t}[j] \in \hat{\boldsymbol{f}}_{s,t}$ and $\boldsymbol{f}_t[j] \in \boldsymbol{f}_t$, we have

$$\mathbb{E}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right] = \boldsymbol{f}_t[j].$$

Besides, for any random variable X and constant C, according to $\mathrm{Var}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2$, we have

$$\begin{aligned}\mathrm{Var}[X] &= \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] \\ &= \mathbb{E}\left[((X - C) - (\mathbb{E}[X] - C))^2\right] \\ &= \mathbb{E}\left[(X - C)^2\right] - (\mathbb{E}[X] - C)^2.\end{aligned}$$

Since last reporting value $\hat{\boldsymbol{f}}_l$ has been a constant for the current value $\hat{\boldsymbol{f}}_{s,t}$, thus, we have

$$\mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right] = \mathbb{E}\left[\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2\right] - \left(\boldsymbol{f}_t[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2.$$

Therefore,

$$\mathbb{E}\left[\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2\right] = \left(\boldsymbol{f}_t[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2 + \mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right].$$

Hence, the expectation of dis in Equation (15) satisfies

$$\begin{aligned}&\mathbb{E}[dis] \\ =&\mathbb{E}\left[\frac{1}{d}\sum_{j=1}^{d}\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2 - \frac{1}{d}\sum_{j=1}^{d}\mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,t}\right]\right] \\ =&\frac{1}{d}\sum_{j=1}^{d}\mathbb{E}\left[\left(\hat{\boldsymbol{f}}_{s,t}[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2\right] - \frac{1}{d}\sum_{j=1}^{d}\mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,t}\right] \\ =&\frac{1}{d}\sum_{j=1}^{d}\left(\left(\boldsymbol{f}_t[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2 + \mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,t}[j]\right]\right) - \frac{1}{d}\sum_{j=1}^{d}\mathrm{Var}\left[\hat{\boldsymbol{f}}_{s,t}\right] \\ =&\frac{1}{d}\sum_{j=1}^{d}\left(\boldsymbol{f}_t[j] - \hat{\boldsymbol{f}}_{l}[j]\right)^2 \\ =&dis^*.\end{aligned}$$

□

D. Proof of the Theorem for Re-Perturbation (Theorem V.1)

Proof. Let FP_H and FP_L be the sub-mechanisms in GPRR with privacy budget ϵ_H and ϵ_L, respectively. Then, we have

$$\forall y \in Y, \Pr\left[\mathrm{FP}_H(x) = y\right] = \begin{cases} p_H, & \text{if } y = x, \\ q_H = \frac{1-p_H}{d-1}, & \text{otherwize,}\end{cases}$$

and

$$\forall y \in Y, \Pr\left[\mathrm{FP}_L(x) = y\right] = \begin{cases} p_L, & \text{if } y = x, \\ q_L = \frac{1-p_L}{d-1}, & \text{otherwize.}\end{cases}$$

For any re-perturbed value Y', we have the transformation probability relationship shown in Figure 6.

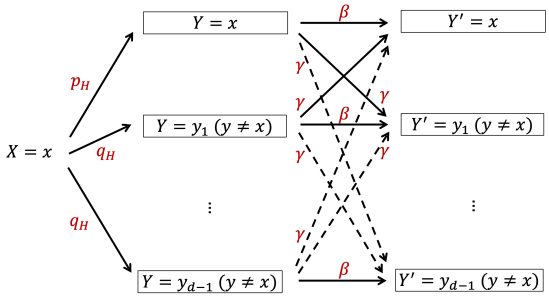


Fig. 6: Probability for perturbation and re-perturbation.

Thus, $\forall y_1', y_2' \in Y'$, we have:

$$\begin{aligned}&\frac{\Pr\left[M(x) = y_1'\right]}{\Pr\left[M(x) = y_2'\right]} \\ =&\frac{\sum_{y\in Y}\Pr\left[\mathrm{FP}(x) = y\right]\cdot\Pr\left[\mathrm{RP}(y) = y_1'\right]}{\sum_{y\in Y}\Pr\left[\mathrm{FP}(x) = y\right]\cdot\Pr\left[\mathrm{RP}(y) = y_2'\right]} \\ \le&\frac{p_H\cdot\beta + (d-1)\cdot q_H\cdot\gamma}{p_H\cdot\gamma + \cdot q_H\cdot\beta + (d-2)\cdot q_H\cdot\gamma} \\ =&\frac{p_H\cdot\frac{dp_L - p_L + p_H - 1}{dp_H - 1} + (d-1)\cdot\frac{1-p_H}{d-1}\cdot\frac{p_H - p_L}{dp_H - 1}}{p_H\cdot\frac{p_H - p_L}{dp_H - 1} + \cdot\frac{1-p_H}{d-1}\cdot\frac{dp_L - p_L + p_H - 1}{dp_H - 1} + (d-2)\cdot\frac{1-p_H}{d-1}\cdot\frac{p_H - p_L}{dp_H - 1}} \\ =&\frac{(d-1)(dp_Hp_L - p_L)}{(p_H + d - 2)(p_H - p_L) + (1-p_H)(dp_L - p_L + p_H - 1)} \\ =&\frac{(d-1)p_L(dp_H - 1)}{(dp_H - 1)(1 - p_L)} \\ =&\frac{p_L}{\frac{1-p_L}{d-1}} \\ =&\frac{p_L}{q_L} \\ =&e^{\epsilon_L}.\end{aligned}$$

Therefore, after the re-perturbing process, it still satisfies ϵ_L-LDP.

Noted that FP_L satisfies ϵ_L-LDP. According to Reference [25], we can calculate FP_L's variance with z_L users is

$$\begin{aligned}Var_L(z_L) &= \frac{1}{z_L^2}\cdot\sum_{j=1}^{d}\left(\frac{z_Lq_L(1-q_L)}{(p_L - q_L)^2} + \frac{z_L(1 - p_L - q_L)}{p_L - q_L}\cdot f_j\right) \\ &= \frac{dq_L(1-q_L)}{z_L(p_L - q_L)^2} + \frac{1 - p_L - q_L}{z_L(p_L - q_L)}\end{aligned}$$

After re-perturbing, the variance becomes

$$\frac{dq_L(1-q_L)}{(z_L + z_H)(p_L - q_L)^2} + \frac{1 - p_L - q_L}{(z_L + z_H)(p_L - q_L)}$$

Therefore the improvement is

$$\begin{aligned}\Delta(Var) &= Var_L(z_L) - Var_L(z_L + z_H) \\ &= \frac{z_H}{z_L(z_L + z_H)}\left(\frac{dq_L(1-q_L)}{(p_L - q_L)^2} + \frac{1 - p_L - q_L}{p_L - q_L}\right).\end{aligned}$$

□

E. Proofs of Theorems for Time Complexity

1) Proof of Theorems IV.1:

Proof. The time complexities of PFO.P, PFO.A and PFO.E are $O(z_{min})$, $O(z_{min} + d \cdot m)$ and $O(d \cdot m)$, respectively. Additionally, the time complexity of *dis* calculation is $O(d)$. Therefore, the overall time complexity of sub-mechanism $\mathcal{M}_{s,t}$ is $O(z_{min} + d \cdot m)$ for both PLPD and PLPA .

