Noname manuscript No.
(will be inserted by the editor)

Personalized w-Event Privacy for Infinite Stream Estimation

Leilei Du[1] · Xu Zhou[1] · Peng Cheng[2] · Lei Chen[3,4] · Xuemin Lin[5] · Wei Xi[6] · Kenli Li[1]

Received: date / Accepted: date

Abstract In many real-life applications, such as event monitoring, log analysis and video querying, w-event privacy is widely used to protect individual privacy within a given time window while maintaining high accuracy in data collection. However, existing w-event privacy studies on infinite data stream typically focus only on homogeneous privacy requirements for all users. In this paper, we propose personalized w-event privacy protection that enables users to set different privacy requirements in private data stream estimation. Specifically, we first design a Personalized Window Size Mechanism (PWSM) that allows users to maintain personalized privacy requirements at each time slot. Then, we propose two solutions—Personalized Budget Distribution (PBD) and Personalized Budget Absorption (PBA)—to accurately estimate streaming data statistics while achieving $\boldsymbol{w}$-Event $\boldsymbol{\mathcal{E}}$ Personalized Differential Privacy (($\boldsymbol{w}$, $\boldsymbol{\mathcal{E}}$)-EPDP). PBD ensures that the privacy budget for the next time step is at least equal to the amount consumed in the previous release. PBA enhances the current time slot's privacy budget by combining the privacy budget from the previous k time slots and borrowing from the next k time slots. In addition, we design two additional solutions—Dynamic Personalized Budget Distribution (DPBD) and Dynamic Personalized Budget Absorption (DPBA)—that allow users to dynamically adjust their privacy requirements at each time slot while achieving (τ, $\boldsymbol{w}_B$, $\boldsymbol{w}_F$)-Event ($\boldsymbol{\mathcal{E}}_B$, $\boldsymbol{\mathcal{E}}_F$)-Personalized Differential Privacy ((τ, $\boldsymbol{w}_B$, $\boldsymbol{w}_F$, $\boldsymbol{\mathcal{E}}_B$, $\boldsymbol{\mathcal{E}}_F$)-EPDP). The proposed methods are all proven to achieve personalized differential privacy levels and establish error upper bounds for each method. From the experimental results, our methods outperform the state-of-the-art algorithms with at least 53.6% smaller error.

Keywords Differential privacy · Stream data · Event privacy · Personalized privacy

Leilei Du
leileidu@hnu.edu.cn

Xu Zhou
zhxu@hnu.edu.cn

Peng Cheng
cspcheng@tongji.edu.cn

Lei Chen
leichen@cse.ust.hk

Xuemin Lin
xuemin.lin@gmail.com

Wei Xi
xiwei@xjtu.edu.cn

Kenli Li
lkl@hnu.edu.cn

[1] Hunan University, Changsha, China
[2] Tongji University, Shanghai, China
[3] HKUST (GZ), Guangzhou, China
[4] HKUST, HK SAR, China
[5] Shanghai Jiaotong University, Shanghai, China
[6] Xi'an Jiaotong University, Xi'an, China

1 Introduction

With the widespread adoption of smart devices and wireless networks, more people are sharing and receiving data through various platforms, making real-time analysis (e.g., event monitoring [23], log analysis [42], and video querying [32]) increasingly necessary. During these data collection and analysis processes, protecting users' privacy is crucial. To prevent user data leakage, Differential Privacy (DP) has emerged as a widely adopted solution in data publishing and statistical analysis.

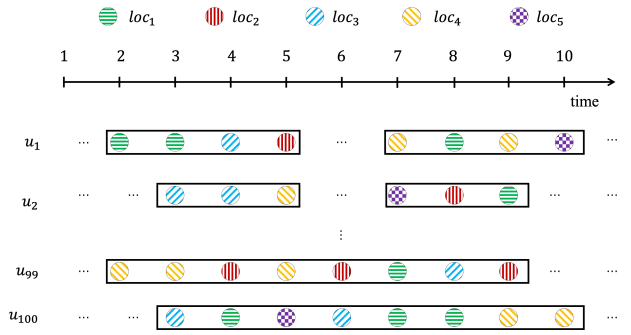


Fig. 1: Different event window sizes for different time slots.

