Infinite Stream Estimation under Personalized w-Event Privacy

Leilei Du[1], Peng Cheng[1], Lei Chen[2,3], Heng Tao Shen[4,5], Xuemin Lin[6], Wei Xi[7]
[1]ECNU, Shanghai, China; [2]HKUST (GZ), Guangzhou, China; [3]HKUST, Hong Kong SAR, China;
[4]Tongji University, Shanghai, China; [5]UESTC, Chengdu, China; [6]SJTU, Shanghai, China; [7]XJTU, Xi'an, China
leileidu@stu.ecnu.edu.cn; pcheng@sei.ecnu.edu.cn; leichen@cse.ust.hk;
shenhengtao@hotmail.com; xuemin.lin@gmail.com; xiwei@xjtu.edu.cn

ABSTRACT

Streaming data collection is indispensable for stream data analysis, such as event monitoring. However, publishing these data directly leads to privacy leaks. w-event privacy is a valuable tool to protect individual privacy within a given time window while maintaining high accuracy in data collection. Most existing w-event privacy studies on infinite data stream only focus on homogeneous privacy requirements for all users. In this paper, we propose personalized w-event privacy protection that allows different users to have different privacy requirements in private data stream estimation. Specifically, we design a mechanism that allows users to maintain constant privacy requirements at each time slot, namely Personalized Window Size Mechanism (PWSM). Then, we propose two solutions to accurately estimate stream data statistics while achieving $\boldsymbol{w}$-event level $\boldsymbol{\epsilon}$ personalized differential privacy ($(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP), namely Personalized Budget Distribution (PBD) and Peronalized Budget Absorption (PBA). PBD always provides at least the same privacy budget for the next time step as the amount consumed in the previous release. PBA fully absorbs the privacy budget from the previous k time slots, while also borrowing from the privacy budget of the next k time slots, to increase the privacy budget for the current time slot. We prove that both PBD and PBA outperform the state-of-the-art private stream estimation methods while satisfying the privacy requirements of all users. We demonstrate the efficiency and effectiveness of our PBD and PBA on both real and synthetic data sets, compared with the recent uniformity w-event approaches, Budget Distribution (BD) and Budget Absorption (BA). Our PBD achieves 68% less error than BD on average on real data sets. Besides, our PBA achieves 24.9% less error than BA on average on synthetic data sets.

PVLDB Reference Format:
Leilei Du, Peng Cheng, Lei Chen, Heng Tao Shen, Xuemin Lin, and Wei Xi. Infinite Stream Estimation under Personalized w-Event Privacy. PVLDB, 14(1): XXX-XXX, 2024.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/dulei715/DynamicWEventCode.

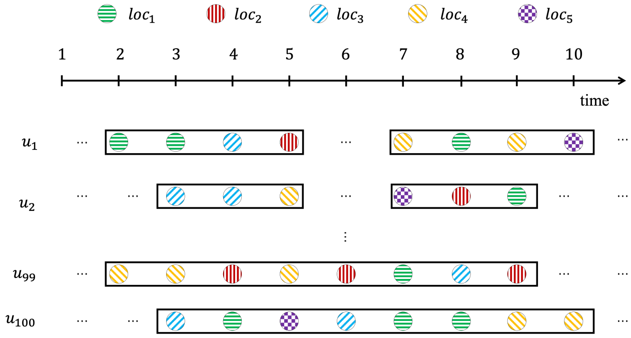


Figure 1: Different event window sizes for different time slots.

1 INTRODUCTION

With the popularity of smart devices and high-quality wireless networks, people can easily access the internet and utilize online services. They continuously report data to platforms and receive services like log stream analysis [34], event monitoring [19], and video querying [27]. To provide better services, these platforms collect data and conduct real-time analysis over aggregated data streams.

However, collecting stream data directly poses severe privacy risks, causing users to refuse communication with platforms. For instance, an AIDS patient may decline to participate in an investigation due to privacy concerns [18]. To resolve this conflict, differential privacy (DP) is proposed to protect individual privacy while ensuring accurate data estimation [11].

Recently, w-event privacy based on DP has emerged for private stream data collection and analysis [29, 30, 33]. It effectively protects the privacy of w consecutive related events while offering accurate stream statistics. However, different users may have different privacy requirements. For instance, entertainers may be reluctant to reveal too much about their locations (i.e., large w-event size), while street artists may be willing to expose their locations (i.e., small w-event size) for more attention. Thus, if we fix the window size w for all users, it is hard to make everyone satisfied.

We illustrate an example of online car-hailing shown in Figure 1.

Example 1. *Suppose there are* 100 *drivers* $U = \{u_1, ..., u_{100}\}$ *who provide their locations within* $\{loc_1, ..., loc_8\}$ *at each time slot. For any driver* u_i*, he/she is protected with* w_i*-event privacy means that his/her events' locations is protected through satisfying* ϵ_i*-DP within at least* w_i *consecutive time slots, where* ϵ_i *is a parameter indicating the strength of the privacy protect required by* u_i*. For instance,* u_1 *wants to protect his/her location sequence within any* 4 *consecutive time slots. Besides,* u_{99} *and* u_{100} *want to protect their location sequences within any* 8 *consecutive time slots. Suppose for each* $u_i \in U \backslash \{u_{99}, u_{100}\}$*, the window size is no more than* 4*. To satisfy all drivers' privacy needs, according to traditional* w*-event*

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

privacy, we need to set the event window size as the maximal value (i.e., $w = 8$), and make full use of the privacy budget to achieve high utility while satisfying 8*-event privacy. Let AE_{avg} denote the average square error at each time slot, defined as the variance when adding Laplace noise (i.e., $AE_{avg} = 2b^2 = 2 \times \left(\frac{1}{\epsilon/w}\right)^2$). Suppose his total privacy budget ϵ is* 1 *and the platform adopts the Uniform method [22]. Then for this example, under* 8*-event privacy, the average square error at each time slot is $AE_{avg} = 2 \times (\frac{w}{\epsilon})^2 = 128$. However, it is not necessary for the first* 98 *drivers to set the window size as* 8 *(only achieving* 8*-event privacy). If we set the window size as $w = 4$ and use the threshold method [20] (or the sample method [20]), then we can get $AE_{avg} \approx 2 \times (\frac{w}{\epsilon})^2 = 32$, which is much less than the error of traditional* 8*-event privacy.*

In this paper, we define the Personalized w-event Private Publishing for Infinite Data Streams (PWPP-IDS) problem to model personalized requirements in stream data publication. To solve PWPP-IDS, there are two challenges: 1) effectively unifying the privacy budget across all users into a single value to maximize publication utility; 2) effectively distribute each user's personalized privacy budget to their personalized window size to maximize publication utility.

To achieve higher publication utility, we solve PWPP-IDS with the centralized DP model [11], where a single centralized privacy budget is needed for publishing statistics at each time slot. Different users may have their personalized and different privacy budgets. If we need to satisfy the privacy requirements of every user, we need to select the minimum privacy budget among them, which can result in the lowest utility. *How to use a privacy budget higher than the minimum one to achieve higher utility while satisfying the privacy requirement of user with the minimum privacy?* It seems unachievable at a glance. We solve this challenge through elaborately applying the Sampling Mechanism [20]. Our method theoretically guarantees that even though the selected unified privacy budget is higher than the minimum privacy budget, no privacy leakages for any users exist.

Intuitively, time slots with higher changing rates contain more information and thus are more important. To maximize utility, we need to allocate more privacy budgets to publications at these important time slots while approximating others. *How to identify important time slots and allocate privacy budgets to achieve maximum publication utility?* To address this challenge, we design two methods, namely Personalized Budget Distribution (PBD) and Peronalized Budget Absorption (PBA), to handle them. PBD takes an optimistic approach, assuming few publications per window and thus allocating larger budget portions to each publication. PBA, in contrast, assumes that the stream data will have low changing rate thus can skip or approximate a large portion of publications. Thus, it maximizes current publication accuracy by borrowing unused budget from skipped publications while nullifying future time slot budgets, enabling effective approximation of subsequent publications. We demonstrate that both PBD and PBA satisfy $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP and provide their average error upper bounds. We summarize our contributions as follows.

- We formally define personalized $\boldsymbol{w}$-event level $\boldsymbol{\epsilon}$-Personalized Differential Privacy for PWPP-IDS in Section 3.
- We propose a basic mechanism, Personalized Window Size Mechanism (PWSM), and two methods, namely Personalized Budget Distribution (PBD) and Peronalized Budget Absorption (PBA), to support personalized $\boldsymbol{w}$-event privacy with theoretical analyses in Section 4.
- We test our methods on both real and synthetic data sets to demonstrate their efficiency and effectiveness in Section 5.

2 RELATED WORK

We classify the related work in the area of data stream estimation under differential privacy and non-uniformity differential privacy.

2.1 Data Stream Estimation under Differential Privacy

Based on the privacy model, there are two types of data stream estimation methods: centralized differential privacy [11] (CDP) based methods and local differential privacy [4] (LDP) based methods.

Data Stream Estimation under CDP. Dwork et al. first address the problem of Differential Privacy (DP) on data streams [13]. They define two types of DP levels: *event-level differential privacy* (event-DP) and *user-level differential privacy* (user-DP).

In event-DP, each single event is hidden in statistic queries. Dwork et al. focus on the finite event scenarios and propose a binary tree method to achieve high statistical utility while maintaining event-DP [13]. Chan et al. extend it to infinite cases, and produce partial summations for binary counting [7]. Dwork et al. introduce a cascade buffer counter that updates adaptively based on stream density [12]. Bolot et al. propose *decayed privacy* which reduces the privacy costs for past data [6]. Chen et al. develop PeGaSus, a perturb-group-smooth framework for multiple queries under event-DP [8]. However, event-DP assumes all element in a stream are independent, making it unsuitable for correlated data stream publishing.

In user-DP, all events for each user are hidden in statistic queries. Fan et al. propose the FAST algorithm with a sampling-and-filtering framework, counting finite stream data under user-DP [17]. Cummings et al. address heterogeneous user data, estimating population-level means while achieving user-DP [9]. However, they only consider finite data. Offering user-DP for infinite data requires infinite perturbation, leading to poor long-term utility [22].

To bridge the gap between event-DP and user-DP, Kellaris et al. propose w-event DP for infinite streams [22]. This ensures ϵ-DP for any group of events within a time window of size w. They introduce two methods, *Budget Distribution* and *Budget Absorption*, to optimize privacy budget use and estimate statistics effectively. However, neither method handles stream data with significant changes. Wang et al. apply the w-event concept to the FAST method, proposing a multi-dimensional stream release mechanism called *ResueDP*, which achieves accurate estimation for both rapid and slow data stream changes [30]. A limitation of all these methods is their reliance on a trusted server to ensure privacy.

Data Stream Estimation under LDP. To overcome the dependence on a trusted server, LDP [4] has recently been proposed and adopted by many major companies such as Microsoft, Apple and Google. Erlingsson et al. introduce RAPPOR to estimate finite streams under LDP [16]. They design a two-layer randomized response mechanism (i.e., permanent randomized response and instantaneous randomized response) to protect each individual's data. However, RAPPOR is limited to uncorrelated stream data. To address the problem of correlated time series data, Erlingsson et al. develop a new privacy model that introduces *shuffling* to amplify the LDP privacy level [15]. However, this model only suits finite stream data. Joseph et al. propose THRESH for evolving data under LDP [21], which consumes privacy budget at global update time slots selected by users' LDP voting. However, it is not applicable to infinite streams as it assumes a fixed number of global updates. Wang et al. extend event-level privacy from CDP to LDP and design the efficient ToPL method under event LDP [31]. Nevertheless, event-level LDP focuses solely on event-level privacy, lacking privacy protection for correlated data in streams. Bao et al. propose an (ϵ, δ)-LDP method (called CGM) for finite streaming data collection using the analytic Guassian mechanism, but requires periodic privacy budget renewal [3]. Ren et al. introduce LDP-IDS for infinite streaming data collection and analysis under w-event LDP [29]. They propose two budget allocation methods and two population allocation methods, bridging the gap between event LDP and user LDP while improving estimation accuracy. However, all these methods cannot be adopted to support personalized event window sizes.

2.2 Non-Uniformity Differential Privacy

Recently, some studies address the non-uniform privacy requirements among items (table columns) or records (table rows) [28].

Alaggan et al. first examine scenarios where each database instance comprises a single user's profile [1]. They focus on varying privacy requirements for different items and formally define Heterogeneous Differential Privacy (HDP).

Jorgensen et al. investigate the privacy preservation for individual rows, introducing Personalized Differential Privacy (PDP) [20]. They design two mechanisms leveraging non-uniform privacy requirements to achieve better utility than standard uniform DP. Kotsogiannis et al. recognize that different data have different sensitivity, then define One-side Differential Privacy (OSPD) and propose algorithms that truthfully release non-sensitive record samples to enhance accuracy in DP-solutions [23].

Andrés et al. introduce a novel non-uniform privacy concept called Geo-Indistinguishability (Geo-I), where the privacy level for any point increases as the distance to this point decreases [2]. Wang et al. [32] and Du et al. [10] explore PDP in spatial crowdsourcing, and develop highly effective private task assignment methods to satisfy diverse workers' privacy and utility requirements. Liu et al. investigate HDP in federated learning [26]. They assume different clients hold DP budget and divide them into private and public parts, then propose two methods to project the "public" clients' models into "private" clients' models to improve the joint model's utility. However, all above studies are not suitable for stream data.