For sub-mechanism $\mathcal{M}_{r,t}$ in PLPD, the sampling size is at most $n/4$. Thus, the time complexity for counting distinct elements in $\tilde{\boldsymbol{\epsilon}}$ is $O(n)$. The time complexity of the FO is $O(n + d \cdot m)$. Thus, the time complexity of $\mathcal{M}_{r,t}$ in PLPD is $O(n+d \cdot m)$. For sub-mechanism $\mathcal{M}_{r,t}$ in PLPA, the sampling size is at most $\max(\lfloor n/(2w_{max})\rfloor) \cdot w_{max} = n/2$. Hence, the time complexity of $\mathcal{M}_{r,t}$ in PLPA is also $O(n + d \cdot m)$.

Consequently, the time complexity for both PLPD and PLPA is $O(n + d \cdot m)$. □

2) Proof of Theorems V.2:

Proof. The time complexity of OPS is $O(m \cdot n)$ in both PLPD$^+$ and PLPA$^+$. Since OPS is executed only once at the beginning of the process, its impact is minimal in subsequent steps.

The time complexity of PUE is $O(m^2)$. After applying PUE, the sampling size increases from z to at most $z \cdot m$. The time complexities of PFO.P, PFO.A and PFO.E are $O(z_{opt})$, $O(z_{opt} \cdot m + d \cdot m)$ and $O(d \cdot m)$, respectively. Additionally, the time complexity of *dis* calculation is $O(d)$. Therefore, the overall time complexity of sub-mechanism $\mathcal{M}_{s,t}$ is $O\left(m^2 + z_{opt} \cdot m + d \cdot m\right)$ for both PLPD$^+$ and PLPA$^+$.

For sub-mechanism $\mathcal{M}_{r,t}$ in PLPD$^+$, the sampling size is at most $n/4$. Thus, the time complexity for counting distinct elements in $\tilde{\boldsymbol{\epsilon}}''_{opt}$ is $O(n)$. The time complexity of the enhanced PFO is $O(n \cdot m + d \cdot m)$. Thus, the time complexity of $\mathcal{M}_{r,t}$ in PLPD$^+$ is $O(n \cdot m + d \cdot m)$. For sub-mechanism $\mathcal{M}_{r,t}$ in PLPA$^+$, the sampling size is at most $\max(\lfloor n/(2w_{opt})\rfloor) \cdot w_{opt} = n/2$. Hence, the time complexity of $\mathcal{M}_{r,t}$ in PLPA$^+$ is also $O(n \cdot m + d \cdot m)$.

Consequently, the time complexity for both PLPD$^+$ and PLPA$^+$ is $O(n \cdot m + d \cdot m)$. □

F. Proofs of Theorems for Privacy Analysis

According to relationship between the definitions of $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP, $\boldsymbol{\epsilon}$-PLDP, we have the following important three lemmas for privacy analysis.

Lemma IX.1. *Given any window size $w \in \mathbb{N}^+$, a mechanism $\mathcal{M}$ satisfies $(w \cdot \mathbf{1}, \boldsymbol{\epsilon})$-EPLDP if it satisfies $\boldsymbol{\epsilon}$-PLDP in any window of size w.*

Proof. Let S_t and $\hat{S}_t$ be any w-neighboring stream prefixes. If $S_t = \hat{S}_t$, then for any user $u_i \in \mathcal{U}$, it is obvious that $\Pr[\mathcal{M}(S_t)] \leq e^{\boldsymbol{\epsilon}(u_i)} \cdot \Pr\left[\mathcal{M}\left(\hat{S}_t\right)\right]$. Otherwise, we can find $\tau_1, \tau_2 \in [t]$ satisfying $\tau_2 - \tau_1 + 1 \leq w$, with $S_t[\tau_1] \neq \hat{S}_t[\tau_1]$ and $S_t[\tau_2] \neq \hat{S}_t[\tau_2]$, meanwhile for any $\tau \in [1, \tau_1 - 1] \cup [\tau_2 + 1, t]$ it satisfies $S_t[\tau] = \hat{S}_t[\tau]$, and for any $\tau \in [\tau_1, \tau_2]$, it satisfies $S_t[\tau]$ and $\hat{S}_t[\tau]$ are neighboring.

Let $\tilde{\boldsymbol{\epsilon}} = \text{Unique}(\boldsymbol{\epsilon})$ be the unique ϵ among all users. Let m be the number of $\tilde{\boldsymbol{\epsilon}}$. For any two w length sub-stream segments $S_{\tau-w+1,\tau}$ and $\hat{S}_{\tau-w+1,\tau}$ satisfying $\tau - w + 1 \leq \tau_1 \leq \tau_2 \leq \tau$, since $\mathcal{M}$ satisfies $\boldsymbol{\epsilon}$-PLDP within any w window, then for any personalized subset-stream segment $S^{(k)}_{\tau-w+1,\tau} \subseteq S_{\tau-w+1}$ and $\hat{S}^{(k)}_{\tau-w+1,\tau} \subseteq S_{\tau-w+1}$ we have $\forall y \in \text{Range}(\mathcal{M})$,

$$\Pr\left[\mathcal{M}\left(S^{(k)}_{\tau-w+1,t}\right) = y\right] \leq e^{\boldsymbol{\epsilon}(u_i)} \Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_{\tau-w+1,\tau}\right) = y\right].$$

Therefore, for any personalized stream prefix $S^{(k)}_t \subseteq S_t, \hat{S}^{(k)}_t \subseteq \hat{S}_t$ and $\forall y \in \text{Range}(\mathcal{M})$ with $y = y_{1:\tau-w} \| y_{\tau-w+1,\tau} \| y_{\tau,t}$, we have

$$\begin{aligned}
&\frac{\Pr\left[\mathcal{M}\left(S^{(k)}_t\right) = y\right]}{\Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_t\right) = y\right]} \\
\leq &\frac{\Pr\left[\mathcal{M}\left(S^{(k)}_{1,\tau-w}\right) = y_{1,\tau-w}\right]}{\Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_{1,\tau-w}\right) = y_{1,\tau-w}\right]} \times \frac{\Pr\left[\mathcal{M}\left(S^{(k)}_{\tau-w+1,\tau}\right) = y_{\tau-w+1,\tau}\right]}{\Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_{\tau-w+1,\tau}\right) = y_{\tau-w+1,\tau}\right]} \\
&\times \frac{\Pr\left[\mathcal{M}\left(S^{(k)}_{\tau+1,t}\right) = y_{\tau+1,t}\right]}{\Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_{\tau+1,t}\right) = y_{\tau+1,t}\right]} \\
= &\frac{\Pr\left[\mathcal{M}\left(S^{(k)}_{\tau-w+1,\tau}\right) = y_{\tau-w+1,\tau}\right]}{\Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_{\tau-w+1,\tau}\right) = y_{\tau-w+1,\tau}\right]} \\
\leq &e^{\boldsymbol{\epsilon}(u_i)}.
\end{aligned}$$