Existing w-event privacy mechanisms [37,41,34] based on DP protect user data privacy across consecutive events. However, these mechanisms use uniform privacy requirements for all users (applying the same privacy budget $\mathcal{E}$ and window size w). This one-size-fits-all approach has significant limitations in real-world applications. For example, celebrities in the entertainment industry may require strong protection of their location data, while street artists may want to share their locations to gain visibility. The fixed privacy budget and window size thus provide inadequate privacy protection for some users while unnecessarily increasing data error rates (excessive privacy protection) for others.

Example 1 Consider an example of online car-hailing shown in Figure 1. It has 100 drivers $U = \{u_1, ..., u_{100}\}$ who share their locations from $\{loc_1, ..., loc_5\}$ at each time slot. Each driver u_i is protected by w_i-event privacy, meaning their location data is safeguarded through $\mathcal{E}$-DP across at least w_i consecutive time slots, where $\mathcal{E}$ represents their required privacy protect strength. For example, u_1 requires location protection across any 4 consecutive time slots, while u_{99} and u_{100} need protection across any 8 consecutive time slots. For the drivers $u_i \in U \backslash \{u_{99}, u_{100}\}$, the window sizes do not exceed 4.

By traditional w-event privacy with $\mathcal{E} = 1$, it necessitates setting the event window sizes to the maximal value (i.e., $w = 8$) for satisfying all drivers' privacy requirements. While this achieves 8-event privacy—providing the strongest privacy protection—it also results in the lowest data utility. Let AE_{avg} denote the average square error at each time slot. When using the Laplace mechanism, the error at any time slot equals the variance of added Laplace noise $\left(\text{i.e., } AE_{avg} = 2 \times \left(\frac{1}{\epsilon/w}\right)^2\right)$. Using the *Uniform method* [25], the error is $AE_{avg} = 2 \times (\frac{w}{\epsilon})^2 = 128$ under 8-event privacy. While the first 98 drivers only require 4-event privacy. Setting the privacy level to 4-event privacy would reduce the error to $AE_{avg} = 2 \times \left(\frac{w}{\epsilon}\right)^2 = 32$. However, 4-event privacy only protects data within a window size of 4, which cannot meet the privacy requirements of the 99-th and the 100-th drivers who need protection within a window size of 8, thus compromising their privacy.

Challenges. From the example above, there are three main challenges:

(1) **Unified Privacy Budget.** Traditional DP requires a uniform privacy budget ϵ for all users to achieve ϵ-DP. However, users have distinct privacy budgets (a type of privacy requirements). While setting ϵ to the minimum value would satisfy everyone' privacy requirements, this approach significantly reduces data utility. The challenge lies in unifying users' distinct privacy budgets into a single value while maximizing the utility of published data.

(2) **Personalized Privacy Budget Allocation.** The rate of change in streaming data fluctuates over time. Within a given window size, time slots with rapid changes contain more information compared to those with slower changes. Since the privacy budget serves as a privacy protection resource, it should be allocated primarily to time slots containing more information. The challenge is determing how to optimally distribute each user's personalized privacy budget across their privacy window size.

(3) **Dynamic Privacy Requirements.** Users have different privacy requirements at different time slots. These varing requirements can lead to privacy budget waste or privacy requirement conflicts between current and historical time slot. The challenge is how to allocate varying privacy budgets while maintaining high utility.

A further challenge is that the dynamic personalized setting is not a straightforward extension of the fixed one. In the fixed case, each user keeps the same privacy requirement over time. In contrast, in the dynamic case, privacy requirements may vary across timestamps, so each release must remain consistent with previously consumed privacy budgets while preserving feasibility for future requirements. Moreover, although privacy constraints are specified at the user level, the system still publishes one shared aggregate result at each timestamp. Therefore, the dynamic personalized setting introduces a new online feasibility problem under heterogeneous time-varying constraints.