In this paper, we propose Personalized Window Size Mechanism (PWSM) with two implementation methods: Personalized Budget Distribution (PBD) and Peronalized Budget Absorption (PBA). Our approach extends traditional w-event privacy mechanisms by introducing $\boldsymbol{\epsilon}$-Personalized Differential Privacy methods to support personalized privacy requirements. This enhancement enables our mechanism and methods to handle both infinite correlated data streams and personalized privacy requirements, building upon the foundations of traditional w-event privacy mechanisms.

Table 1: Summary for related work.

Model Types		Methods	Is infinite and correlated	Is personalized privacy
Centralized DP	event-level privacy	Finite B-tree [13]	✗	✗
		Infinite B-tree [7]	✗	✗
		Adaptive-density Counter [12]	✗	✗
		Decayed Privacy [6]	✗	✗
		PeGaSus [8]	✗	✗
	user-level privacy	FAST [17]	✓	✗
		Private heterogeneous mean estimation [9]	✓	✗
	w-event privacy	BD & BA [22]	✓	✗
		ResuseDP [30]	✓	✗
Local DP	event-level privacy	RAPPOR [16]	✗	✗
		ToPL [31]	✗	✗
	user-level privacy	Shuffling LDP [15]	✓	✗
		THRESH [21]	✓	✗
		CGM [3]	✓	✗
	w-event privacy	LDP-IDS [29]	✓	✗
Item heterogeneous		HDP [1]	✗	✗
Record heterogenous		PDP [20]	✗	✓
		OSDP [23]	✗	✓
		Geo-I [2]	✗	✓
		PWSM, VPDM [32]	✗	✓
		PUCE, PGT [10]	✗	✓
		PFA, PFA+ [26]	✗	✓
Our mechanisms			✓	✓

3 PROBLEM SETTINGS

In this section, we first introduce key concepts, including data streams. Next, we present the new definition of $\boldsymbol{w}$-event $\boldsymbol{\epsilon}$ personalized DP. Finally, we provide the problem definition: Personalized w-event Private Publishing for Infinite Data Streams (PWPP-IDS). Table 2 outlines the notations used throughout this paper.

3.1 Data Stream

Definition 1. (Data Stream). Let $D_t \in \mathcal{D}$ be a database with d columns and n rows (each row representing a user) at t-th time slot. The infinite database sequence $S = [D_1, D_2, \ldots]$ is called a data stream, where $S[t]$ is the t-th element in S (i.e., $S[t] = D_t$).

For any data stream S, its substream between time slot t_l and t_r (where $t_l < t_r$) is noted as $S_{t_l,t_r} = [D_{t_l}, D_{t_l+1}, \ldots, D_{t_r}]$. For $t_l = 1$, we denote $S_t = [D_1, D_2, \ldots, D_t]$ and call it the *stream prefix* of S.

Definition 2. (Data Stream Count Publishing). Let $Q : \mathcal{D} \rightarrow \mathbb{R}^d$ be a count query. Then, $Q(S[t]) = Q(D_t) = \boldsymbol{c}_t$ is the count data to be published at time slot t, where $\boldsymbol{c}_t(j)$ represents the count of the j-th column of D_t. The infinite count data series $[\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots]$ is called a data stream count publishing.

3.2 $\boldsymbol{w}$-event level $\boldsymbol{\epsilon}$-Personalized DP

Definition 3. (w-neighboring stream prefixes [7, 22]). 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 $S_t[k]$, $S'_t[k]$ such that $k \leq t$ and $S_t[k] \neq S'_t[k]$, it holds that $S_t[k]$ and $S'_t[k]$ are neighboring [22] in centralized DP, and
(2) for each $S_t[k_1], S_t[k_2], S'_t[k_1], S'_t[k_2]$ with $k_1 < k_2, S_t[k_1] \neq S'_t[k_1]$ and $S_t[k_2] \neq S'_t[k_2]$, it holds that $k_2 - k_1 + 1 \leq w$.

Definition 4. ($\boldsymbol{w}$-event level ϵ-Personalized DP, $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP). Let $\mathcal{M}$ be a mechanism that takes a stream prefix of arbitrary size as input. Let O be the set of all possible outputs of $\mathcal{M}$. Given a universe of users $U = \{u_1, u_2, \ldots, u_{|U|}\}$, then $\mathcal{M}$ is $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP if $\forall O \subseteq O$,

Table 2: Notations.

Notations	Description
$\mathcal{D}$	the database domain
D_t	a database at time slot t
S	a data stream
u_i	the i-th user
$\boldsymbol{x}_{i,t}$	u_i's data at time slot t
$\boldsymbol{c}_t$	a real statistical histogram at time slot t
$\boldsymbol{r}_i$	an estimation statistic histogram at time slot t
w_i	u_i's privacy window size
ϵ_i	u_i's privacy budget

$\forall w_i \in \boldsymbol{w}$ and $\forall S_t, S'_t$ satisfying $S_t \sim_{w_i} S'_t$,

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

where $u_i \in U$ requires w_i-event level window size, and ϵ_i denotes u_i's privacy budget requirement for the w_i events.

We denote the pair (w_i, ϵ_i) as u_i's *privacy requirement*. Specifically, when $w = 1$, we call it $\boldsymbol{\epsilon}$-Personalized Differential Privacy, $\boldsymbol{\epsilon}$-PDP [20].

3.3 Definition of PWPP-IDS

Given a data stream S, the server obtains the data stream count publishing as $\boldsymbol{c} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots]$. To protect user privacy, however, the server only receives the obfuscated data stream S' and publishes the estimation data stream count (i.e., estimation count) $\boldsymbol{r} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots]$. We present our problem definition as follows.

Definition 5. (PWPP-IDS). Given a user set $U = \{u_1, u_2, ..., u_n\}$, each u_i holds a privacy requirement pair (w_i, ϵ_i) and a series data $\boldsymbol{x}_{i,t}$ for $t \in \mathbb{N}^+$. All the $\boldsymbol{x}_{i,t}$ for $u_i \in U$ at time slot t form D_t. All the D_t form an infinite data stream $S = [D_1, D_2, \ldots]$. PWPP-IDS is to publish an obfuscated histogram $\boldsymbol{r} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots]$ of S at each time slot t achieving $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP with the error between $\boldsymbol{r}$ and $\boldsymbol{c}$ minimized, namely $\forall T \in \mathbb{N}^+$:

$$\begin{aligned} \min_{\epsilon_\theta} \quad & \sum_{t \in [T]} \|\boldsymbol{r}_t - \boldsymbol{c}_t\|_2^2 \\ s.t. \quad & \sum_{k=\min(t-w_i+1,1)}^{t} \epsilon_{i,k} \leq \epsilon_i, \quad \forall u_i \in U \end{aligned}$$

where $\epsilon_{i,k}$ indicates the privacy budget at time slot k.

4 PERSONALIZED WINDOW SIZE MECHANISM

In this section, we analyze the errors in reporting obfuscated data stream counts and introduce Optimal Budget Selection (OBS) method to minimize these errors. We then propose Personalized Window Size Mechanism (PWSM) to address PWPP-IDS. The core idea of PWSM is to select the optimal privacy budget $\epsilon_{opt}(t)$ at each time slot t and report obfuscated count results that satisfy $\epsilon_{opt}(t)$-DP.

4.1 Reporting Errors

For each time slot, we use the Sampling Mechanism (SM) [20] to satisfy all users' privacy requirements (i.e., achieving $\boldsymbol{\epsilon}$-PDP). SM consists of two steps: *sample* (SM_s) and *disturb* (SM_d). In SM_s, the server first sets a privacy budget threshold ϵ_θ, then constructs a sampling subset D_S by appending items x_i with $\epsilon_i \geq \epsilon_\theta$ to D_S, while sampling other items x_j with $\epsilon_j < \epsilon_\theta$ at a probability of $p_j = \frac{e^{\epsilon_j}-1}{e^{\epsilon_\theta}-1}$. In SM_d, the server employs a DP mechanism (e.g., the Laplace Mechanism) to report an obfuscated result that achieves ϵ_θ-DP.

SM introduces two types of errors: *sampling error* and *noise error*. At each time slot t, given a privacy budget threshold ϵ_θ, the total data reporting error is $err(\epsilon_\theta) = err_s(\epsilon_\theta) + err_{dp}(\epsilon_\theta)$. Here, $err_s(\epsilon_\theta)$ represents the *sampling error* from sampling users with privacy budgets below ϵ_θ, while $err_{dp}(\epsilon_\theta)$ represents the *noise error* from adding noise to achieve ϵ_θ-DP. Next, we introduce these sampling and noise errors in detail.

Definition 6. (Sampling Error [20]). Given a privacy budget threshold ϵ_θ and m kinds of privacy budget requirements $\tilde{\epsilon}_1, \tilde{\epsilon}_2, \ldots, \tilde{\epsilon}_m$ from n users with $\tilde{\epsilon}_i < \tilde{\epsilon}_j$ for $i < j$ and $i, j \in [m]$ where $\tilde{\epsilon}_i$ is declared by n_i users ($\sum_{i=1}^{m} n_i = n$), the sampling error $err_s(\epsilon_\theta)$ is defined as

$$\begin{aligned} err_s(\epsilon_\theta) &= Var(count(\boldsymbol{r}_t)) + bias(\boldsymbol{r}_t)^2 \\ &= \sum_{\tilde{\epsilon}_i < \epsilon_\theta} n_i p_i (1 - p_i) + \left(\sum_{\tilde{\epsilon}_i < \epsilon_\theta} n_i (1 - p_i) \right)^2, \end{aligned} \tag{1}$$

where $p_i = \frac{e^{\tilde{\epsilon}_i}-1}{e^{\epsilon_\theta}-1}$.

Definition 7. (Noise Error). The noise error $err_{dp}(\epsilon_\theta)$ is defined as the error of the Laplace mechanism, namely,

$$err_{dp}(\epsilon_\theta) = \frac{2}{\epsilon_\theta^2}. \tag{2}$$

Various metrics exist to measure the errors of Laplace mechanisms for noise error, including variance [20, 29], scale [14, 22], and (α, β)-usefulness [5, 14]. In this work, we employ variance as our metric [20].

Based on Equations (1) and (2), we can observe that err_s depends on n_i, $\tilde{\epsilon}_i$ and ϵ_θ, and is independent of $\boldsymbol{r}_t$. Similarly, err_{dp} depends on ϵ_θ, and is independent of $\boldsymbol{r}_t$.

4.2 Optimal Budget Selection

Given the privacy budget requirements $(\epsilon_{1,t}, \epsilon_{2,t}, \ldots, \epsilon_{n,t})$ of n users, we can determine the frequency of each privacy budget requirement and select the optimal ϵ_θ that minimizes the data reporting error err. This process is detailed in Algorithm 1.

Taking n privacy budgets as input, the Optimal Budget Selection (OBS) algorithm counts the different privacy budgets. Assume there are $\tilde{n}$ distinct privacy budgets, with n_k users requiring $\tilde{\epsilon}_k$ for $k \in [\tilde{n}]$. Let $\tilde{\epsilon}$ be the set of different privacy budget and N be their corresponding frequencies (Lines 1-2). Then, OBS finds the minimum reporting error err_{min} (lines 4-8). Specifically, it iterates over all $\tilde{\epsilon}_k \in \tilde{\boldsymbol{\epsilon}}$ and selects the value $\tilde{\epsilon}_k$ with the smallest $err = err_s(\tilde{\epsilon}_k) + err_{dp}(\tilde{\epsilon}_k)$ as the optimal privacy budget ϵ_{opt} with the minimal error err_{min} .

Example 2 (Running Example of the OBS Algorithm). *Suppose we have* 10 *privacy budgets as input:* $\boldsymbol{\epsilon} = $ *(0.1, 0.4, 0.4, 0.1, 0.4, 0.4, 0.8, 0.8, 0.8, 0.4). OBS first determines* $\tilde{\boldsymbol{\epsilon}} = $ *(0.1, 0.4, 0.8),* $\tilde{n} = |\tilde{\boldsymbol{\epsilon}}| = 3$, *and* $N = $ *(2, 5, 3). Based on these statistics, OBS iterates through the* 3 *privacy budgets in* $\tilde{\boldsymbol{\epsilon}}$ *and calculates their relative errors:* $err_1 = 0 + \frac{2}{0.1^2} = 200$, $err_2 = 2 \times \frac{e^{0.1}-1}{e^{0.4}-1} \times (1 - \frac{e^{0.1}-1}{e^{0.4}-1}) + (2 \times (1 - \frac{e^{0.1}-1}{e^{0.4}-1}))^2 + \frac{2}{0.4^2} = 15.31$ *and* $err_3 = 2 \times \frac{e^{0.1}-1}{e^{0.8}-1} \times (1 - \frac{e^{0.1}-1}{e^{0.8}-1}) + 5 \times \frac{e^{0.4}-1}{e^{0.8}-1} \times (1 - \frac{e^{0.4}-1}{e^{0.8}-1}) + (2 \times (1 - \frac{e^{0.1}-1}{e^{0.4}-1}) + 5 \times (1 - \frac{e^{0.4}-1}{e^{0.8}-1}))^2 + \frac{2}{0.8^2} = 27.73$. *Finally, OBS returns* 0.4 *with the minimal error* 15.31.