Therefore, $\mathcal{M}$ satisfies $(w \cdot \mathbf{1}, \boldsymbol{\epsilon})$-EPLDP. □

Lemma IX.2. *A mechanism $\mathcal{M}$ satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP if it satisfies $(\max(\boldsymbol{w}) \cdot \mathbf{1}, \boldsymbol{\epsilon})$-EPLDP.*

Proof. For any user u_i with $\boldsymbol{\epsilon}(u_i) = \epsilon_k$ and $\boldsymbol{w}(u_i) = w_k$, let $S^{(k)}_t, \hat{S}^{(k)}_t \subseteq S_t$ be any pair of w_k-neighboring stream prefixes. If $S^{(k)}_t = \hat{S}^{(k)}_t$, then it is evident that $\forall y \in \text{Range}(\mathcal{M}), \Pr\left[\mathcal{M}\left(S^{(k)}_t\right) = y\right] \leq e^{\epsilon_k} \cdot \Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_t\right) = y\right]$. Otherwise, there exists $\tau_1, \tau_2 \in [t]$ satisfying $\tau_2 - \tau_1 + 1 \leq w_k$ with $S_t[\tau_1] \neq \hat{S}_t[\tau_1]$ and $S_t[\tau_2] \neq \hat{S}_t[\tau_2]$. Because $w_k \leq \max(\boldsymbol{w})$, then we can choose $\tau'_1 \in [1, \tau_1], \tau'_2 \in [\tau_2, t]$ with $\tau'_2 - \tau'_1 + 1 = \max(\boldsymbol{w})$. Thus $S^{(k)}_t \sim_{\max(\boldsymbol{w})} \hat{S}^{(k)}_t$. Because $\mathcal{M}$ satisfies $(\max(\boldsymbol{w}) \cdot \mathbf{1}, \boldsymbol{\epsilon})$-EPLDP, then $\forall y = \text{Range}(\mathcal{M})$, we have

$$\Pr\left[\mathcal{M}\left(S^{(k)}_t\right) = y\right] \leq e^{\epsilon_k} \cdot \Pr\left[\mathcal{M}\left(\hat{S}^{(k)}_t\right) = y\right],$$

Therefore, $\mathcal{M}$ satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP.

□

Lemma IX.3. *A mechanism $\mathcal{M}$ satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPLDP if for any window size $\hat{w} \in [\min(\boldsymbol{w}), \max(\boldsymbol{w})]$, $\mathcal{M}$ satisfies $(\hat{w} \cdot \mathbf{1}, \hat{\boldsymbol{\epsilon}})$-EPLDP where*

$$\hat{\boldsymbol{\epsilon}}(u_i) = \begin{cases} \boldsymbol{\epsilon}(u_i), & \text{if } \boldsymbol{w}(u_i) \leq \hat{w}, \\ \boldsymbol{\epsilon}(u_i) / \lceil \boldsymbol{w}(u_i)/\hat{w} \rceil, & \text{otherwize.} \end{cases}$$

Proof. All users can be classified into two cases: (1) the users whose window sizes are no more than $\hat{w}$; (2) the users whose window sizes are larger than $\hat{w}$.

For any user u_i in case (1), it satisfies $\boldsymbol{w}(u_i) = w_k \leq \hat{w}$. Then $\forall S^{(k)}_t, \hat{S}^{(k)}_t$ with $S^{(k)}_t \sim_{w_k} \hat{S}^{(k)}_t$, we have $S^{(k)}_t \sim_{\hat{w}}$

$\hat{S}_t^{(k)}$. Because $\mathcal{M}$ satisfies $(\hat{w}\cdot\mathbf{1},\hat{\boldsymbol{\epsilon}})$-EPLDP , we have $\forall y\in$ Range$(\mathcal{M})$,

$$\frac{\Pr\left[\mathcal{M}\left(S_t^{(k)}\right)=y\right]}{\Pr\left[\mathcal{M}\left(\hat{S}_t^{(k)}\right)=y\right]}\leq e^{\hat{\boldsymbol{\epsilon}}(u_i)}=e^{\boldsymbol{\epsilon}(u_i)}.$$

For any user u_i in case (2), it satisfies $\boldsymbol{w}(u_i)=w_k>\hat{w}$. Let $S_t^{(k)},\hat{S}_t^{(k)}$ be any two personalized stream prefixes satisfying $S_t\sim_{w_k}\hat{S}_t$. If $S_t^{(k)}=\hat{S}_t^{(k)}$, then it is obvious that $\Pr\left[\mathcal{M}\left(S_t^{(k)}\right)=y\right]/\Pr\left[\mathcal{M}\left(\hat{S}_t^{(k)}\right)=y\right]=1\leq e^{\boldsymbol{\epsilon}(u_i)}$. Otherwise, we can find τ_1,τ_2 satisfying $\tau_2-\tau_1+1\leq w_k, S_t^{(k)}[\tau_1]\neq\hat{S}_t^{(k)}[\tau_1], S_t^{(k)}[\tau_2]\neq\hat{S}_t^{(k)}[\tau_2]$ and $\forall\tau\in[1,\tau_1-1]\cup[\tau_2+1,t]$, $S_t^{(k)}[\tau]=\hat{S}_t^{(k)}[\tau]$. Let $z=\lceil w_k/\hat{w}\rceil$, then we can construct a list of $z-1$ stream prefixes $\check{\boldsymbol{S}}=\left\langle\check{S}_t^{(k,1)},\check{S}_t^{(k,2)},\ldots,\check{S}_t^{(k,z-1)}\right\rangle$ satisfying $S_t^{(k)}\sim_{\hat{w}}\check{S}_t^{(k,1)}$, $\check{S}_t^{(k,z-1)}\sim_{\hat{w}}\hat{S}_t^{(k)}$ and $\forall j\in[2,z-1]$, $\check{S}_t^{(k,j-1)}\sim_{\hat{w}}\check{S}_t^{(k,j)}$. By adding $S_t^{(k)}$ to the head of $\check{\boldsymbol{S}}$ and $\hat{S}_t^{(k)}$ to tail, we can get the extended list of $z+1$ stream prefixes $\check{\boldsymbol{S}}^{+}=\left\langle S_t^{(k)},\check{S}_t^{(k,1)},\check{S}_t^{(k,2)},\ldots,\check{S}_t^{(k,z-1)},\hat{S}_t^{(k)}\right\rangle$. For any neighboring pair prefixes x,x' in $\check{\boldsymbol{S}}^{+}$, because $\mathcal{M}$ satisfies $(\hat{w}\cdot\mathbf{1},\hat{\boldsymbol{\epsilon}})$-EPLDP, we have $\forall y\in$ Range$(\mathcal{M})$, $\Pr[\mathcal{M}(x)=y]/\Pr[\mathcal{M}(x')=y]\leq e^{\epsilon_k/z}$. Therefore, for $\forall y\in$ Range$(\mathcal{M})$, we have