Contributions. This paper studies a more general problem than the fixed personalized setting, namely Dynamic Personalized w-Event Private Publishing for Infinite Data Streams (DPWEPP-IDS), where each user may specify time-varying backward and forward privacy requirements. This setting is practically important because privacy preferences in real systems may evolve over time, and technically challenging because each release must remain compatible with both historical budget consumption and future privacy feasibility. To address this problem, we develop a unified view of personalized stream release. Our main observation is that heterogeneous personalized privacy requirements must ultimately be transformed into a valid system-level release decision, because only one aggregate statistic is published at each time slot. The main contribution is not replacing a global window or budget with personalized parameters, but

constructing and maintaining a valid shared release budget under overlapping, heterogeneous, and time-varying personalized privacy requirements. This heterogeneous-to-release unification challenge does not arise in classical homogeneous w-event privacy. We summarize our contributions as follows:

- We formulate Dynamic Personalized w-Event Private Publishing for Infinite Data Streams and define the corresponding privacy notation, namely (τ, $\boldsymbol{w}_B$, $\boldsymbol{w}_F$)-Event ($\boldsymbol{\mathcal{E}}_B$, $\boldsymbol{\mathcal{E}}_F$)-Personalized Differential Privacy ((τ, $\boldsymbol{w}_B$, $\boldsymbol{w}_F$, $\boldsymbol{\mathcal{E}}_B$, $\boldsymbol{\mathcal{E}}_F$)-EPDP), which generalize the fixed personalized setting in Section 3.
- We identify a new online feasibility challenge under dynamic personalized privacy: at each timestamp, the mechanism must reconcile heterogeneous user-specific requirements with both past budget consumption and future privacy feasibility, while still producing one shared aggregate release in Section 3.
- We propose a unified framework for personalized stream release. In the fixed setting, this framework is instantiated as PWSM with two mechanisms, PBD and PBA in Section 4. In the dynamic setting, it is generalized to DPWSM with two mechanisms, DPBD and DPBA in Section 5.
- We provide privacy guarantees and utility bounds for all mechanisms, analyze their computational properties in Section 4 and 5, and experimentally evaluate them on real and synthetic datasets in Section 6.

Compared with the conference version [13], this paper studies a more general setting in which each user's privacy requirement may vary over time. This generalization introduces a new online feasibility problem: each release must satisfy heterogeneous user-specific privacy constraints while remaining consistent with previously consumed budgets and feasible for future requirements. To address this problem, we develop the dynamic framework DPWSM together with two mechanisms, DPBD and DPBA, and provide corresponding privacy guarantees, utility analysis, and new experiments for this generalized setting.

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 three types of data stream estimation methods: centralized differential privacy [16] (CDP) based methods, local differential privacy [4] (LDP) based methods and shuffled differential privacy [9, 10].

Data Stream Estimation under CDP. Dwork et al. [18] first address the problem of Differential Privacy (DP) on data streams. They define two types of DP levels: *event-level differential privacy* (event-DP) and *user-level differential privacy* (user-DP).

Table 1: Summary for related work.

Model Types		Methods	Infinite & correlated	Personalized privacy
Centralized DP	event-level privacy	Finite B-tree [18]	✗	✗
		Infinite B-tree [7]	✗	✗
		Adaptive-density Counter [17]	✗	✗
		Decayed Privacy [6]	✗	✗
		PeGaSus [8]	✗	✗
	user-level privacy	FAST [21]	✓	✗
		Private heterogeneous mean estimation [11]	✓	✗
		Dynamic user-DP [12]	✓	✗
		DPI [22]	✓	✗
		SMM-TM, RBM [15]	✓	✗
	w-event privacy	BD, BA [25]	✓	✗
		ResuseDP [37]	✓	✗
		SPAS [29]	✓	✗
Local DP	event-level privacy	RAPPOR [20]	✗	✗
		ToPL [39]	✗	✗
	user-level privacy	CGM [3]	✓	✗
		DDRM [43]	✓	✗
		StaSwitch [46]	✓	✗
	w-event privacy	LDP-IDS [34]	✓	✗
Shuffled DP	event-level privacy	Concurrent-SDP [36]	✓	✗
	user-level privacy	LPS-SS [28]	✗	✗
		ExSub [38]	✗	✗
Item heterogeneous		HDP [1]	✗	✗
Record heterogenous		PDP [24]	✗	✓
		OSDP [26]	✗	✓
		Geo-I [2]	✗	✓
		PWSM, VPDM [40]	✗	✓
		PUCE, PGT [14]	✗	✓
		PFA, PFA+ [31]	✗	✓
Our mechanisms			✓	✓