Algorithm 1: Optimal Budget Selection (OBS)

```
Input: personalized privacy budget set $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$
Output: $\epsilon_{opt}$, $err_{min}$
Set $\tilde{\boldsymbol{\epsilon}} = (\tilde{\epsilon}_1, \tilde{\epsilon}_2, \ldots, \tilde{\epsilon}_{\tilde{n}})$ as the set of different $\epsilon \in \boldsymbol{\epsilon}$;
Set $N = (n_1, n_2, \ldots, n_{\tilde{n}})$ as the corresponding frequency of $\tilde{\epsilon}_k \in \tilde{\boldsymbol{\epsilon}}$;
Initialize $err_{min}$ as the upper bound of error value;
for $\tilde{\epsilon}_k \in \tilde{\boldsymbol{\epsilon}}$ do
    $err = err_s(\tilde{\epsilon}_k) + err_{dp}(\tilde{\epsilon}_k)$;
    if $err < err_{min}$ then
        $err_{min} = err$;
        $\epsilon_{opt}$ as $\tilde{\epsilon}_k$;
return $\epsilon_{opt}$, $err_{min}$
```

4.3 Personalized Window Size Mechanism

Budget division [22, 29] is a traditional framework for publishing private stream data under w-event privacy. It comprises two basic methods, namely *Uniform* and *Sampling* and two adaptive methods, namely *Budget Distribution* (BD) and *Budget Absorption* (BA). The adaptive methods leverage the stream's variation tendency, resulting in more accurate obfuscated estimations.

In this subsection, we extend the adaptive budget division framework to a personalized context and introduce our Personalized Window Size Mechanism (PWSM). Based on PWSM, we propose two methods: namely Personalized Budget Distribution (PBD) and Peronalized Budget Absorption (PBA).

In real applications, users must specify their privacy budgets and window sizes. System administrators first define a discretized privacy budget range (e.g., $\{0.1, 0.5, 0.9\}$) and a window size range (e.g., $\{40, 80, 120\}$). Then, they can map ascending privacy budget values to descending privacy budget levels (e.g., High, Medium, Low) and ascending window size values to ascending window size levels (e.g., Small, Medium, Large). Users can then select both a privacy budget level and a window size level based on their needs and past experience. After users submit these selections, the server converts them into the corresponding values.

As shown in Algorithm 2, the PWSM algorithm takes three inputs: the historical estimation *His*, personalized privacy budget $\boldsymbol{\epsilon}$, and personalized window size set $\boldsymbol{w}$. Both $\boldsymbol{\epsilon}$ and $\boldsymbol{w}$ are fixed values collected from all users during system initialization. PWSM first calculates all users' privacy budget resources $\boldsymbol{\epsilon}_t$ at the current time slot t to satisfy $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP (line 1). It then divides $\boldsymbol{\epsilon}_t$ into two parts: $\boldsymbol{\epsilon}_t^{(1)}$ and $\boldsymbol{\epsilon}_t^{(2)}$ (line 2). Using $\boldsymbol{\epsilon}_t^{(1)}$, PWSM calculates the dissimilarity *dis* between the current count value and the last reported one by invoking the SM method [20] (line 3). Next, it sets the change threshold as the reporting error *err* calculated with $\boldsymbol{\epsilon}_t^{(2)}$ (line 4). Finally, PWSM adaptively decides whether to publish a new obfuscated estimation or skip (i.e., use the last published one to approximate) by comparing *dis* to $\sqrt{err}$ (lines 5-9).

To determine whether to publish a new obfuscated estimation or skip, we need to introduce a judgment measure called the *personalized private dissimilarity measure*.

Algorithm 2: PWSM

```
  Input: historical estimation $His$, EPDP privacy requirement $(\boldsymbol{w}, \boldsymbol{\epsilon})$
  Output: $\boldsymbol{r}$
1 Get the current privacy budgets $\boldsymbol{\epsilon}_t$ of all users as $\boldsymbol{\epsilon}$ and $\boldsymbol{w}$;
2 Divide $\boldsymbol{\epsilon}_t$ into two parts $\boldsymbol{\epsilon}_t^{(1)}$ and $\boldsymbol{\epsilon}_t^{(2)}$ satisfying $\boldsymbol{\epsilon}_t = \boldsymbol{\epsilon}_t^{(1)} + \boldsymbol{\epsilon}_t^{(2)}$;
3 Calculate dissimilarity $dis$ between current estimation and the last
    estimation by $SM(\boldsymbol{\epsilon}_t^{(1)})$;
4 Calculate the reporting error $err$ of current estimation by $OBS(\boldsymbol{\epsilon}_t^{(2)})$;
5 if $dis > \sqrt{err}$ then
6     Calculate current estimation $\boldsymbol{r}$ by $SM(\boldsymbol{\epsilon}_t^{(2)})$;
7 else
8     Set current estimation $\boldsymbol{r}$ as the last reporting value;
9 return $\boldsymbol{r}$;
```

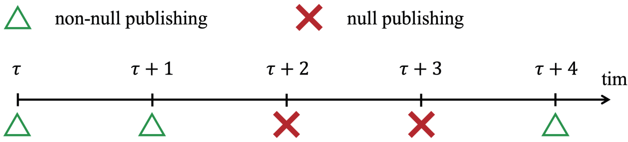


Figure 2: A non-null publishing example.

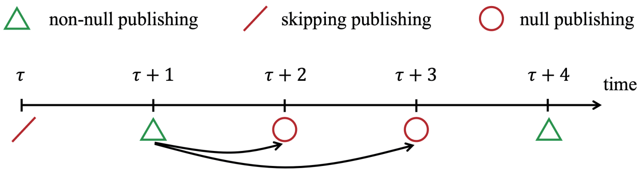


Figure 3: A nullified time slot example.

Personalized Private Dissimilarity Measure. The personalized dissimilarity measure dis^* is defined as the absolute error between the true statistic $\tilde{c}_t$ under SM_s (i.e., the *sample* step of SM) at current time slot t and the last publishing $\boldsymbol{r}_l$, namely,

$$dis^* = \frac{1}{d}\sum_{k=1}^{d} |\tilde{c}_t[k] - \boldsymbol{r}_l[k]|. \tag{3}$$

Our goal is to privately obtain the personalized dissimilarity dis^* using the optimal privacy budget ϵ_{opt} calculated through *OBS* algorithm. The personalized private dissimilarity measure *dis* is then defined as:

$$dis = dis^* + Lap\left(\frac{1}{d \cdot \epsilon_{opt}}\right), \tag{4}$$

where *Lap* represents the Laplace mechanism.

Next, we introduce two methods for PWSM: Personalized Budget Distribution (PBD) and Peronalized Budget Absorption (PBA).

4.4 Personalized Budget Distribution and Peronalized Budget Absorption

Based on the framework PWSM in Algorithm 2, the reporting value is either a newly disturbed statistic value or an approximation from the previous reporting value. We now introduce the following notations to clarify this process.

Basic notations. For a sequence of publications $(r_1, r_2, ..., r_t)$ of length t, we define a "null publishing" as an approximation value and "non-null publishing" as a new value. For any time slot $2 \leq \tau \leq t$, we refer to $r_{\tau-1}$ as the last reporting value (or last publishing) of time slot τ. In the sequence $(r_1, r_2, ..., r_\tau)$, we define the most recent non-null publishing r_l where $l < \tau$ as the last non-null publishing. For example in Figure 2, the publications at time slots $\tau, \tau+1, \tau+4$ are non-null publishing, while those at $\tau+2$ and $\tau+3$ are null publishing.

```
Algorithm 3: Dissimilarity Calculation (DC)
Input: $D_t$, current personalized privacy budget list $\boldsymbol{\epsilon}_t$, historical data
       publication $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$
Output: $\boldsymbol{r}_t$
1 $\epsilon_{opt}$= OBS($\boldsymbol{\epsilon}_t$) ;
2 $\tilde{D}_t = SM_s(D_t, \boldsymbol{\epsilon}_t, \epsilon_{opt})$;
3 $\tilde{c}_t = Q(\tilde{D}_t)$;
4 Get the last non-null publishing $\boldsymbol{r}_l$ from $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$;
5 return $dis = \frac{1}{d}\sum_{j=1}^{d} |\tilde{c}_t[j] - r_l[j]| + Lap(1/(d \cdot \epsilon_{opt}))$;
```

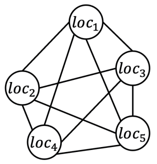
```
Algorithm 4: Personalized Budget Distribution
Input: $D_t$, EPDP privacy requirement ($\boldsymbol{w}$, $\boldsymbol{\epsilon}$), historical data
       publication $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$
Output: $\boldsymbol{r}_t$
1 Get the current window average budget $\bar{\epsilon_i} = \epsilon_i / w_i$ for each $i \in [n]$;
2 $\boldsymbol{\epsilon}_t^{(1)} = (\bar{\epsilon}_1/2, \bar{\epsilon}_2/2, \ldots, \bar{\epsilon}_n/2)$;
3 Get dissimilarity $dis$ by DC($D_t, \boldsymbol{\epsilon}_t^{(1)}, r_1, ..., r_{t-1}$) in Algorithm 3;
4 $\epsilon_{rm,i} = \epsilon_i/2 - \sum_{k=t-w_i+1}^{t-1} \epsilon_{i,k}^{(2)}$;
5 $\boldsymbol{\epsilon}_t^{(2)} = (\epsilon_{rm,1}/2, \epsilon_{rm,2}/2, \ldots, \epsilon_{rm,n}/2)$;
6 $\epsilon_{opt}^{(2)}, err_{opt}^{(2)}$ = OBS($\boldsymbol{\epsilon}_t^{(2)}$);
7 if $dis > \sqrt{err_{opt}^{(2)}}$ then
8     $\tilde{D}_t^{(2)} = SM_s(D_t, \boldsymbol{\epsilon}_t^{(2)}, \epsilon_{opt}^{(2)})$;
9     $\tilde{\boldsymbol{c}}_t^{(2)} = Q(\tilde{D}_t^{(2)})$;
10    return $\boldsymbol{r}_t = SM_d(\tilde{\boldsymbol{c}}_t^{(2)}, \epsilon_{opt}^{(2)})$;
11 else
12    return $\boldsymbol{r}_t = \boldsymbol{r}_{t-1}$;
```

Given a privacy budget ϵ with a window size w, we can calculate the average privacy budget per time slot as $\bar{\epsilon} = \epsilon/w$. This $\bar{\epsilon}$, which we calla budget share, represents the smallest indivisible unit of the privacy budget. Our goal is to maintain the total privacy budget within any w window size below ϵ while keeping it sufficiently large. When publishing new obfuscated data costs x budget shares ($x > 1$), the following $x-1$ time slots will use approximated values from their last reporting values. We refer to these $x - 1$ time slots as nullified time slots. For example, in Figure 3, with a privacy budget ϵ of 4 and a window size of 4, the budget share $\bar{\epsilon}$ equals $\epsilon/w = 1$. When time slot $\tau + 1$ uses 3 shares, the time slots $\tau + 2$ and $\tau + 3$ become nullified time slots.

Personalized Budget Distribution (PBD). As shown in Algorithm 4, PBD inputs the current user data D_i, historical data publication, and all users' privacy budget and window size requirements. The privacy budget ϵ_i of each user u_i is divided into two parts: 1) calculate the dissimilarity between the current data distribution and the last published obfuscated data distribution (denoted as Part$_{DC}$) (Lines 2-3); 2) calculate the new obfuscated publication at the current time slot (denoted as Part$_{NOP}$) (Lines 4-6 and Lines 8-10).

In Part$_{DC}$, we allocate half of the average privacy budget per time slot for dissimilarity calculation (i.e., $\frac{\epsilon_i}{2w_i}$ for u_i). The process then calls the Dissimilarity Calculation (Algorithm 3) to determine the dissimilarity. Within Algorithm 3, the OBS algorithm selects the optimal budget threshold ϵ_{opt}. Finally, it uses the SM [20] to compute the dissimilarity dis (lines 2-5).

	u_1	u_2	u_3	...
$\boldsymbol{\epsilon}$	ϵ_1	ϵ_2	ϵ_3	...
$\boldsymbol{w}$	4	2	3	...



	1	2	3	4	5
u_1	loc_2	loc_1	loc_1	loc_3	loc_2
u_2	loc_1	loc_1	loc_3	loc_3	loc_4
u_3	loc_5	loc_4	loc_4	loc_2	loc_4
...	...	...	...	...	...

Figure 4: An Information example for PBD.

In Part$_{NOP}$, we first calculate the remaining privacy budget $\epsilon_{rm,i}$ for each u_i. We then set the publication privacy budget for each u_i to half of $\epsilon_{rm,i}$. Similar to dissimilarity calculation, we use the OBS algorithm to determine the optimal privacy budget $\epsilon_{opt}^{(2)}$ and its corresponding error $err_{opt}^{(2)}$. At this point, we have obtained two measurements: the dissimilarity dis and the square root of error $\sqrt{err_{opt}^{(2)}}$. We compare these two measurements to determine whether to publish a new obfuscated statistic result or approximate the current result with the last publication. If the dis is greater than $\sqrt{err_{opt}^{(2)}}$, it indicates that the difference between the current data and the last published data exceeds the error of noise, then we republish a new obfuscated statistic result. Otherwise, we take the last published result instead.