$$\begin{aligned}
&\frac{\Pr\left[\mathcal{M}\left(S_t^{(k)}\right)=y\right]}{\Pr\left[\mathcal{M}\left(\hat{S}_t^{(k)}\right)=y\right]}\\
=&\frac{\Pr\left[\mathcal{M}\left(S_t^{(k)}\right)=y\right]}{\Pr\left[\mathcal{M}\left(\check{S}_t^{(k,1)}\right)=y\right]}\times\prod_{j=2}^{z-1}\frac{\Pr\left[\mathcal{M}\left(\check{S}_t^{(k,j-1)}\right)=y\right]}{\Pr\left[\mathcal{M}\left(\check{S}_t^{(k,j)}\right)=y\right]}\\
&\times\frac{\Pr\left[\mathcal{M}\left(\check{S}_t^{(k,z-1)}\right)=y\right]}{\Pr\left[\mathcal{M}\left(\hat{S}_t^{(k)}\right)=y\right]}\\
\leq&e^{\epsilon_k/z}\times\left(e^{\epsilon_k/z}\right)^{z-2}\times e^{\epsilon_k/z}\\
=&e^{\epsilon_k}\\
=&e^{\boldsymbol{\epsilon}(u_i)}.
\end{aligned}$$

Therefore, $\mathcal{M}$ satisfies $(\boldsymbol{w},\boldsymbol{\epsilon})$-EPLDP. □

Next, we give one special lemmas for the privacy analysis of PLPD and PLPA, and one corollary for that of PLPD$^+$ and PLPA$^+$.

Lemma IX.4. *For both PLPD and PLPA, in any time window composed of $w_{max}=\max(\boldsymbol{w})$ consecutive time slots, each user reports to the server at most once.*

Proof. We only need to prove that the number of users within any w_{max} window is no more than n, i.e., $\sum_{\tau=t-w_{max}+1}^{t}|U_\tau|\leq n$. Then, because we sample a fresh set of users U_τ at each time slot τ satisfying $U_{s,\tau}\cap U_{r,\tau}=\emptyset$, we can guarantee each user reports only once within any window of size w_{max}.

For PLPD, in $M_{s,t}$, $\lfloor n/(2w_{max})\rfloor$ users are allocated at each time slot t. Thus, for any t and $k\in[t]$, there are $\sum_{\tau=k-w_{max}+1}^{k}|U_{s,\tau}|\leq n/2$ users. In $M_{r,t}$, it either publishes with additional users $U_{r,t}$ or approximates the last release without any user assignment. In the former case, there are at most $|U_{r,t}|=\left(n/2-\sum_{\tau=k-w_{max}+1}^{k-1}|U_{r,k}|\right)/2$ users. Because $M_{r,t}$ always uses at most half of the available users, we have $0\leq\sum_{\tau=k-w_{max}+1}^{k}|U_{r,\tau}|\leq n/2$. In the latter case, $|U_{r,t}|=0$. Therefore, for any t and $k\in[t]$, the total number of publication users within a w_{max} window for PLPD is

$$\sum_{\tau=k-w_{max}+1}^{k}|U_\tau|=\sum_{\tau=k-w_{max}+1}^{k}|U_{s,\tau}|+\sum_{\tau=k-w_{max}+1}^{k}|U_{r,\tau}|\leq n.$$

For PLPA, similarly, in $M_{s,t}$, there are $\sum_{\tau=k-w_{max}+1}^{k}|U_{s,\tau}|\leq n/2$ users for any t and $\tau\in[t]$. In $M_{r,t}$, assume there are at most c new publications in any w_{max} window. We denote these publication time slots as $(\tau_1,\tau_2,\ldots,\tau_c)$. For any publication time slot $\tau_j\,(j\in[c])$, we denote the number of its absorbed population share as η_j, thus after absorption, time slot τ_j will occupy η_j+1 population shares. Figure 7 illustrates an example for $c=3$ and $w_{max}=9$, in which, for instance, time slot 3 absorbs population shares from time slots 1 and 2, while nullifying the population at time slots 4 and 5. Noted that the absorbed time

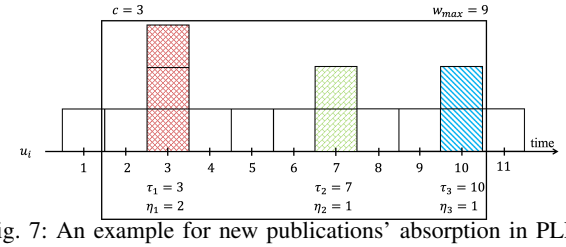


Fig. 7: An example for new publications' absorption in PLPA.

slot length and nullified time slot length are symmetric with respect to the new publication, according to the definition of τ_j, we have:

$$w_{max}\geq\sum_{j=1}^{c}(1+2\eta_j)-\eta_1-\eta_c.$$

Then, for any t and $k\in[t]$, we have

$$\begin{aligned}
\sum_{\tau=k-w_{max}+1}^{k}|U_{r,\tau}|&\leq\left\lfloor\frac{n}{2w_{max}}\right\rfloor\cdot\sum_{j=1}^{c}(1+\eta_j)\\
&\leq\frac{n\cdot\sum_{j=1}^{c}(1+\eta_j)}{2\sum_{j=1}^{c}(1+2\eta_j)-2\eta_1-2\eta_c}\\
&=\frac{n\cdot\sum_{j=1}^{c}(1+\eta_j)}{2\sum_{j=1}^{c}(1+\eta_j)+2\sum_{j=2}^{c-1}\eta_j}\\
&\leq n/2.
\end{aligned}$$

Therefore, for any t and $k\in[t]$, the total number of publication users within a w_{max} window for PLPA is

$$\sum_{\tau=k-w_{max}+1}^{k}|U_\tau|=\sum_{\tau=k-w_{max}+1}^{k}|U_{s,\tau}|+\sum_{\tau=k-w_{max}+1}^{k}|U_{r,\tau}|\leq n.$$

□

From Lemma IX.4, we can get the following obvious corollary.