In event-DP, each single event is hidden in statistic queries. Dwork et al. [18] focus on the finite event scenarios and propose a binary tree method to achieve high statistical utility while maintaining event-DP. Chan et al. [7] extend it to infinite cases, and produce partial summations for binary counting. Dwork et al. [17] introduce a cascade buffer counter that updates adaptively based on stream density. Bolot et al. [6] propose *decayed privacy* which reduces the privacy costs for past data. Chen et al. [8] develop PeGaSus, a perturb-group-smooth framework for multiple queries under event-DP. 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. [21] propose the FAST algorithm, which uses a sampling-and-filtering framework to count finite stream data under user-DP. Cummings et al. [11] address heterogeneous user data by estimating population-level means while achieving user-DP. However, they only consider finite data. Dong et al. [12] introduce continual observation mechanisms under user-DP for dynamic data streams, achieving utility guarantees without prior data restrictions and providing down-neighborhood optimality for count and sum functions. However, their approach assumes

independence between different stages. Feng et al. [22] develop the DPI framework with bidirectional reweighting, 0-DP synopsis generation, and dynamic error control, ensuring that privacy preservation does not significantly degrade accuracy over time. Dvijotham et al. [15] tackle cascading correlations in data through two methods: Streaming Matrix Multiplication for Toeplitz Matrices (SMM-TM) and Recursive Binary Tree Mechanism (RBM). These approaches reduce the impact of data dependencies on differential privacy in streaming continual counting tasks. However, providing user-DP for infinite data requires infinite perturbation, resulting in poor long-term utility [25].

To bridge the gap between event-DP and user-DP, Kellaris et al. [25] propose w-event DP for infinite streams. This ensures ϵ-DP for any group of events within a time window of size w. They introduce two methods, *Budget Distribution* (BD) and *Budget Absorption* (BA), to optimize privacy budget use and estimate statistics effectively. However, neither method handles stream data with significant changes. Wang et al. [37] 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. Li et al. [29] propose *SPAS* for the continuous release of infinite data streams under w-event differential privacy. It improves adaptability through data-dependent strategy prediction, adaptive sampling, and privacy budget allocation. However, SPAS assumes a single global privacy requirement and does not support heterogeneous user-specific privacy budgets or window sizes, nor dynamically changing personalized requirements over time. Overall, existing centralized methods for w-event private stream release, including BD, BA, ResueDP, and SPAS, are designed for the classical homogeneous setting with a single global privacy requirement, and do not model heterogeneous user-specific privacy requirements or their time-varying extensions.

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. Similary to DP, data stream estimation under LDP can be classified into event-LDP, user-LDP and w-event LDP.

Erlingsson et al. [20] introduce RAPPOR to estimate finite streams under LDP. They design a two-layer randomized response mechanism (i.e., permanent randomized response and instantaneous randomized response) to protect each individual's data. Wang et al. [39] extend event-level privacy from CDP to LDP and design the efficient ToPL method under event LDP. Nevertheless, both RAPPOR and ToPL focuses solely on event-level privacy, lacking privacy protection for correlated data in streams.

To address the problem of correlated time series data, Bao et al. [3] propose CGM, an (ϵ, δ)-LDP method that uses the analytic Guassian mechanism for streaming data collection. However, CGM is limited to finite streaming data. Xue et al. [43] introduce DDRM for continual frequency estimation under LDP. While it dynamically allocates privacy budgets and employs difference trees to reduce unnecessary consumption, DDRM suffers from eventual budget depletion, which compromises estimation accuracy. Ye et al. [46] develop the StaSwitch mechanism, which employs a stateful switch operation for efficient privacy budgetmanagement. Though this allows flexible privacy parameter settings and improves data utility, the budget still accumulates over time.

Ren et al. [34] introduce LDP-IDS for infinite streaming data collection and analysis under w-event LDP. 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.

Data Stream Estimation under SDP. Tenenbaum et al. [36] propose a shuffle-based continual observation mechanism that supports concurrent streaming queries with provable accuracy guarantees. However, its privacy notion is limited to event-level and does not extend to user-level protection. Li et al. [28] propose a shuffle-based LDP streaming framework with subsampling that achieves double privacy amplification and improved utility. However, it is only suitable for finite stream data. Wang et al. [38] propose ExSub, a user-level differentially private streaming analytics framework under the local and shuffle models that achieves near-centralized accuracy for finite-length data streams. However, it relies on bounded user changes and a predefined time horizon.