We illustrate the process of Personalized Budget Distribution with an example as follows:

Example 3. *Suppose there are n users distributed across* 5 *locations, forming a complete graph. Figure 4 illustrates the privacy budget requirements, window size requirements and locations for the first three users across time slots* 1 *to* 5*. Figure 5 demonstrates the estimation process of PBD. The total privacy budget for each user u_i is evenly split into two parts, each containing $\epsilon_i/2$. The first part is allocated for dissimilarity calculation, while the second is for publication noise calculation. For instance, ϵ_1 is divided into $\boldsymbol{\epsilon}_1^{(1)}(u_1) = \epsilon_1/2$ and $\boldsymbol{\epsilon}_1^{(2)}(u_1) = \epsilon_1/2$. We compute the privacy budget usage $\epsilon_{i,t}^{(1)}$ for dissimilarity and $\epsilon_{i,t}^{(2)}$ for noise statistic publication for each user at each time slot. These values are recorded in an $n \times 2$ matrix at each time slot in Figure 5. Using u_1 as an example, $\epsilon_{1,t}^{(1)} = \boldsymbol{\epsilon}_1^{(1)}(u_1)/w_1 = \epsilon_1/8$. At time slot* 1*, $\epsilon_{1,1}^{(2)} = \boldsymbol{\epsilon}_1^{(2)}(u_1)/2 = \epsilon_1/4$. The algorithm calculates the dissimilarity dis at time slot* 1 *using all $\epsilon_{i,1}^{(1)}$, and the error $err_{opt}^{(2)}$ using all $\epsilon_{1,t}^{(2)}$. Assume dis $> \sqrt{err_{opt}^{(2)}}$, then a new obfuscated statistic $\boldsymbol{r}_1$ is published at time slot* 1*. At time slot* 2*, assume dis $\leq \sqrt{err_{opt}^{(2)}}$, then $\epsilon_{i,2}^{(2)}$ is not used to publish a new obfuscated statistic result, and its usage is set to zeros for all users. At time slot* 3*, $\epsilon_{1,3}^{(2)} = (\epsilon_1/2 - \epsilon_{1,1}^{(2)})/2 = \epsilon_1/8$. The vector below each matrix in Figure 5 represents the total privacy budget used at the current time slot for each user. For example, at time slot* 1*, the total privacy budget usage for u_1 is $\epsilon_{1,1}^{(1)} + \epsilon_{1,1}^{(2)} = 3\epsilon_1/8$.*

Peronalized Budget Absorption (PBA). Algorithm 5 outlines the process of PBA. The dissimilarity calculation (Part$_{DC}$) in PBA is

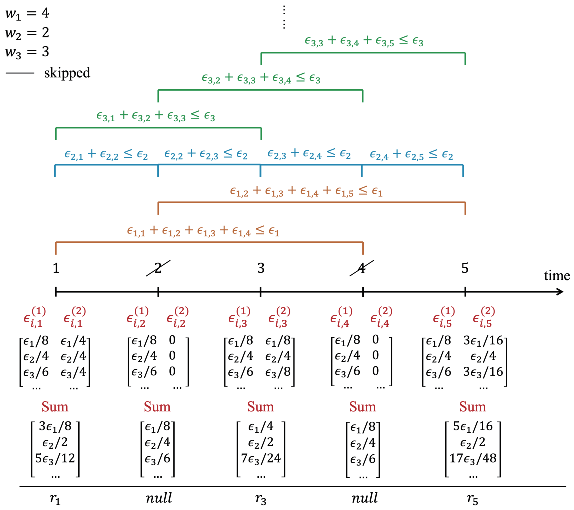


Figure 5: A process example for PBD.

identical to that of PBD. However, PBA and PBD differ significantly in their strategies on allocating the publication privacy budget.

For Part$_{NOP}$ in PBA, we assume an average privacy budget of $\frac{\epsilon_i}{2w_i}$ (one share) for each u_i at each time slot t. A publication at time slot t can use more than one share by borrowing from its successor time slots. The variable $t_{i,N}$ in Line 4 represents the number of successor time slots occupied by the last publication. We calculate the maximal $\tilde{t}_N$ of all $t_{i,N}$ and determine whether the current time has been occupied ($t-l \leq \tilde{t}_N$). If so, we approximate the publication using the last published result. Otherwise, we calculate the remaining budget shares from the precursor time slots (i.e., $t_{A,i}$ in Line 9) and set the current publication budget as the total absorbed shares (Line 10). The subsequent steps follow the same process as outlined in Algorithm 4.

Example 4. *We continue use the demonstration case shown in Figure 4. Figure 6 illustrates the estimation process of PBA. The dissimilarity calculation process in PBA is identical to that in Example 3. For Part$_{NOP}$, at time slot 1, with no budget to absorb, all users utilize one share (i.e., $\epsilon_i/(2w_i)$) to publish a new obfuscated statistic result. Assume time slot 2 is skipped (i.e., dis $\leq \sqrt{err^{(2)}_{opt}}$). At time slot 3, $t_{1,N} = 1$, $t_{2,N} = 0$, and $t_{3,N} = 1.5$. Assuming the nullified bound $\tilde{t}_N$ is 1.8. Since $t - l = 3 - 1 = 2 > \tilde{t}_N$, a new obfuscated statistic result is reported. The publication budget set is calculated as $\boldsymbol{\epsilon}^{(2)}_3 = (\epsilon_1/4, \epsilon_2/2, \epsilon_3/3, \ldots)$. At time slot 4, $t_{1,N} = 1$, $t_{2,N} = 1$ and $t_{3,N} = 1$ (Actually, all $t_{i,N} = 1$). As $t-l = 4-3 = 1 \leq \tilde{t}_N$, no output is produced. At time slot 5, all $t_{i,N}$ remain 1, and $t-l = 5-3 = 2 > \tilde{t}_N$. The absorbed time slots $t_{A,i}$ all equal 1. The resulting publication budget set is $\boldsymbol{\epsilon}^{(2)}_5 = (\epsilon_1/8, \epsilon_2/4, \epsilon_3/6, \ldots)$.*

4.5 Analyses

W1e analyze the time cost and privacy aspects of our PBD and PBA.

Time Cost Analysis. Let m be the number of distinct privacy requirements (w_i, ϵ_i), where $m \leq n$. The time complexity of OBS is $O(m)$ for both PBD and PBA. The Sample Mechanism and Query operations each have a time complexity of $O(n)$. Thus, the time complexities of PBD and PBA both are $O(n)$.

Algorithm 5: Personalized Budget Absorption

```
Input: $D_t$, EPDP privacy requirement $(\boldsymbol{w}, \boldsymbol{\epsilon})$, historical data
       publication $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$
Output: $\boldsymbol{r}_t$
1  Get the current window average budget $\bar{\epsilon}_i = \epsilon_i/w_i$ for each $i \in [n]$;
2  $\boldsymbol{\epsilon}^{(1)}_t = (\bar{\epsilon}_1/2, \bar{\epsilon}_2/2, \ldots, \bar{\epsilon}_n/2)$;
3  Get dissimilarity $dis$ by DC($D_t, \boldsymbol{\epsilon}^{(1)}_t, r_1, ..., r_{t-1}$) in Algorithm 3;
4  Set nullified time slots $t_{i,N} = \frac{\epsilon^{(2)}_{i,l}}{\epsilon_i/(2w_i)} - 1$ for $i \in [n]$ where $l$ is the
     last non-null publishing time slot;
5  Set nullified time slot bound $\tilde{t}_N = \max_{i \in [n]} t_{i,N}$;
6  if $t - l \leq \tilde{t}_N$ then
7      return $\boldsymbol{r}_t = \boldsymbol{r}_{t-1}$;
8  else
9      Set absorbed time slots $t_{A,i} = \max(t - l - t_{i,N}, 0)$ for $i \in [n]$;
10     Set publication budget $\epsilon^{(2)}_{i,t} = \frac{\epsilon_i}{2w_i} \cdot \min(t_{A,i}, w_i)$ for $i \in [n]$;
11     $\boldsymbol{\epsilon}^{(2)}_t = \left(\epsilon^{(2)}_{1,t}, \epsilon^{(2)}_{2,t}, \ldots, \epsilon^{(2)}_{n,t}\right)$;
12     Get remaining budget $\epsilon_{rm,i} = \epsilon_i/2 - \sum_{k=t-w_i+1}^{t-1} \epsilon^{(2)}_{i,k}$;
13     $\boldsymbol{\epsilon}^{(2)}_t = (\epsilon_{rm,1}/2, \epsilon_{rm,2}/2, \ldots, \epsilon_{rm,n}/2)$;
14     $\epsilon^{(2)}_{opt}, err^{(2)}_{opt}$ = OBS($\boldsymbol{\epsilon}^{(2)}_t$);
15     if $dis > \sqrt{err^{(2)}_{opt}}$ then
16         $\tilde{D}^{(2)}_t = SM_s(D_t, \boldsymbol{\epsilon}^{(2)}_t, \epsilon^{(2)}_{opt})$;
17         $\tilde{\boldsymbol{c}}^{(2)}_t = Q(\tilde{D}^{(2)}_t)$;
18         return $\boldsymbol{r}_t = SM_d(\tilde{\boldsymbol{c}}^{(2)}_t, \epsilon^{(2)}_{opt})$;
19     else
20         return $\boldsymbol{r}_t = \boldsymbol{r}_{t-1}$;
```

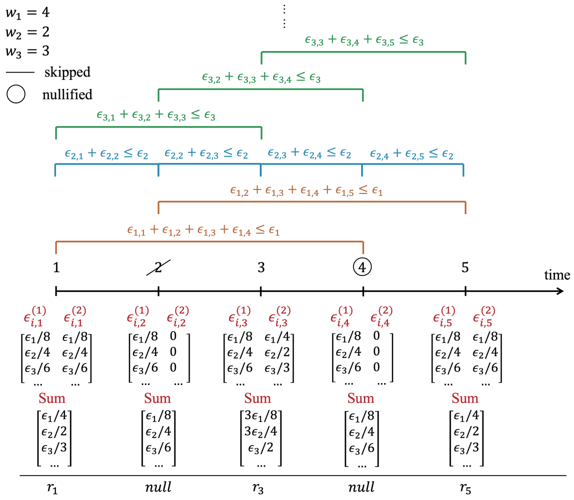


Figure 6: A process example for PBA.

Privacy Analysis. The privacy analysis for PBD and PBA:

Theorem 4.1. *PBD and PBA satisfy $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP.*

Proof. (1) PBD satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP.

In the process of Part$_{DC}$, for each user u_i, the dissimilarity budget at each time slot is $\epsilon_i/(2w_i)$. Then for each time slot t, we have

$$\sum_{k=\max(t-w_i+1,1)}^{t} \epsilon^{(1)}_{i,k} = \epsilon_i/2. \tag{5}$$

In Part$_{NOP}$, for each user u_i at time slot t, only half of the publication budget is used when publication occurs: $\epsilon_{i,t}^{(2)} = (\epsilon_i/2 - \sum_{k=\max(t-w_i+1,1)}^{t-1} \epsilon_{i,k}^{(2)})/2$. For any time slot $t \in [1, w_i]$, the summation publication budgets used for u_i is at most $\sum_{k=1}^{w_i} \epsilon_i/(2 \cdot 2^k) \leq (\epsilon_i/2) \cdot (1 - \frac{1}{2^{w_i}}) \leq \epsilon_i/2$. Suppose $\sum\limits_{k=\max(t-w_i+1,1)}^{t} \epsilon_{i,k}^{(2)} \leq \epsilon_i/2$ for $t = w_i + s$ (i.e., $\sum\limits_{k=\max(s+1,1)}^{w_i+s} \epsilon_{i,k}^{(2)} \leq \epsilon_i/2$). Then for $t = w_i + s + 1$, we have:

$$\sum_{k=\max(s+2,1)}^{w_i+s+1} \epsilon_{i,k}^{(2)} = \sum_{k=\max(s+2,1)}^{w_i+s} \epsilon_{i,k}^{(2)} + \epsilon_{i,w_i+s+1}^{(2)}. \tag{6}$$

Since $\epsilon_{i,w_i+s+1}^{(2)}$ is at most half of the remaining publication budget at time slot $w_i + s$:

$$\epsilon_{i,w_i+s+1}^{(2)} \leq (\epsilon_i/2 - \sum_{k=\max(s+2,1)}^{w_i+s} \epsilon_{i,k}^{(2)})/2. \tag{7}$$

According to Equations (6) and (7), we have:

$$\begin{aligned}
\sum_{k=\max(s+2,1)}^{w_i+s+1} \epsilon_{i,k}^{(2)} &\leq \sum_{\substack{k=\max(\\ s+2,1)}}^{w_i+s} \epsilon_{i,k}^{(2)} + (\epsilon_i/2 - \sum_{\substack{k=\max(\\ s+2,1)}}^{w_i+s} \epsilon_{i,k}^{(2)})/2 \\
&= \epsilon_i/4 + (\sum_{k=\max(s+2,1)}^{w_i+s} \epsilon_{i,k}^{(2)})/2 \\
&\leq \epsilon_i/4 + \epsilon_i/4 \\
&= \epsilon_i/2.
\end{aligned} \tag{8}$$

Therefore, for any $t \geq 1$, we have:

$$\sum_{k=\max(t-w_i+1,1)}^{t} \epsilon_{i,k}^{(2)} \leq \epsilon_i/2. \tag{9}$$

According to the Composition Theorems [14], we have:

$$\begin{aligned}
\sum_{k=\max(t-w_i+1,1)}^{t} \epsilon_{i,k} &= \sum_{\substack{k=\max(t\\ -w_i+1,1)}}^{t} \epsilon_{i,k}^{(1)} + \sum_{\substack{k=\max(t\\ -w_i+1,1)}}^{t} \epsilon_{i,k}^{(2)} \\
&\leq \epsilon_i.
\end{aligned} \tag{10}$$

For any user u_i and any two w_i-neighboring stream prefixes S_t and S'_t (i.e., $S_t \sim_{w_i} S'_t$), let t_s be the earliest time slot where $S_t[t_s] \neq S'_t[t_s]$ and t_e be the latest time slot where $S_t[t_e] \neq S'_t[t_e]$. Then we have $t_e - t_s + 1 \leq w_i$. Denoting the output of our PBD as $PBD(S_t[t]) = o_t \in \mathcal{O}$, for any $O \subseteq \mathcal{O}$, we have:

$$\begin{aligned}
\frac{\Pr[PBD(S_t)] \in O}{\Pr[PBD(S'_t)] \in O} &\leq \Pi_{k=t_s}^{t_e} \frac{\Pr[PBD(S_t[k]) = o_k]}{\Pr[PBD(S'_t[k]) = o_k]} \\
&\leq e^{\sum_{k=t_s}^{t_e} \epsilon_{i,k}} \\
&\leq e^{\sum_{k=\max(t_e-w_i+1,1)}^{t_e} \epsilon_{i,k}} \leq e^{\epsilon_i}.
\end{aligned} \tag{11}$$

Therefore, PBD satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP where $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ and $\boldsymbol{\epsilon} = ((u_1, \epsilon_1), (u_2, \epsilon_2), \ldots, (u_n, \epsilon_n))$.