Corollary IX.1. *For both PLPD$^+$ and PLPA$^+$, in any time window composed of w_{opt} consecutive time slots, each user reports to the server at most once.*

1) Proof of Theorem IV.2:

Proof. According to Lemma IX.4, for any user u_i in any window of size w_{max}, it reports at most once. Thus for any two personalized w_{max}-neighboring stream segments $S^{(k)}_{\tau-w_{max}+1,\tau}, \hat{S}^{(k)}_{\tau-w_{max}+1,\tau}$, we have $\forall y \in \text{Range}(\mathcal{M})$,

$$\frac{\Pr\left[\mathcal{M}\left(S_{\tau-w_{max}+1,\tau}\right)=y\right]}{\Pr\left[\mathcal{M}\left(\hat{S}_{\tau-w_{max}+1,\tau}\right)=y\right]} \leq e^{\boldsymbol{\epsilon}(u_i)}.$$

Hence, $\mathcal{M}$ satisfies $\boldsymbol{\epsilon}$-PLDP in any window of size w_{max}. According to Lemma IX.1, $\mathcal{M}$ satisfies $(w_{max}\cdot\mathbf{1},\boldsymbol{\epsilon})$-EPLDP. Then, according to Lemma IX.2, $\mathcal{M}$ satisfies $(\boldsymbol{w},\boldsymbol{\epsilon})$-EPLDP. □

2) Proof of Theorems V.3: According to Corollary IX.1 and the process of OPS, we can conclude that both PLPD$^+$ and PLPA$^+$ $\mathcal{M}$ satisfy $\hat{\boldsymbol{\epsilon}}$-PLDP in any window of size w_{opt}, where $\hat{\boldsymbol{\epsilon}}$ satisfies:

$$\hat{\boldsymbol{\epsilon}}(u_i)=\begin{cases}\boldsymbol{\epsilon}(u_i), & \text{if } \boldsymbol{w}(u_i)\leq\hat{w},\\ \boldsymbol{\epsilon}(u_i)/\lceil \boldsymbol{w}(u_i)/\hat{w}\rceil, & \text{otherwize.}\end{cases}$$

According to Lemma IX.1, $\mathcal{M}$ satisfies $(w_{opt}\cdot\mathbf{1},\boldsymbol{\epsilon})$-EPLDP. Then according to Lemma IX.3, we can conclude that PLPD$^+$ and PLPA$^+$ satisfy $(\boldsymbol{w},\boldsymbol{\epsilon})$-EPLDP.

G. Proofs of Theorems for Utility Analysis

Lemma IX.5. *Given a series privacy budget-count set $P=\{(\epsilon_k,n_k)|\epsilon_k\in\tilde{\boldsymbol{\epsilon}},n_k\in\tilde{\boldsymbol{n}},\sum_k n_k=n\}$ with sampling size z from n users, where $\tilde{\boldsymbol{\epsilon}}$ is the total distinct privacy budget set with $\forall\epsilon_i,\epsilon_j\in\tilde{\boldsymbol{\epsilon}},\epsilon_i\neq\epsilon_j$ and $\tilde{\boldsymbol{n}}$ is the all distinct appearing count. (ϵ_k,n_k) indicates that there are n_k users proposing privacy budget requirement ϵ_k. The error upper bound of* FO *process is :*

$$\widetilde{err}_{\text{FO}}(P,z)=\frac{n-z}{z(n-1)}+\frac{n(d-1)}{z\min(\tilde{\boldsymbol{n}})}\cdot\frac{2e^{\min(\boldsymbol{\epsilon})}+d-2}{\left(e^{\min(\boldsymbol{\epsilon})}-1\right)^2}. \tag{16}$$

Let $\hat{n}_{\max(\tilde{\boldsymbol{\epsilon}})}$ be the user number of requiring the maximum privacy budget $\max(\tilde{\boldsymbol{\epsilon}})$, the error upper bound of PFO *process is :*

$$\widetilde{err}_{\text{PFO}}(P,z)=\frac{n-z}{z(n-1)}+\frac{n(d-1)}{z\hat{n}_{\max(\tilde{\boldsymbol{\epsilon}})}}\cdot\frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}. \tag{17}$$

Proof. For the FO case, according to Equation (6) by setting $p_k=\frac{e^{\epsilon_k}}{e^{\epsilon_k}+d-1}, q_k=\frac{1}{e^{\epsilon_k}+d-1}$, and denoting $\lambda_k=\frac{q_k(1-q_k)}{g_k(p_k-q_k)^2}, \mu_k=\frac{1-p_k-q_k}{g_k(p_k-q_k)}$, we can get the error as

$$\begin{aligned}
&err_{\text{FO}}(P,z)\\
=&\frac{n-z}{z(n-1)}+\frac{1/z}{\sum_{k=1}^m 1/\left(d\lambda_k+\mu_k\right)}\\
\leq&\frac{n-z}{z(n-1)}+\frac{1}{zm}\sum_{k=1}^m\left(d\lambda_k+\mu_k\right)\\
=&\frac{n-z}{z(n-1)}+\frac{d-1}{zm}\sum_{k=1}^m\frac{1}{g_k}\left(\frac{2}{e^{\epsilon_k}-1}+\frac{d}{(e^{\epsilon_k}-1)^2}\right)\\
=&\frac{n-z}{z(n-1)}+\frac{d-1}{zm}\sum_{k=1}^m\frac{n}{n_k}\left(\frac{2}{e^{\epsilon_k}-1}+\frac{d}{(e^{\epsilon_k}-1)^2}\right)\\
\leq&\frac{n-z}{z(n-1)}+\frac{n(d-1)}{z\min(\tilde{\boldsymbol{n}})}\cdot\left(\frac{2}{e^{\min(\tilde{\boldsymbol{\epsilon}})}-1}+\frac{d}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}\right)\\
=&\frac{n-z}{z(n-1)}+\frac{n(d-1)}{z\min(\tilde{\boldsymbol{n}})}\cdot\frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}.
\end{aligned} \tag{18}$$