2.2 Personalized and Heterogeneous Differential Privacy

Recently, some studies address the non-uniform privacy requirements among items (table columns) or records (table rows) [33].

Alaggan et al. [1] first examine scenarios where each database instance comprises a single user's profile. They focus on varying privacy requirements for different items and formally define Heterogeneous Differential Privacy (HDP). Jorgensen et al. [24] investigate the privacy preservation for individual rows, introducing Personalized Differential Privacy (PDP). They design two mechanisms leveraging non-uniform privacy requirements to achieve better utility than standard uniform DP. Kotsogiannis et al. [26] recognize that different data have different sensitivity, then define One-sided Differential Privacy (OSDP) and propose algorithms that truthfully release non-sensitive record samples to enhance accuracy in DP-solutions. Andrés et al. [2] 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. Wang et al. [40] and Du et al. [14] explore PDP in spatial crowd-

sourcing, and develop highly effective private task assignment methods to satisfy diverse workers' privacy and utility requirements. Liu et al. [31] investigate HDP in federated learning. 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. More recently, Sun et al. [35] propose Personalized Truncation for personalized differential privacy (PDP) in count, sum, and SJA query processing. This line of work is related in spirit but does not consider infinite-stream continual release or w-event privacy. However, all above studies are not suitable for stream data.

3 Problem Settings

In this section, we introduce key concepts, including data streams and differential privacy (DP). We then define two types of personalized privacy requirements that address different real-world scenarios. Finally, we provide the problem definition of Dynamic Personalized w-Event Private Publishing for Infinite Data Streams (DPWEPP-IDS). Table 2 summarizes the notations used throughout this paper.

Table 2: Notations.

Notations	Description
$\mathcal{D}$	the database domain
D_t	a database at time slot t
S	a data stream
U	the user set
u_i	the i-th user in U
$\boldsymbol{x}_{i,t}$	u_i's data at time slot t
$\boldsymbol{c}_t$	a real statistical histogram at time slot t
$\boldsymbol{r}_t$	an estimation statistic histogram at time slot t
$\boldsymbol{\epsilon}$	all users' privacy budget requirement at any time slot
ϵ_i	u_i's privacy budget requirement in at any time slot
$\boldsymbol{w}$	all users' fixed window size requirements
w_i	u_i's fixed window size requirement
$\boldsymbol{\mathcal{E}}$	all users' fixed privacy budget requirements
$\mathcal{E}_i$	u_i's fixed privacy budget requirement
$\boldsymbol{w}_{B,t}$	all users' backward window size requirements at time slot t
$w_{B,i,t}$	u_i's backward window size requirement at time slot t
$\boldsymbol{\mathcal{E}}_{B,t}$	all users' backward privacy budget requirement at time slot t
$\mathcal{E}_{B,i,t}$	u_i's backward privacy budget requirement at time slot t
$\boldsymbol{w}_{F,t}$	all users' forward window size requirement at time slot t
$w_{F,i,t}$	u_i's forward window size requirement at time slot t
$\boldsymbol{\mathcal{E}}_{F,t}$	all users' forward privacy budget requirement at time slot t
$\mathcal{E}_{F,i,t}$	u_i's forward privacy budget requirement at time slot t

3.1 Data Stream

Definition 1 (Data Stream [25]). 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, a substream between time slot t_l and t_r (where $t_l < t_r$) is denoted as $S_{t_l,t_r} = [D_{t_l}, D_{t_l+1}, \ldots, D_{t_r}]$. When $t_l = 1$, we denote $S_t = [D_1, D_2, \ldots, D_t]$ as the *stream prefix* of S.

Definition 2 (Data Stream Count Publishing). Let $Q : \mathcal{D} \to \mathbb{R}^d$ 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.

Definition 3 (w-neighboring stream prefixes [7,25]). 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 $k \le t$ and $S_t[k] \neq S'_t[k]$, it holds that $S_t[k]$ and $S'_t[k]$ are neighboring [25] 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 $k_2 - k_1 + 1 \le w$.