(2) PBA satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP.

The Part$_{DC}$ in PBA is identical to that that in PBD. Consequently, for each time slot t, we have:

$$\sum_{k=\max(t-w_i+1,1)}^{t} \epsilon_{i,k}^{(1)} = \epsilon_i/2. \tag{12}$$

In Part$_{NOP}$, for any user u_i and any window of size w_i, there are s_i publication time slots in the window. We denote these publication time slots as $(k_1, k_2, \ldots, k_{s_i})$. For any publication time slot k_j ($j \in [s_i]$), the quantity of its absorbing unused budgets is denoted as α_{i,k_j}. Figure 7 illustrates an example where $s_i = 3$ and $w_i = 9$.

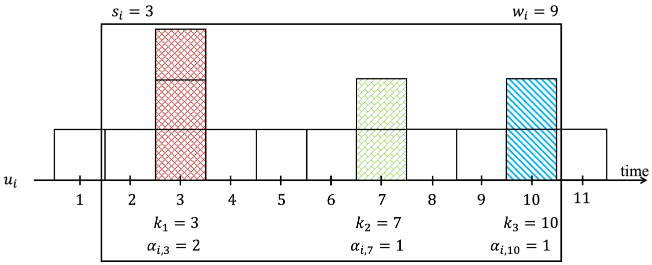


Figure 7: An example for parameters in PBA.

Based on Algorithm 5, we have:

$$w_i \geq \sum_{j=1}^{s_i} (1 + 2\alpha_{i,k_j}) - \alpha_{i,k_1} - \alpha_{i,k_{s_i}}. \tag{13}$$

Then, for the total publication budgets used in any window, we have

$$\begin{aligned}
\sum_{k=\max(t-w_i+1,1)}^{t} \epsilon_{i,k}^{(2)} &\leq \frac{\epsilon_i}{2w_i} \cdot \sum_{j=1}^{s_i} (1 + \alpha_{i,k_j}) \\
&\leq \frac{\epsilon_i \cdot \sum_{j=1}^{s_i} (1 + \alpha_{i,k_j})}{2\sum_{j=1}^{s_i} (1 + 2\alpha_{i,k_j}) - 2\alpha_{i,k_1} - 2\alpha_{i,k_{s_i}}} \\
&= \frac{\epsilon_i \cdot \sum_{j=1}^{s_i} (1 + \alpha_{i,k_j})}{2\sum_{j=1}^{s_i} (1 + \alpha_{i,k_j}) + 2\sum_{j=2}^{s_i-1} \alpha_{i,k_j}} \\
&\leq \epsilon_i/2.
\end{aligned} \tag{14}$$

Based on Equations (12) and (14), and applying the Composition Theorems [14], we obtain:

$$\begin{aligned}
\sum_{k=\max(t-w_i+1,1)}^{t} \epsilon_{i,k} &= \sum_{\substack{k=\max(t\\ -w_i+1,1)}}^{t} \epsilon_{i,k}^{(1)} + \sum_{\substack{k=\max(t\\ -w_i+1,1)}}^{t} \epsilon_{i,k}^{(2)} \\
&\leq \epsilon_i.
\end{aligned} \tag{15}$$

The subsequent proof process follows the same steps as in PBD. Ultimately, we demonstrate that PBA also satisfies $(\boldsymbol{w}, \boldsymbol{\epsilon})$-EPDP. □

Utility Analysis. For each user u_i in PBD and PBA, we define w_L as the smallest window size among all users. For each u_i, given (w_i, ϵ_i), let $\epsilon_L = \min_{i \in [n]} \frac{\epsilon_i}{w_i}$ and $\epsilon_R = \max_{i \in [n]} \frac{\epsilon_i}{w_i}$ be the minimum and maximum values of $\frac{\epsilon_i}{w_i}$, respectively. Let n_A be the number of times ϵ_R appears among all users. We assume that at most $\tilde{s} \leq w_L$ publications occur at time slots $q_1, q_2, \ldots, q_{\tilde{s}}$ in the window of size w_L. We also assume there is no budget absorption from past time slots outside the window. Furthermore, for each user, each publication approximates the same number of skipped or nullified publications.

We first present a crucial lemma.

Lemma 4.1. *Given m distinct privacy budget-quantity pairs $P = \{(\epsilon_j, n_j) | j \in [m], \sum_{j \in [m]} n_j = n\}$ where pair (ϵ_j, n_j) indicates that ϵ_j appears n_j times in the user privacy requirement, and a query with sensitivity I, the error upper bound $\widetilde{err}_O(P)$ of the SM process with privacy budget chosen from OBS is:*

$$\min\left(\frac{2I^2}{\min_j \epsilon_j^2}, (n - n_M)(n - n_M + \frac{1}{4}) + \frac{2I^2}{\max_j \epsilon_j^2}\right),$$

where $n_M = n_k$ with $k = \arg\max_{j \in [m]} \epsilon_j$.

Proof. Let M_L be the SM with privacy budget chosen as $\min_j \epsilon_j$. According to the SM process, all budget types will be selected. In this case, the sampling error err_s is 0 and the noise error err_{dp} is $2 \cdot (\frac{I}{\min_j \epsilon_j})^2 = \frac{2I^2}{\min_j \epsilon_j^2}$. Thus, the total error of M_L is $err_{M_L} = \frac{2I^2}{\min_j \epsilon_j^2}$. Let M_R be the SM with privacy budget chosen as $\max_j \epsilon_j$. In this case, $(m-1)$ types of privacy budget are chosen with probability $p_k = \frac{e^{\epsilon_k}-1}{e^{\max_j \epsilon_j}-1}$ less than 1 ($k \in [m]$). For the sampling error, we have:

$$\begin{aligned}
err_s &= \sum_{\epsilon_k < \max_j \epsilon_j} n_k p_k (1-p_k) + \left(\sum_{\epsilon_k < \max_j \epsilon_j} n_k (1-p_k) \right)^2 \\
&< \sum_{\epsilon_k < \max_j \epsilon_j} n_k \left(\frac{p_k + 1 - p_k}{2} \right)^2 + \left(\sum_{\epsilon_k < \max_j \epsilon_j} n_k \right)^2 \\
&= \frac{1}{4}(n - n_M) + (n - n_M)^2 \\
&= (n - n_M)(n - n_M + \frac{1}{4}).
\end{aligned}$$

The noise error err_{dp} in this case is $2 \cdot (\frac{I}{\max_j \epsilon_j})^2 = \frac{2I^2}{\max_j \epsilon_j^2}$. Thus, the total error of M_R is $err_{M_R} = (n - n_M)(n - n_M + \frac{1}{4}) + \frac{2I^2}{\max_j \epsilon_j^2}$. According to the OBS process, we have $\widetilde{err}_O(P) \le err_{M_L}$ and $\widetilde{err}_O(P) \le err_{M_R}$. Therefore,

$$\begin{aligned}
\widetilde{err}_O(P) &\le \min(err_{M_L}, err_{M_R}) \\
&= \min\left(\frac{2I^2}{\min_j \epsilon_j^2}, (n - n_M)(n - n_M + \frac{1}{4}) + \frac{2I^2}{\max_j \epsilon_j^2} \right)
\end{aligned}$$

□

For PBD we present Theorem 4.2 as follows.

Theorem 4.2. *The average error per time slot in PBD is at most* $\min\left(\frac{8}{d^2\epsilon_L}, Z + \frac{8}{d^2\epsilon_R}\right) + \min\left(\frac{32\cdot(4^{\tilde{s}}-1)}{3\tilde{s}\epsilon_L}, Z + \frac{32\cdot(4^{\tilde{s}}-1)}{3\tilde{s}\epsilon_R}\right)$ *where* $Z = (n - n_A)(n - n_A + \frac{1}{4})$*, if at most* $\tilde{s}$ *publications occur in any window with size* w_L*.*

Proof. Given a privacy budget-quantity pair set P, let $EOPT(P)$ be the optimal privacy budget chosen from OBS. Given a positive number β, we define $\beta \cdot P = \{(\beta \cdot \epsilon_j, n_j) | (\epsilon_j, n_j) \in P\}$. For each user u_i with privacy requirement pair (w_i, ϵ_i), we calculate their average budget per window as $\frac{\epsilon_i}{w_i}$. We denote the set of all average budgets as $\overline{\epsilon} = \{\frac{\epsilon_i}{w_i} | i \in [n]\}$. We then construct the privacy budget-quantity pair set of each type of average budget as $P_A = \{(\epsilon_j, n_j) | \epsilon_j \in \overline{\epsilon}\}$. Let $Z = (n - n_A)(n - n_A + \frac{1}{4})$ be the sampling error upper bound, where n_A is the quantity of $\max_{i \in [n]} \frac{\epsilon_i}{w_i}$ in $\overline{\epsilon}$.

When Part$_{DC}$ is not private, the error stems from Part$_{NOP}$. In Part$_{NOP}$, errors arise from both publications and approximations. According to the Part$_{NOP}$, an approximation error does not exceed the publication error at the most recent publication time slot. For the average error $\overline{err}_{NOP}$ of all time slots within the window of size w_L, based on the PBD process, we have:

$$\begin{aligned}
\overline{err}_{NOP} &= \frac{1}{w_L} \sum_{k \in [\tilde{s}]} \frac{w_L}{\tilde{s}} \cdot \widetilde{err}_O\left(\frac{1}{2^{k+1}} P_A \right) \\
&< \frac{1}{\tilde{s}} \sum_{k \in [\tilde{s}]} \min\left(\frac{2}{(\frac{\epsilon_L}{2^{k+1}})^2}, Z + \frac{2}{(\frac{\epsilon_R}{2^{k+1}})^2} \right) \\
&< \frac{1}{\tilde{s}} \min\left(\sum_{k \in [\tilde{s}]} \frac{8 \cdot 4^k}{\epsilon_L^2}, \tilde{s} \cdot Z + \sum_{k \in [\tilde{s}]} \frac{8 \cdot 4^k}{\epsilon_R^2} \right) \\
&= \min\left(\frac{32 \cdot (4^{\tilde{s}} - 1)}{3\tilde{s}\epsilon_L^2}, Z + \frac{32 \cdot (4^{\tilde{s}} - 1)}{3\tilde{s}\epsilon_R^2} \right).
\end{aligned} \tag{16}$$

When Part$_{DC}$ is private, the error from Part$_{DC}$ can lead to two scenarios: (1) falsely skipping a publication or (2) falsely performs a publication. Both cases are bounded by the error in Part$_{DC}$. In Part$_{DC}$, we execute the SM with OBS. The sensitivity of dis is $1/d$. For the average error $\overline{err}_{DC}$ of each time slot in window size w_L, according to Lemma 4.1, we have:

$$\begin{aligned}
\overline{err}_{DC} &< \min\left(\frac{2}{d^2 \min_{i \in [n]} (\frac{\epsilon_i}{2w_i})^2}, Z + \frac{2}{d^2 \max_{i \in [n]} (\frac{\epsilon_i}{2w_i})^2} \right) \\
&= \min\left(\frac{8}{d^2\epsilon_L^2}, Z + \frac{8}{d^2\epsilon_R^2} \right).
\end{aligned} \tag{17}$$

Based on Equation (17) and (16), we can get the average error upper bound as $\overline{err}_{DC} + \overline{err}_{NOP}$.

□

PBD achieves low error when the number of publications $\tilde{s}$ per window is small. However, the error increases exponentially with $\tilde{s}$. Additionally, the error in Part$_{DC}$ (the first part of the error upper bound in PBD) rises as w_L increases, however, it diminishes as d increases. This is because a large d reduces sensitivity leading to smaller noise error.