For the PFO case, without loss generation, let $\min(\tilde{\boldsymbol{\epsilon}})=\epsilon_1<\epsilon_2<\cdots<\epsilon_m=\max(\tilde{\boldsymbol{\epsilon}})$. According to Theorem V.1, for users with privacy budget ϵ_k, the error decreases from $\frac{1}{n_k}\cdot\left(\frac{dq_k(1-q_k)}{(p_k-q_k)^2}+\frac{1-p_k-q_k}{p_k-q_k}\right)=\frac{g_k}{n_k}\cdot(d\lambda_k+\mu_k)$ to $\frac{1}{\sum_{j=k}^m n_j}\cdot\left(\frac{dq_k(1-q_k)}{(p_k-q_k)^2}+\frac{1-p_k-q_k}{p_k-q_k}\right)=\frac{g_k}{\sum_{j=k}^m n_j}\cdot(d\lambda_k+\mu_k)$. Namely, λ_k varies to $\lambda'_k=\frac{n_k}{\sum_{j=k}^m n_j}\cdot\lambda_k$ and μ_k varies to $\mu'_k=\frac{n_k}{\sum_{j=k}^m n_j}\cdot\mu_k$. For the error $err_{\text{PFO}}(P,z)$, we have

$$\begin{aligned}
&err_{\text{PFO}}(P,z)\\
=&\frac{n-z}{z(n-1)}+\frac{1/z}{\sum_{k=1}^m 1/\left(d\lambda'_k+\mu'_k\right)}\\
\leq&\frac{n-z}{z(n-1)}+\frac{1}{zm}\sum_{k=1}^m\left(d\lambda'_k+\mu'_k\right)\\
=&\frac{n-z}{z(n-1)}+\frac{d-1}{zm}\sum_{k=1}^m\frac{n}{\sum_{j=k}^m n_j}\left(\frac{2}{e^{\epsilon_k}-1}+\frac{d}{(e^{\epsilon_k}-1)^2}\right)\\
\leq&\frac{n-z}{z(n-1)}+\frac{n(d-1)}{z\hat{n}_{\max(\tilde{\boldsymbol{\epsilon}})}}\cdot\left(\frac{2}{e^{\min(\tilde{\boldsymbol{\epsilon}})}-1}+\frac{d}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}\right)\\
=&\frac{n-z}{z(n-1)}+\frac{n(d-1)}{z\hat{n}_{\max(\tilde{\boldsymbol{\epsilon}})}}\cdot\frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}.
\end{aligned}$$

□

1) Proof of Theorem IV.3 and IV.4:

Proof. Let $P=\{(\epsilon_k,n_k)|\epsilon_k\in\tilde{\boldsymbol{\epsilon}},n_k\in\tilde{\boldsymbol{n}},\sum_k n_k=n\}$ be the distinct privacy budget-count set. At any time slot t, if the sampling size is z, according to Equation (16) in Lemma IX.5, the error upper bound of executing FO is $\frac{n}{z(n-1)}-\frac{1}{n-1}+\frac{n(d-1)}{z\min(\tilde{\boldsymbol{n}})}\cdot\frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}$.

Suppose $\mathcal{M}_{s,t}$ is not private, then the error only comes from sub-mechanism $\mathcal{M}_{r,t}$. We discuss $\mathcal{M}_{r,t}$ in PLPD and PLPA, respectively.

In PLPD, for any new publication at time slot τ, the error is $\widetilde{err}_{FO}(P,z)$, where z is the sampling size at τ. For other skipped time slots, the error is no larger than that of the last new publication. Since the population is distributed to the s new publications in an exponentially decreasing way, the sampling population size sequence is then $n/4,n/8,/\ldots,n/2^{s+1}$. Thus, by denoting $A=\frac{d-1}{\min(\tilde{\boldsymbol{n}})}\cdot\frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}$ we have

$$\begin{aligned}
&\overline{err}_{PLPD,r,t}\\
=&\frac{1}{w_{max}}\cdot\sum_{\tau=1}^{s}\frac{w_{max}}{s}\cdot err_{FO}\left(P,\frac{n}{2^{\tau+1}}\right)\\
\leq&\frac{1}{s}\cdot\sum_{\tau=1}^{s}\widetilde{err}_{FO}\left(P,\frac{n}{2^{\tau+1}}\right)\\
=&\frac{1}{s}\sum_{\tau=1}^{s}\left(\frac{n}{\frac{n}{2^{\tau+1}}(n-1)}-\frac{1}{n-1}+\frac{n}{\frac{n}{2^{\tau+1}}}\cdot A\right)\\
=&\frac{1}{s}\cdot\left(-\frac{s}{n-1}+\left(\frac{1}{n-1}+A\right)\sum_{\tau=1}^{s}2^{\tau+1}\right)\\
=&\frac{1}{s}\cdot\left(-\frac{s}{n-1}+\left(\frac{1}{n-1}+A\right)\cdot 4\cdot(2^s-1)\right)\\
=&\frac{4(2^s-1)}{s}\cdot\left(\frac{1}{n-1}+A\right)-\frac{1}{n-1}.
\end{aligned} \tag{19}$$

In PLPA, since there are s new publications, there are $w_{max}-s$ approximate publications. Because the skipped and nullified publications occur in pairs (i.e., the numbers of skipped publications and nullified publications are equal)

and each new publication corresponds the same number of skipped publications, the numbers of skipped and nullified publications are both $n_{skp} = n_{nlf} = \frac{w_{max}-s}{2\cdot s}$. For each new publication, the population size is $n_{skp} \cdot \frac{n}{2w_{max}} = \frac{(w_{max}-s)n}{4sw_{max}}$. For each skipped publication, its error is no more than the new publication, otherwise it will not skip. Further more, a new publication sampling size lower bound at each skipped publication time slot is increasing from $\frac{n}{2w_{max}}$ in an exponential fashion, namely, they are $\frac{n}{2w_{max}}, \frac{n\cdot 2}{2w_{max}}, \ldots, \frac{n\cdot 2^{n_{skp}}}{2w_{max}}$. For each nullified publication, because we have no information about the underlying statistic and nullification is enforced prior to their arrival, we cannot quantify this kind of error. Thus, we denote the nullified publication error as err_{nlf}. For the error of $\mathcal{M}_{r,t}$ in PLPA, by denoting $A = \frac{d-1}{\min(\tilde{\boldsymbol{n}})} \cdot \frac{2e^{\min(\tilde{\boldsymbol{\epsilon}})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}})}-1\right)^2}$, $B = n_{skp} = \frac{w_{max}-s}{2s}$ and $C = \frac{s}{w_{max}}$, we have

$$
\begin{aligned}
&\overline{err}_{PLPA,r,t}\\
\leq&\frac{1}{w_{max}} \cdot \sum_{\tau=1}^{s} \left(\widetilde{err}_{FO}\left(P, \frac{(w_{max}-s)n}{4sw_{max}}\right) + \sum_{j=1}^{\frac{w_{max}-s}{2s}} \widetilde{err}_{FO}\left(P, \frac{n\cdot 2^j}{2w_{max}}\right) \right.\\
&\left. + \frac{w_{max}-s}{2s}\cdot err_{nlf}\right)\\
=&\frac{s}{w_{max}} \cdot \left(-\frac{1}{n-1} + \frac{4sw_{max}}{(n-1)\cdot(w_{max}-s)} + \frac{4sw_{max}}{(w_{max}-s)}\cdot A - \frac{w_{max}-s}{2s(n-1)} \right.\\
&\left. + \left(\frac{w_{max}}{n-1} + w_{max}\cdot A\right) \cdot \sum_{j=1}^{\frac{w_{max}-s}{2s}} \frac{1}{2^{j-1}} + \frac{w_{max}-s}{2s}\cdot err_{nlf}\right)\\
=&C\cdot\left(B\cdot err_{nlf} - \frac{B+1}{n-1} + \left(\frac{w_{max}}{n-1} + w_{max}\cdot A\right)\cdot\left(2-\frac{1}{2^{B+1}}+\frac{2}{B}\right)\right)\\
=&BC\cdot err_{nlf} - \frac{(B+1)C}{n-1} + \left(\frac{s}{n-1} + s\cdot A\right)\cdot\left(2-\frac{1}{2^{B+1}}+\frac{2}{B}\right).
\end{aligned}
\tag{20}
$$