Definition 4 ((τ, w)-backward neighboring stream prefixes). Let w be a positive integer. Two stream prefixes S_t, S'_t are (τ, w)-backward neighboring (denoted as $S_t \sim_{B,\tau,w} S'_t$), if

1. for each $S_t[k]$, $S'_t[k]$ such that $k \in [t]$ and $S_t[k] \neq S'_t[k]$, it holds that $S_t[k]$ and $S'_t[k]$ are neighboring, and
2. for each $S_t[k]$, $S'_t[k]$, with $k < \tau$, $S_t[k] \neq S'_t[k]$, it holds that $\tau - k + 1 \le w$.

Definition 5 ((τ, w)-forward neighboring stream prefixes). Let w be a positive integer. Two stream prefixes S_t, S'_t are w-forward neighboring (denoted as $S_t \sim_{F,\tau,w} S'_t$), if

1. for each $S_t[k]$, $S'_t[k]$ such that $k \in [t]$ and $S_t[k] \neq S'_t[k]$, it holds that $S_t[k]$ and $S'_t[k]$ are neighboring, and
2. for each $S_t[k]$, $S'_t[k]$, with $k > \tau$, $S_t[k] \neq S'_t[k]$, it holds that $k - \tau + 1 \le w$.

Remark 1 (Equivalence of forward and shifted backward neighboring). For any τ and w, the (τ, w)-forward neighboring relation is equivalent to the $(\tau + w - 1, w)$-backward neighboring relation, since both restrict the changed events to the same window $[\tau, \tau + w - 1]$. We nevertheless keep both notations because they correspond to two semantically different types of dynamic privacy requirements in our model: a backward requirement is anchored at the current time slot and constrains privacy loss accumulated from the recent past, whereas a forward requirement is declared at the current time slot and constrains feasible future releases in the upcoming window. This distinction is convenient for stating user requirements and for presenting the online feasibility rules in Section 5.

3.2 Differential Privacy

There are two parts in the differential privacy paradigm: a large number of respondents (data owners) and a trust curator (server). The goal of differential privacy mechanisms is to publish statistic of D while not comprise the privacy of respondents.

Threat Model. This work follows the centralized-DP paradigm. A trusted curator collects the stream data $x_{i,t}$ and the corresponding personalized privacy requirements from users, and releases only perturbed aggregate statistics r_t. The adversary is any party that observes the released sequence $(r_1, r_2, \dots)$. As is standard in differential privacy, we assume the adversary knows the mechanism, the neighboring relation, and all public system parameters, and may have arbitrary side information. Our privacy objective is therefore to bound the information leaked about any user's contribution within the relevant event window through the released outputs. This trusted-curator assumption is consistent with prior centralized w-event stream release methods reviewed in Section 2.

Definition 6 (ϵ-differential privacy [16,25]). A mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{O}$ satisfies ϵ-differential privacy (or ϵ-DP), where $\epsilon \geq 0$ if for all sets $O \subseteq \mathcal{O}$, and every pair of neighboring databases $D, D' \in \mathcal{D}$, it holds that

$$\Pr[\mathcal{M}(D) \in O] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in O].$$

Definition 7 ($\boldsymbol{\epsilon}$-Personalized Differential Privacy [24]). Given a set of users $U = \{u_1, \dots, u_n\}$ with privacy requirements (preferences) $\boldsymbol{\epsilon} = \{\epsilon_1, \dots, \epsilon_n\}$, a randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{O}$ satisfies $\boldsymbol{\epsilon}$-personalized differential privacy (or $\boldsymbol{\epsilon}$-PDP), if for every pair of neighboring datasets $D, D' \subseteq \mathcal{D}$ with $D, D \overset{tp_i}{\sim} D'$ [24], and for all sets $O \subseteq \mathcal{O}$ of possible outputs, it holds that

$$\Pr[\mathcal{M}(D) \in O] \leq e^{\epsilon_i} \cdot \Pr[\mathcal{M}(D') \in O], \tag{1}$$

where tp_i is the tuple set associate with u_i.