For PBA, assume α skipped publications occur before a publication. Let $\epsilon_{\tilde{L}}$ and $\epsilon_{\tilde{R}}$ be the minimum and maximum publication privacy budget among all users at time slots $t = w_L$ and $t = (\alpha + 1)$, respectively. According to the PBA process, there will be α nullified publications after the publication. These nullified publications are filled by the last time slot's publication without comparison. Consequently, the nullified publication error depends on the data distribution at nullified time slots. We denote the average error of each nullified publication in PBA as $\overline{err}_{nlf}$. For PBA, we have Theorem 4.3 as follows.

Theorem 4.3. *The average error per time slot in PBA is at most* $\min(\frac{8}{d^2\epsilon_L}, Z + \frac{8}{d^2\epsilon_R}) + \frac{1}{2\alpha+1}(\widetilde{err}_{NOP}^{(s,p)} + \alpha \cdot \overline{err}_{nlf})$ *where* $\widetilde{err}_{NOP}^{(s,p)}$ *is* $\min(\frac{2}{\epsilon_L^2}H_{\alpha+1}^2, (\alpha+1)Z + \frac{2}{\epsilon_R^2}H_{\alpha+1}^2)$ *when* $\alpha \le w_L$ *and* $\min(\frac{2}{\epsilon_L^2}H_{w_L}^2, w_L Z + \frac{2}{\epsilon_R^2}H_{w_L}^2) + (\alpha - w_L + 1)\min(\frac{2}{\epsilon_{\tilde{L}}^2}, Z + \frac{2}{\epsilon_{\tilde{R}}^2})$ *when* $\alpha > w_L$ *and* $Z = (n - n_A)(n - n_A + \frac{1}{4})$ *and* H_x^2 *is the x-th square harmonic number, if there are* α *skipped publications occur in average before each publication.*

Proof. Similar to PBD, we first analyze the error of Part$_{NOP}$ in PBA by assuming Part$_{DC}$ is not private. We then add the error of Part$_{DC}$, which is identical to that in PBD, to obtain the final total error. When Part$_{DC}$ is not private, the error stems from Part$_{NOP}$. In Part$_{NOP}$, each publication corresponds to α skipped publications preceding it and α nullified publications succeeding it.

For each user u_i's skipped publication, the publication privacy budget lower bound doubles with each time slot increase until it reaches $\epsilon_i/2$ or a publication occurs. For example, in Figure 8, where $\alpha = 5$, the publication time slot is t_6. At time slot t_1, each u_i's publication budget lower bound is $\epsilon_i/(2w_i)$. Take u_1 as an example: it reaches $\epsilon_1/2$ at time slot t_4. The publication lower bound for u_1 remains at $\epsilon_1/2$ until time slot t_6. Let the publication budget lower bound set for all users at skipped time slots (spanning α time slot) be $\hat{\boldsymbol{\epsilon}} = \{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_\alpha\}$. Then, the error upper bound of each skipped publication is the error of publishing new data using $\boldsymbol{\epsilon}_k$ ($k \in [\alpha]$). For example in Figure 8, the error upper bound at t_3 is the error of publication a new obfuscated statistic result using $\{\frac{3\epsilon_1}{2}, \frac{\epsilon_2}{2}, \frac{3\epsilon_3}{16}, \frac{\epsilon_4}{4}\}$.

Let $Z = (n - n_A)(n - n_A + \frac{1}{4})$ be the sampling error upper bound, where n_A is the number of users with maximum value of $\frac{\epsilon_i}{w_i}$. We now consider two cases: $\alpha \le w_L$ and $\alpha > w_L$.

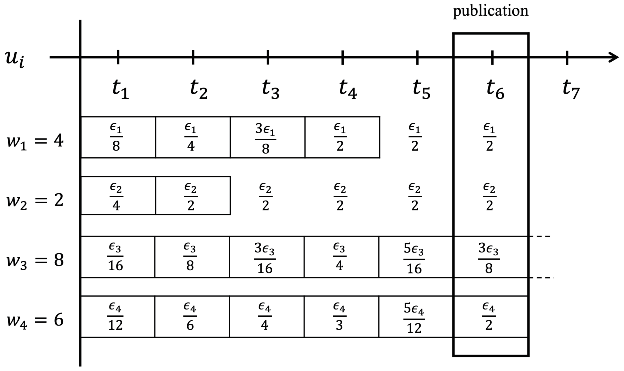


Figure 8: An example of the publication budget lower bound in PBA.

(1) case 1: $\alpha \leq w_L$.

In this case, the publication budget lower bound doubles with each time slot increase. Let $err^{(sk)}_{NOP}(\alpha)$ and $err^{(pb)}_{NOP}$ be the total error upper bounds of the α skipped publications and the publication in Part$_{NOP}$, respectively. Let $err^{(s,p)}_{NOP}$ be the error of all skipped publications and the publication in Part$_{NOP}$. According to Lemma 4.1, we have

$$\begin{aligned} err^{(sk)}_{NOP}(\alpha) &< \sum_{k\in[\alpha]} \min\left(\frac{2}{(k\epsilon_L)^2}, Z + \frac{2}{(k\epsilon_R)^2}\right) \\ &\leq \min\left(\frac{2}{\epsilon_L^2}H_\alpha^2, \alpha Z + \frac{2}{\epsilon_R^2}H_\alpha^2\right) \end{aligned} \tag{18}$$

and

$$\begin{aligned} err^{(s,p)}_{NOP} &< err^{(sk)}_{NOP}(\alpha) + err^{(pb)}_{NOP} \\ &= err^{(sk)}_{NOP}(\alpha+1) \\ &= \min\left(\frac{2}{\epsilon_L^2}H_{\alpha+1}^2, (\alpha+1)Z + \frac{2}{\epsilon_R^2}H_{\alpha+1}^2\right). \end{aligned} \tag{19}$$

Thus, we derive the average error upper bound $\overline{err}_{NOP}$ of each time slot in Part$_{NOP}$ as

$$\overline{err}_{NOP} < \frac{1}{2\alpha+1}(\widetilde{err}^{(s,p)}_{NOP} + \alpha \cdot \overline{err}_{nlf}) \tag{20}$$

where $\widetilde{err}^{(s,p)}_{NOP}$ is the final value in Equation (19).

(2) case 2: $\alpha > w_L$.

In this case, we have

$$\begin{aligned} err^{(s,p)}_{NOP} <& err^{(sk)}_{NOP}(w_L) + \sum_{k=w_L+1}^{\alpha+1} \min\left(\frac{2}{\epsilon_{\tilde{L}}^2}, Z + \frac{2}{\epsilon_{\tilde{R}}^2}\right) \\ =& err^{(sk)}_{NOP}(w_L) + (\alpha - w_L + 1)\min\left(\frac{2}{\epsilon_{\tilde{L}}^2}, Z + \frac{2}{\epsilon_{\tilde{R}}^2}\right) \\ <& \min\left(\frac{2}{\epsilon_L^2}H_{w_L}^2, w_L Z + \frac{2}{\epsilon_R^2}H_{w_L}^2\right) \\ &+ (\alpha - w_L + 1)\min\left(\frac{2}{\epsilon_{\tilde{L}}^2}, Z + \frac{2}{\epsilon_{\tilde{R}}^2}\right). \end{aligned} \tag{21}$$

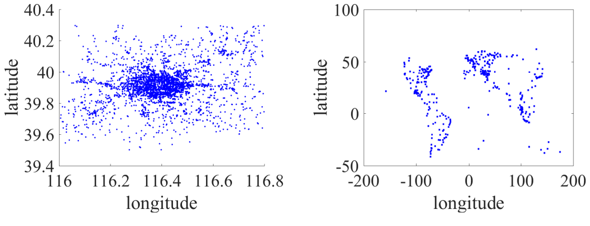
Therefore, we obtain the average error upper bound $\overline{err}_{NOP}$ for each time slot in Part$_{NOP}$ as

$$\overline{err}_{NOP} < \frac{1}{2\alpha+1}(\widetilde{err}^{(s,p)}_{NOP} + \alpha \cdot \overline{err}_{nlf}) \tag{22}$$

where $\widetilde{err}^{(s,p)}_{NOP}$ is the value derived in Equation (21).

When Part$_{DC}$ is private, its error is identical to that in PBD:

$$\overline{err}_{DC} < \min\left(\frac{8}{d^2\epsilon_L^2}, Z + \frac{8}{d^2\epsilon_R^2}\right). \tag{23}$$



(a) Taxi

(b) Foursquare

Figure 9: Real datasets.

Based on Equation (23), (20) and (22), we can derive the average error upper bound for each time slot in PBA as:

$$\min\left(\frac{8}{d^2\epsilon_L^2}, Z + \frac{8}{d^2\epsilon_R^2}\right) + \frac{1}{2\alpha+1}(\widetilde{err}^{(s,p)}_{NOP} + \alpha \cdot \overline{err}_{nlf}), \tag{24}$$

where $\widetilde{err}^{(s,p)}_{NOP}$ is the final result from Equation (19) when $\alpha \leq w_L$, and from Equation (21) when $\alpha > w_L$. □

5 EXPERIMENTS

5.1 Datasets

We evaluate our solutions on both real and synthetic datasets.

Real datasets. We use two real-world datasets, *Taxi* [37, 38] and *Foursquare* [35, 36], to evaluate the performance of our algorithms.

Taxi. It contains real-time trajectories of 10,357 taxis' in Beijing from February 2 to February 8, 2008. Each taxi has up to 154,699 records, where each record comprises *taxi id*, *data time*, *longitude* and *latitude*. For the spatial dimension, we first remove all duplicate records, then extract records with longitude between 116 and 116.8 and latitude between 39.5 and [40.3], resulting in 14,859,377 records. We denote this area ([116, 116.8] × [39.5, 40.3]) as A_E. Figure 9(a) shows 50% of uniformly extracted trajectory points in A_E. We further divide A_E uniformly into a 10 × 10 grids, designating these 100 cells as the location space. For the time dimension, we sample records every minute and get 8,889 records.

Foursquare. It contains 33,278,683 Foursquare check-ins from 266,909 users, during April 2012 to September 2013. Each record consists of user id, venue id (place), and time. We convert the venue id to the country where the venue is located. After removing invalid records, we uniformly extract 5% of users' check-ins as shown in Figure 9(b). We set the publication time interval to 100 minutes, thus divide the chick-ins period into 7,649 time slots.

Synthetic datasets. We generate three binary stream datasets using different sequence models. Let $p_t = f(t)$ be the probability of setting the real value to 1 at time slot t. We set the length of each binary stream as T and the number of users as N. For each stream, we first generate a probability sequence $(p_1, p_2, ..., p_T)$. At each time slot t, each user's real value is set to 1 with probability p_t and 0 otherwise. The probability functions we use are as follows:

- TLNS function. In TLNS, $p_t = p_{t-1} + \mathcal{N}(0, Q)$, where $\mathcal{N}(0, Q)$ is Gaussian noise with standard variance $\sqrt{Q} = 0.0025$. We set $p_0 = 0.05$ as the initial value. If $p_t < 0$, we set $p_t = 0$; If $p_t > 1$, we set $p_t = 1$.
- Sin function. In Sin, $p_t = A\sin(\omega t) + h$, where $A = 0.05$, $\omega = 0.01$ and $h = 0.075$.
- Log function. In Log, $p_t = A/(1 + e^{-bt})$, where $A = 0.25$ and $b = 0.01$.

5.2 Experiment Setup

We divide the total time series into two batches for all datasets, with each batch containing at most half of the total time slots.

We compare our PBD and PBA with two non-personalized methods: Budget Distribution (BD) and Budget Absorption (BA) [22]. We also compare against a simple personalized LDP method, Personalized LDP Budget Uniform (PLBU), which extends LDP Budget Uniform (LBU) [29] by replacing the inner CDP mechanism with an LDP mechanism.

Let ϵ and w be the privacy budget and window size in non-personalized static methods (BD and BA). For non-personalized static methods, we set the ϵ to vary from 0.2 to 1.0 and w to vary from 40 to 200. To make our PBD and PBA comparable with BD and BA, we set the lower bound of each user's privacy budget as ϵ and the upper bound of each user's window size as w in PBD and PBA to match the requirement of privacy level.

Given $\tilde{n}$ different privacy budgets $\tilde{\boldsymbol{\epsilon}} = \{\epsilon_1, ..., \epsilon_{\tilde{n}}\}$, let $N(\epsilon_i)$ be the count of budget value ϵ_i, and $N(\tilde{\boldsymbol{\epsilon}}) = \sum_{i=1}^{\tilde{n}} N(\epsilon_i)$ be the total count of all the budgets. For any $\epsilon_i \in \tilde{\boldsymbol{\epsilon}}$, we define the privacy budget ratio of ϵ_i as $\frac{N(\epsilon_i)}{N(\tilde{\boldsymbol{\epsilon}})}$. Similarly, we define the window size ratio of any w_i in different window sizes $\tilde{\boldsymbol{w}} = \{w_1, ..., w_{\tilde{n}}\}$ as $\frac{N(w_i)}{N(\tilde{\boldsymbol{w}})}$. We set the privacy domain as $\{0.5, 1.0\}$ and the window size domain as $\{10, 20\}$. We alter the ratio o of $\epsilon_i = 0.5$ and $w_i = 10$ from 0.1 to 0.9.