When sub-mechanism $\mathcal{M}_{s,t}$ is private, in both PLPD and PLPA, there is a sample $|U_{s,t}|$ of fixed sampling size $z_{min} = \lfloor n/(2w_{max})\rfloor$ of users for *dis* calculation. The error in $\mathcal{M}_{s,t}$ will lead to a *false* publication decision, i.e., making $\mathcal{M}_{r,t}$ (1) falsely skip a publication, or (2) falsely perform a new publication. Both of the two cases are due to the error of *dis* calculation, which is $\frac{1}{d^3} err_{FO}(P, z_{min})^2$. Therefore the average error of $\mathcal{M}_{s,t}$ within any window of size w_{max} is

$$
\begin{aligned}
\overline{err}_{s,t} &\leq \frac{1}{d^3}\cdot \widetilde{err}_{FO}(P, z_{min})^2\\
&= \frac{1}{d^3}\left(-\frac{1}{n-1} + \frac{2w_{max}}{n-1} + 2w_{max}A\right)^2.
\end{aligned}
\tag{21}
$$

Thus, the error upper bound at each time slot is $\frac{1}{d^3}\left(-\frac{1}{n-1} + \frac{2w_{max}}{n-1} + 2w_{max}A\right)^2 + \frac{4(2^s-1)}{s}\cdot\left(\frac{1}{n-1}+A\right) - \frac{1}{n-1}$ in PLPD and $\frac{1}{d^3}\left(-\frac{1}{n-1} + \frac{2w_{max}}{n-1} + 2w_{max}A\right)^2 + BC\cdot err_{nlf} - \frac{(B+1)C}{n-1} + \left(\frac{s}{n-1} + s\cdot A\right)\cdot\left(2-\frac{1}{2^{B+1}}+\frac{2}{B}\right)$ in PLPA. □

2) Proof of Theorem V.4 and V.5:

Proof. Let $P_{opt} = \{(\epsilon_k, n_k) | \epsilon_k \in \tilde{\boldsymbol{\epsilon}}_{opt}, n_k \in \tilde{\boldsymbol{n}}_{opt}, \sum_k n_k = n\}$ be the distinct privacy budget-count set after executing OPS. At any time slot t, if the sampling size is z, according to Equation (17) in Lemma IX.5, the error upper bound of executing PFO is $\frac{n}{z(n-1)} - \frac{1}{n-1} + \frac{n(d-1)}{z\tilde{n}_{\max(\tilde{\boldsymbol{\epsilon}}_{opt})}}\cdot\frac{2e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})}-1\right)^2}$.

Similar to the process in utility process (Section IX-G1) of PLPD and PLPA. We first assume the sub-mechanism $\mathcal{M}_{s,t}$ is not private.

In this way, for the error of $\mathcal{M}_{r,t}$ in PLPD$^+$, by setting $A = \frac{d-1}{\tilde{n}_{\max(\tilde{\boldsymbol{\epsilon}}_{opt})}}\cdot\frac{2e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})}+d-2}{\left(e^{\min(\tilde{\boldsymbol{\epsilon}}_{opt})}-1\right)^2}$, we have

$$
\begin{aligned}
&\overline{err}_{PLPD^+,r,t}\\
=&\frac{1}{w_{opt}}\cdot\sum_{\tau=1}^{s_{opt}}\frac{w_{opt}}{s_{opt}}\cdot err_{PFO}\left(P_{opt}, \frac{n}{2^{\tau+1}}\right)\\
\leq&\frac{4(2^{s_{opt}}-1)}{s_{opt}}\cdot\left(\frac{1}{n-1}+A\right) - \frac{1}{n-1}.
\end{aligned}
$$

For the error of $\mathcal{M}_{r,t}$ in PLPD$^+$, by further setting $B = \frac{w_{opt}-s_{opt}}{2s_{opt}}$ and $C = \frac{s_{opt}}{w_{opt}}$, we have

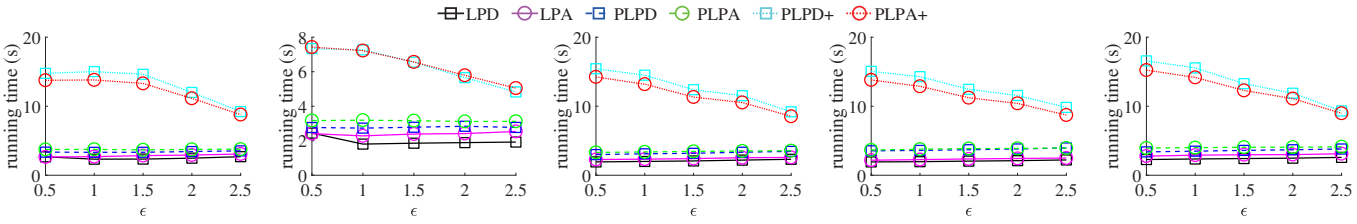
$$
\begin{aligned}
&\overline{err}_{PLPA^+,r,t}\\
\leq&\frac{1}{w_{opt}}\cdot\sum_{\tau=1}^{s_{opt}}\left(\widetilde{err}_{PFO}\left(P_{opt}, \frac{(w_{opt}-s_{opt})n}{4sw_{opt}}\right) + \sum_{j=1}^{\frac{w_{opt}-s_{opt}}{2s_{opt}}}\widetilde{err}_{PFO}\left(P, \frac{n\cdot 2^j}{2w_{opt}}\right)\right.\\
&\left. + \frac{w_{opt}-s_{opt}}{2s_{opt}}\cdot err_{nlf}\right)\\
=&BC\cdot err_{nlf} - \frac{(B+1)C}{n-1} + \left(\frac{s_{opt}}{n-1} + s_{opt}\cdot A\right)\cdot\left(2-\frac{1}{2^{B+1}}+\frac{2}{B}\right).
\end{aligned}
$$