Table 3: Experimental settings.

Parameters	Values
static privacy budget ϵ	0.2, 0.4, **0.6**, 0.8, 1.0
static window size w	40, 80, **120**, 160, 200
personalized privacy budget ϵ_i	$\epsilon, \ldots, 0.8, 1.0$
personalized window size w_i	$40, 80, \ldots, w$
users' quality ratio o	0.1, 0.3, **0.5**, 0.7, 0.9

The parameters are shown in Table 3, where the default values are in bold font. We run the experiments on an Intel(R) Xeon(R) Silver 4210R CPU @ 2.4GHz with 128 RAM in Java. Each experiment is run 10 times, and we report the average result.

5.3 Measures

We evaluate the performance of different mechanisms based on their running time and data utility. We measure data utility as *Average Mean Relative Error* (*AMRE*) and *Average Jensen-Shannon Divergence* ($AJSD$, $\bar{D}_{JS}$). Let T represent the number of time slots and d denote the dimension of data.

AMRE is defined as the average value of Mean Relative Error (*MRE*), which is shown in Equation (25).

$$AMRE = \frac{1}{T}\sum_{\tau=1}^{T} MRE_{\tau} = \frac{1}{T}\sum_{\tau=1}^{T}\frac{1}{d}\|\boldsymbol{r}_{\tau} - \boldsymbol{c}_{\tau}\|_2^2. \tag{25}$$

AJSD is defined as the average value of Jensen-Shannon Divergence (JSD, D_{JS}) [25], which is based on Kullback-Leibler Divergence [24], as shown in Equation (26).

$$\begin{aligned}\bar{D}_{JS}(\boldsymbol{r}\|\boldsymbol{c}) &= \frac{1}{T}\sum_{\tau=1}^{T} D_{JS}(\boldsymbol{r}\|\boldsymbol{c})\\ &= \frac{1}{T}\sum_{\tau=1}^{T}\left(\frac{1}{2}D_{KL}(\boldsymbol{r}\|\boldsymbol{v}) + \frac{1}{2}D_{KL}(\boldsymbol{c}\|\boldsymbol{v})\right) \qquad (26)\\ &= \frac{1}{2T}\sum_{\tau=1}^{T}\sum_{j=1}^{d}\left(\boldsymbol{r}_{\tau}(j)\log\left(\frac{\boldsymbol{r}_{\tau}(j)}{\boldsymbol{v}_{\tau}(j)}\right) + \boldsymbol{c}_{\tau}(j)\log\left(\frac{\boldsymbol{c}_{\tau}(j)}{\boldsymbol{v}_{\tau}(j)}\right)\right),\end{aligned}$$

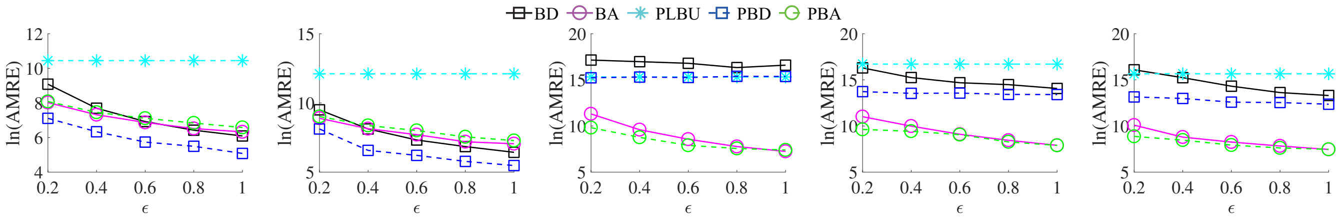
where $\boldsymbol{v}$ represents the average distribution of $\boldsymbol{r}$ and $\boldsymbol{c}$, i.e., $\boldsymbol{v}(j) = \frac{1}{2}(\boldsymbol{r}(j) + \boldsymbol{c}(j))$. For time slot τ, $r_{\tau}(j)$ and $c_{\tau}(j)$ represent the j-th dimensional values in the obfuscated and original data, respectively.

5.4 Overall Utility Analysis

Figure 10 shows the natural logarithm of *AMRE* as the privacy budget ϵ varies. Across all datasets, *AMRE* decreases as ϵ increases, because a larger ϵ results in smaller noise variance, leading to a lower *AMRE*. The decrease in *AMRE* is more pronounced on real datasets compared to synthetic ones. It is because data density function changes rapidly in real datasets, while changing gradually in synthetic datasets. When the density function changes rapidly, the dissimilarity at each time slot becomes large. In this case, PBD publishes more new statistical results than PBA because PBD always reserves part of its privacy budget for the next time slot, even though the budget decreases over time within a window. Thus, PBD leads to higher accuracy than PBA. When the density function changes gradually, the dissimilarity at each time slot remains small. In this case, publishing one highly accurate statistical result at a time slot is more important than publishing multiple new statistical results. Therefore, PBA performs significantly better than PBD. PLBU performs worse than other methods across all datasets except for TLNS, since LDP methods achieve lower accuracy than CDP methods under the same privacy budget. In real datasets, our PBD consistently outperforms other methods. The *AMRE* of PBD is on average 70.8% (17.5% in terms of $\ln(AMRE)$) lower than that of BD on Taxi dataset and 69.6% (15.9% in terms of $\ln(AMRE)$) lower on Foursquare dataset. Our PBA performs slightly worse than BA, since our PBA is more sensitive to noise in high-dimensional data. For synthetic datasets, our PBA consistently outperforms other methods. Compared to BA, the *AMRE* of PBA is lower on average of 36.9% (6.0% in terms of $\ln(AMRE)$) on TLNS dataset, 27.7% (4.2% in terms of $\ln(AMRE)$) on Sin dataset, and 28.9% (4.5% in terms of $\ln(AMRE)$) on Log dataset. Moreover, our PBD consistently outperforms BD.

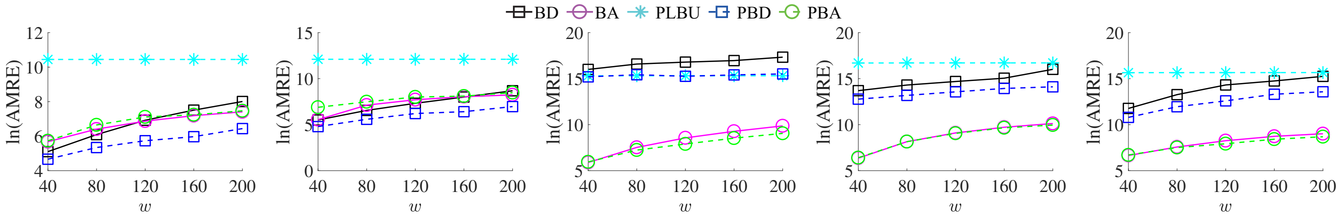
Figure 11 shows the natural logarithm of *AMRE* as the window size w varies. As w increases, *AMRE* rises gently, particularly on the synthetic datasets. This occurs because a large window size results in a small privacy budget at each time slot, leading to increased error. PLBU shows lower performance than other methods on all datasets except for TLNS, since LDP methods achieve lower accuracy than CDP methods under equivalent privacy budgets. For real datasets, our PBD achieves the lowest error compared to others methods. The *AMRE* of PBD is on average 63.1% (15.6% in terms of $\ln(AMRE)$) lower than that of BD on Taxi dataset and 68.4% (16.5% in terms of $\ln(AMRE)$) on Foursquare dataset. For synthetic datasets, our PBA demonstrates the lowest error among all methods. Compared to BA, the *AMRE* of PBA is lower by average of 35.1% (5.4% in terms of $\ln(AMRE)$) for TLNS, 4.2% (0.4% in terms of $\ln(AMRE)$) for Sin, and 16.6% (2.2% in terms of $\ln(AMRE)$) for Log. Moreover, our PBD consistently outperforms BD across all datasets.

In summary, our PBD demonstrates superior performance on real datasets, with an *AMRE* at least 63% lower than BD. For synthetic datasets, our PBA outperforms BA with at least 16% smaller *AMRE*.



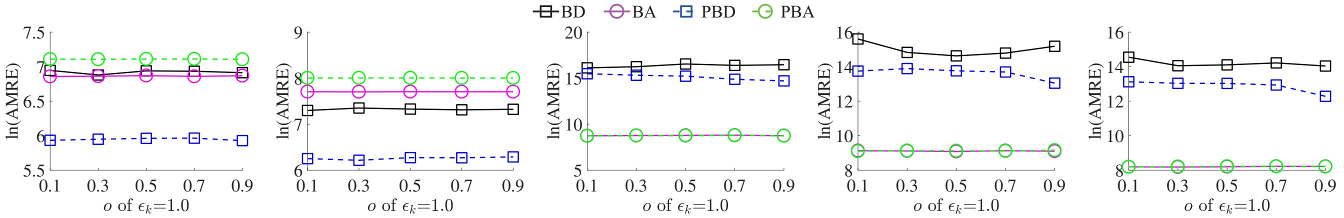
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Figure 10: ***AMRE*** **with ϵ varied.**



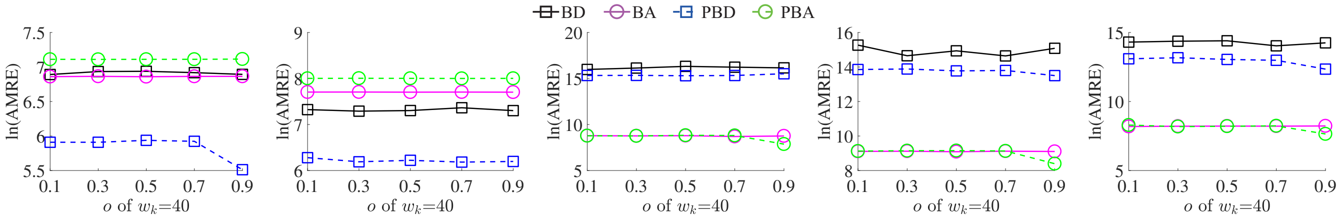
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Figure 11: ***AMRE*** **with w varied.**



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

Figure 12: ***AMRE*** **with ratio for privacy budget varied.**



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

Figure 13: ***AMRE*** **with ratio for window size varied.**

5.5 Impact of User Requirement Type

We define a set of users with privacy requirement as (w_k, ϵ_k)-*requirement type*. In this subsection, we examine the impact of user type on the utility. For our analysis, we set ϵ_k candidate set as $\{0.6, 1.0\}$ with a default value of 0.6, and the w_k candidate set as $\{40, 120\}$ with a default value of 120. We first vary the users' quantity ratio of $\epsilon_k = 1.0$ from 0.1 to 0.9 while keeping $w_k = 120$, and then vary the users' quantity ratio of $w_k = 40$ from 0.1 to 0.9 while keeping $\epsilon_k = 0.6$. We analyze the impact of these ratio variations on *AMRE*.

Figure 12 illustrates the change in users' quantity ratio for $\epsilon_k = 1.0$ from 0.1 to 0.9, with a fixed window size of $w_k = 120$. Figure 13 shows the effect on changing users' quantity for $w_k = 40$ from 0.1 to 0.9, with a fixed privacy budget of $\epsilon_k = 0.6$. We observe that as the users' quantity ratio increases, the *AMRE* remains relatively stable. However, when the users' quantity ratio of $\epsilon_k = 1.0$ or $w_k = 40$ exceeds 0.8, we can see a significant decrease in *AMRE* for PBD and PBA. This occurs because when the ratios surpasses a certain threshold, the optimal budget from OBS in Algorithm 1 becomes dominated by a higher ϵ, resulting in lower error.

6 CONCLUSION

In this paper, we address the problem of Personalized w-event Private Publishing for Infinite Data Streams. We propose a mechanism called PWSM and two methods called PBD and PBA to solve this problem in scenarios with personalized privacy budget and window sizes for each users. We also compare our PBD and PBA with recent solutions to demonstrate their efficiency and effectiveness.

REFERENCES

[1] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. 2016. Heterogeneous Differential Privacy. *J. Priv. Confidentiality* 7, 2 (2016).

[2] Miguel E. Andrés, Nicolás Emilio Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: differential privacy for location-based systems. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung (Eds.). ACM, 901–914.

[3] Ergute Bao, Yin Yang, Xiaokui Xiao, and Bolin Ding. 2021. CGM: An Enhanced Mechanism for Streaming Data Collectionwith Local Differential Privacy. *Proc. VLDB Endow.* 14, 11 (2021), 2258–2270.

[4] Raef Bassily and Adam D. Smith. 2015. 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*, Rocco A. Servedio and Ronitt Rubinfeld (Eds.). ACM, 127–135.

[5] Avrim Blum, Katrina Ligett, and Aaron Roth. 2008. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, Cynthia Dwork (Ed.). ACM, 609–618.

[6] Jean Bolot, Nadia Fawaz, S. Muthukrishnan, Aleksandar Nikolov, and Nina Taft. 2013. Private decayed predicate sums on streams. In *Joint 2013 EDBT/ICDT Conferences, ICDT '13 Proceedings, Genoa, Italy, March 18-22, 2013*, Wang-Chiew Tan, Giovanna Guerrini, Barbara Catania, and Anastasios Gounaris (Eds.). ACM, 284–295.

[7] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and Continual Release of Statistics. *ACM Trans. Inf. Syst. Secur.* 14, 3 (2011), 26:1–26:24.

[8] Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2017. PeGaSus: Data-Adaptive Differentially Private Stream Processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 1375–1388.

[9] Rachel Cummings, Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2022. Mean Estimation with User-level Privacy under Data Heterogeneity. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[10] Leilei Du, Peng Cheng, Libin Zheng, Wei Xi, Xuemin Lin, Wenjie Zhang, and Jing Fang. 2023. Dynamic Private Task Assignment under Differential Privacy. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*. IEEE, 2740–2752.

[11] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II (Lecture Notes in Computer Science)*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.), Vol. 4052. Springer, 1–12.

[12] Cynthia Dwork. 2010. Differential Privacy in New Settings. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, Moses Charikar (Ed.). SIAM, 174–183.

[13] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, Leonard J. Schulman (Ed.). ACM, 715–724.

[14] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407.

[15] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, Timothy M. Chan (Ed.). SIAM, 2468–2479.

[16] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. 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*, Gail-Joon Ahn, Moti Yung, and Ninghui Li (Eds.). ACM, 1054–1067.

[17] Liyue Fan and Li Xiong. 2014. An Adaptive Approach to Real-Time Aggregate Monitoring With Differential Privacy. *IEEE Trans. Knowl. Data Eng.* 26, 9 (2014), 2094–2106.

[18] Maria Feijoo-Cid, Antonia Arreciado Marañón, Ariadna Huertas, Amado Rivero-Santana, Carina Cesar, Valeria Fink, María Isabel Fernández-Cano, and Omar Sued. 2023. 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* 31, 3 (2023), 135–155.

[19] Peng Guo, Tao Jiang, Qian Zhang, and Kui Zhang. 2012. Sleep Scheduling for Critical Event Monitoring in Wireless Sensor Networks. *IEEE Trans. Parallel Distributed Syst.* 23, 2 (2012), 345–352.

[20] Zach Jorgensen, Ting Yu, and Graham Cormode. 2015. Conservative or liberal? Personalized differential privacy. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, Johannes Gehrke, Wolfgang Lehner, Kyuseok Shim, Sang Kyun Cha, and Guy M. Lohman (Eds.). IEEE Computer Society, 1023–1034.

[21] Matthew Joseph, Aaron Roth, Jonathan R. Ullman, and Bo Waggoner. 2018. Local Differential Privacy for Evolving Data. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 2381–2390.

[22] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. 2014. Differentially Private Event Sequences over Infinite Streams. *Proc. VLDB Endow.* 7, 12 (2014), 1155–1166.

[23] Ios Kotsogiannis, Stelios Doudalis, Samuel Haney, Ashwin Machanavajjhala, and Sharad Mehrotra. 2020. One-sided Differential Privacy. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE, 493–504.

[24] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[25] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* 37, 1 (1991), 145–151.

[26] Junxu Liu, Jian Lou, Li Xiong, Jinfei Liu, and Xiaofeng Meng. 2021. Projected Federated Averaging with Heterogeneous Differential Privacy. *Proc. VLDB Endow.* 15, 4 (2021), 828–840.

[27] WonJun Moon, Sangeek Hyun, Sanguk Park, Dongchan Park, and Jae-Pil Heo. 2023. Query - Dependent Video Representation for Moment Retrieval and Highlight Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 23023–23033.

[28] Takao Murakami and Yusuke Kawamoto. 2019. Utility-Optimized Local Differential Privacy Mechanisms for Distribution Estimation. In *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, Nadia Heninger and Patrick Traynor (Eds.). USENIX Association, 1877–1894.

[29] Xuebin Ren, Liang Shi, Weiren Yu, Shusen Yang, Cong Zhao, and Zongben Xu. 2022. 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*, Zachary Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 1064–1077.

[30] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. 2016. RescueDP: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. In *35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*. IEEE, 1–9.

[31] Tianhao Wang, Joann Qiongna Chen, Zhikun Zhang, Dong Su, Yueqiang Cheng, Zhou Li, Ninghui Li, and Somesh Jha. 2021. 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*, Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi (Eds.). ACM, 1237–1253.

[32] Zhibo Wang, Jiahui Hu, Ruizhao Lv, Jian Wei, Qian Wang, Dejun Yang, and Hairong Qi. 2019. Personalized Privacy-Preserving Task Allocation for Mobile Crowdsensing. *IEEE Trans. Mob. Comput.* 18, 6 (2019), 1330–1341.

[33] Zhibo Wang, Wenxin Liu, Xiaoyi Pang, Ju Ren, Zhe Liu, and Yongle Chen. 2020. Towards Pattern-aware Privacy-preserving Real-time Data Collection. In *39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*. IEEE, 109–118.

[34] Yutong Xie, Zhaoying Pan, Jinge Ma, Luo Jie, and Qiaozhu Mei. 2023. A Prompt Log Analysis of Text-to-Image Generation Systems. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 3892–3902.

[35] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. 2015. NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *J. Netw. Comput. Appl.* 55 (2015), 170–180.

[36] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory Cultural Mapping Based on Collective Behavior Data in Location-Based Social Networks. *ACM Trans. Intell. Syst. Technol.* 7, 3 (2016), 30:1–30:23.

[37] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, Chid Apté, Joydeep Ghosh, and Padhraic Smyth (Eds.). ACM, 316–324.

[38] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories. In *18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2010, November 3-5, 2010, San Jose, CA, USA, Proceedings*, Divyakant Agrawal, Pusheng Zhang, Amr El Abbadi, and Mohamed F. Mokbel (Eds.). ACM, 99–108.

7 APPENDIX

7.1 Running time Analysis

In this subsection, we compare the running time of BD, BA, PBD and PBA.

Figure 14 shows the running time as the privacy budget varies from 0.2 to 1. For synthetic datasets, the running time remains stable across different privacy budgets. This stability occurs because as datasets change gradually and skipped time slots increase, different privacy budgets have minimal impact on the number of new publications. In real datasets, the running time of BA increases slightly, likely due to larger privacy budgets requiring more comparisons between dissimilarity and error when datasets change rapidly. PBD requires the highest computation time among all methods, particularly with synthetic datasets, while BD requires the least time for real datasets and some synthetic datasets (i.e., TLNS and Sin). It is because non-personalized methods (BD and BA) have fewer steps in BD than in BA for publication judgments (which is denoted as algorithm complexity running time, TC_{ac}). As a result, BD requires less time than BA for most datasets. Personalized methods (PBD and PBA), however, require additional steps for optimal budget selection. These methods also have a higher probability of dissimilarity exceeding error (since they achieve lower error rates than non-personalized methods), resulting in fewer skips or nullifications compared to non-personalized methods. Fewer skips or nullifications leads to more comparisons in the total stream publication (which is defined as publication number running time, TC_{pn}). When time slots are sufficiently large, TC_{pn} has a greater impact than TC_{ac}. This effect becomes particularly noticeable with data changing slowly (as seen in synthetic datasets).

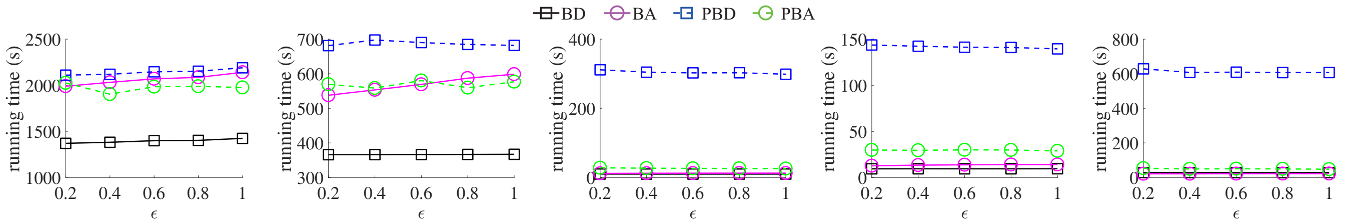
Figure 15 shows the running time as the window size changes from 40 to 200. All methods except PBD maintain stable running times as the window size increases. For PBD, its running time increases when the window sizes increase, because larger windows result in smaller per-user privacy budgets within each window. Reserving half of the privacy budget for future publications leads to larger dissimilarity and error. Since PBD's optimal budget selection step mitigates error's growth, the dissimilarity increases at a lower rate than the error, resulting in more frequent publications. Similar to Figure 14, BD requires the least running time among all methods on real datasets and most synthetic datasets (i.e., TLNS and Sin), while PBD requires the highest running time. It is because TC_{ac} is lower in BD than in BA. Additionally, personalized methods introduce an optimal budget selection step that increases the running time by TC_{pn}. Compared to PBA, the dissimilarity of PBD increases more rapidly than the error, resulting in fewer skips or nullifications and thus a larger TC_{pn} in PBD. When time slots are sufficiently large, TC_{pn} has a greater impact than TC_{ac}, causing PBD to have the longest running time.

7.2 Experimental Result under $AJSD$ Metric

In this subsection, we compare the performance of BD, BA, PLBU, PBD and PBA using $AJSD$ metric.

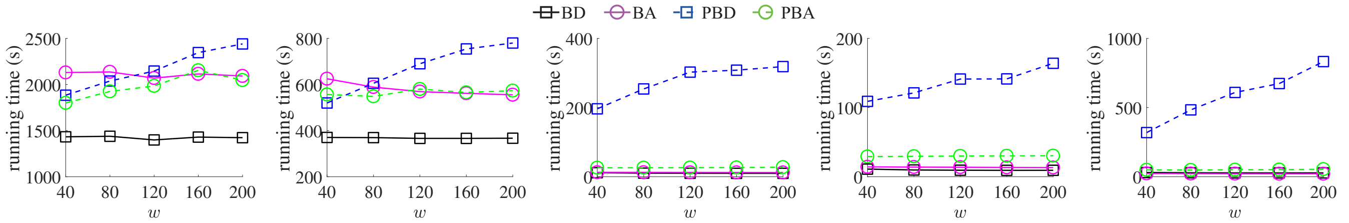
Figure 16 shows the results of $AJSD$ as the privacy budget $\mathcal{E}$ varies from 0.2 to 1. For all methods, $AJSD$ decreases as $\mathcal{E}$ increases, which aligns with the $AMRE$ results in Section 5.4. PLBU performs worse than other methods across all datasets except TLNS as LDP methods achieve lower accuracy than CDP methods under the same privacy budget. Both PBD and PBA consistently outperform BD. PBD achieves the best accuracy on the Taxi dataset, while PBA performs best with the three synthetic datasets. In the Foursquare dataset, BA outperforms other methods, due to the dataset's sparsity causing larger error in $AJSD$ calculation.

Figure 17 shows the results of $AJSD$ as the window size w varies from 20 to 200. $AJSD$ also increases with larger window sizes for all methods. PLBU shows lower utility than other methods in all datasets except TLNS, since LDP methods achieve lower accuracy than CDP methods under equivalent privacy budgets. Consistent with the results in Figure 16, both PBD and PBA outperform BD. PBD achieves the best performance in the Taxi dataset, while PBA leads in the three synthetic datasets. For the Foursquare dataset, BA achieves the lowest $AJSD$. However, this result may be unreliable due to the dataset's sparsity, which causes large calculation errors in the $AJSD$ measurements.



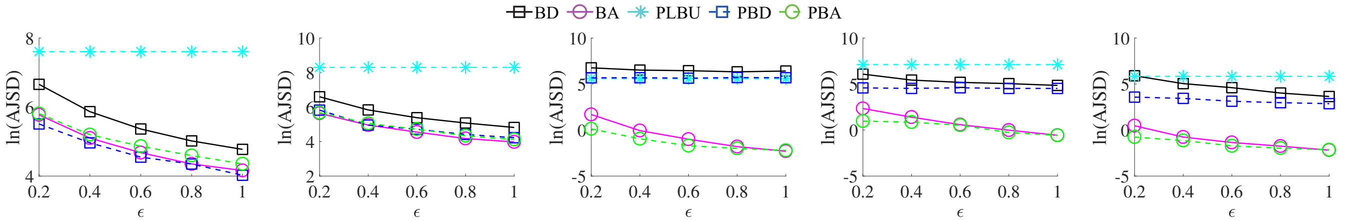
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Figure 14: The running time with ϵ varied.



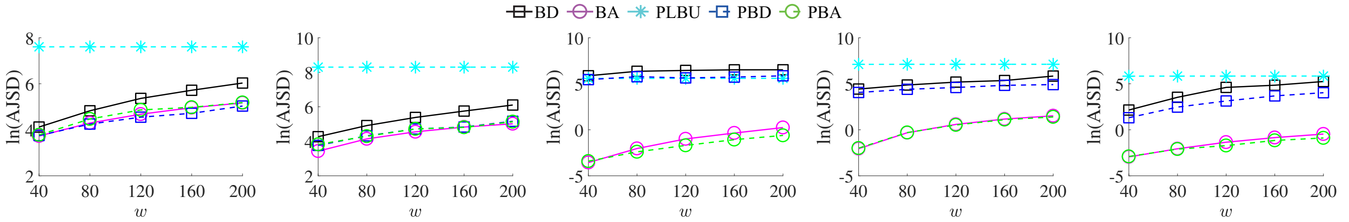
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Figure 15: The running time with w varied.



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

Figure 16: The $AJSD$ with ϵ varied.



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

Figure 17: The $AJSD$ with w varied.