When $\mathcal{M}_{s,t}$ is private, we can get the error as

$$
\begin{aligned}
\overline{err}_{s,t} &\leq \frac{1}{d^3}\cdot\widetilde{err}_{PFO}(P, z_{opt})^2\\
&= \frac{1}{d^3}\left(-\frac{1}{n-1} + \frac{2w_{opt}}{n-1} + 2w_{opt}A\right)^2.
\end{aligned}
$$

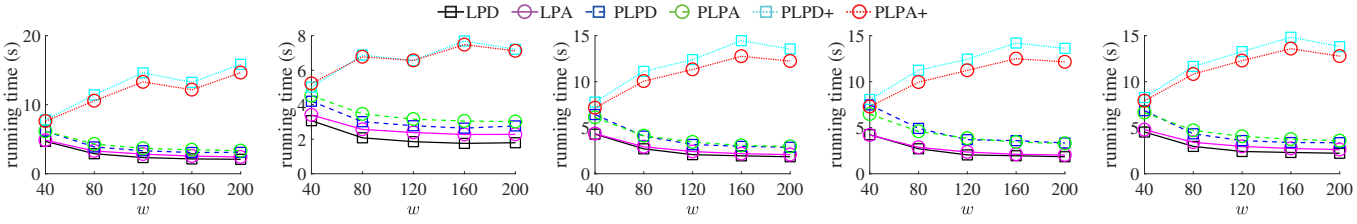
Therefore, the error upper bound is $\frac{1}{d^3}\left(-\frac{1}{n-1} + \frac{2w_{opt}}{n-1} + 2w_{opt}A\right)^2 + \frac{4(2^{s_{opt}}-1)}{s_{opt}}\cdot\left(\frac{1}{n-1}+A\right) - \frac{1}{n-1}$ in PLPD$^+$, and $\frac{1}{d^3}\left(-\frac{1}{n-1} + \frac{2w_{opt}}{n-1} + 2w_{opt}A\right)^2 + BC\cdot err_{nlf} - \frac{(B+1)C}{n-1} + \left(\frac{s_{opt}}{n-1} + s_{opt}\cdot A\right)\cdot\left(2-\frac{1}{2^{B+1}}+\frac{2}{B}\right)$ in PLPA$^+$. □

H. Extra Experiment Results



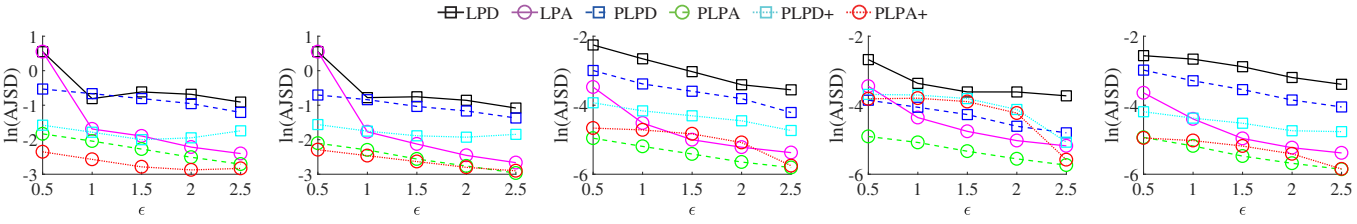
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Fig. 8: The running time with ϵ varied.



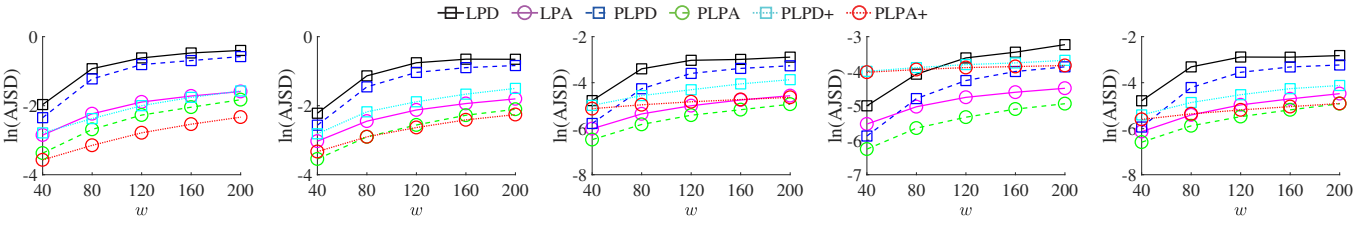
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Fig. 9: The running time with w varied.



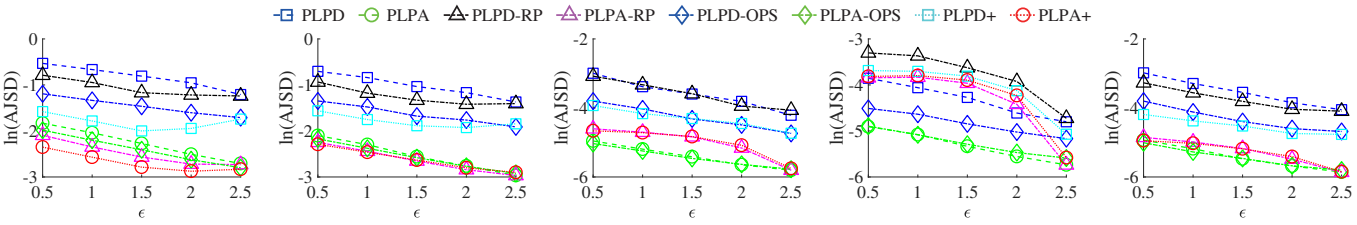
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Fig. 10: $AJSD$ with ϵ varied.



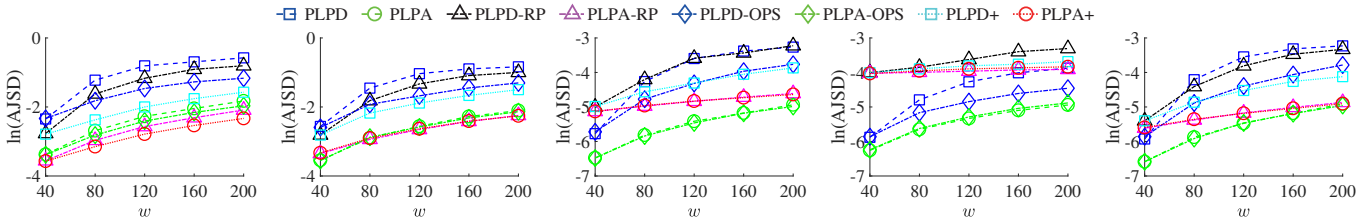
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Fig. 11: $AJSD$ with w varied.



(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Fig. 12: The ablation study of $AJSD$ with ϵ varied.



(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Fig. 13: The ablation study of $AJSD$ with w varied.