Definition 8 ($\boldsymbol{w}$-Event $\boldsymbol{\mathcal{E}}$ Personalized Differential Privacy). Let $\mathcal{M}$ be a mechanism that takes a stream prefix of arbitrary size as input. Let $\mathcal{O}$ be the set of all possible outputs of $\mathcal{M}$. Let U be the set of all users. $\mathcal{M}$ is $\boldsymbol{w}$-Event $\boldsymbol{\mathcal{E}}$ Personalized Differential Privacy (or $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$-EPDP) if $\forall O \subseteq \mathcal{O}, \forall i \in [|U|]$ with $w_i \in \boldsymbol{w}$ and $\mathcal{E}_i \in \boldsymbol{\mathcal{E}}$ and $\forall S_t, S'_t$ satisfying $S_t \sim_{w_i} S'_t$, it holds that

$$\Pr[M(S_t) \in O] \leq e^{\mathcal{E}_i} \Pr[M(S'_t) \in O]. \tag{2}$$

When $\boldsymbol{w} = \mathbf{1}$, $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$-EPDP simplifies to $\boldsymbol{\mathcal{E}}$-PDP [24].

Definition 9 ($(\tau, \boldsymbol{w}_B, \boldsymbol{w}_F)$-Event $(\boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-Personalized Differential Privacy). Let $\mathcal{M}$ be a mechanism that takes a stream prefix of arbitrary size as input. Let $\mathcal{O}$ be the set of all possible outputs of $\mathcal{M}$. Let U be the set of all users. Then $\mathcal{M}$ is $(\tau, \boldsymbol{w}_B, \boldsymbol{w}_F)$-Event $(\boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-Personalized Differential Privacy (or $(\tau, \boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-EPDP) if $\forall O \subseteq \mathcal{O}, \forall i \in [|U|]$ with $(w_{B,i}, w_{F,i}, \mathcal{E}_{B,i}, \mathcal{E}_{F,i}) \in (\boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$ and $\forall S_t, S'_t$ satisfying $S_t \sim_{B,\tau,w_{B,i}} S'_t$ and $S_t \sim_{F,\tau,w_{F,i}} S'_t$, it holds that

$$\Pr[M(S_t) \in O] \leq e^{\mathcal{E}_{B,i}+\mathcal{E}_{F,i}} \Pr[M(S'_t) \in O].$$

Scope and Limitation of Protection. The proposed notions $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$-EPDP and $(\tau, \boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-EPDP protect user data contributions, rather than the privacy preferences themselves. In particular, the personalized budgets and window sizes are treated as mechanism parameters. These notions inherit the bounded-window semantics of classical w-event privacy: they protect each user's contribution only within the corresponding fixed or dynamic event window through the released outputs, but do not provide full trajectory-level privacy over an unbounded stream. Therefore, our goal is not to resolve the general trajectory-level limitation of w-event privacy, but to extend the w-event paradigm to heterogeneous and time-varying personalized privacy requirements. The trusted curator assumption is standard in centralized stream release; protecting against an untrusted curator is outside the scope of this paper and belongs to local/shuffled privacy settings.

3.3 Personalized Privacy Requirement

In this paper, we consider two kinds of personalized privacy requirements.

Fixed Personalized Privacy Requirement. For any user u_i, they have a privacy level requirement $\mathcal{E}_i$ within a specific window size w_i meaning the privacy level in this window should achieve $\mathcal{E}_i$-DP. We define w_i as the *fixed personalized window size requirement* and $\mathcal{E}_i$ as the *fixed personalized privacy budget requirement*. Together, the pair $(w_i, \mathcal{E}_i)$ constitutes the *fixed personalized privacy requirement*.

Dynamic Personalized Privacy Requirement. For any user u_i at time slot t, there are two privacy requirements: a dynamic backward requirement and a dynamic forward requirement. The backward requirement specifies a privacy level $\mathcal{E}_{B,i,t}$ within a dynamic backward window size $w_{B,i,t}$, with the window ending at time slot t to achieve $\mathcal{E}_{B,i,t}$-DP. The forward requirement specifies a privacy level $\mathcal{E}_{F,i,t}$ within a dynamic forward window size $w_{F,i,t}$, with the window beginning at time slot t to achieve $\mathcal{E}_{F,i,t}$-DP. These components constitute u_i's *backward window size requirement at time slot t*, *backward privacy budget requirement at time slot t*, *forward window size requirement at time slot t*, and *forward privacy budget requirement at time slot t*. Together, the pairs $(w_{B,i,t}, \mathcal{E}_{B,i,t})$ and $(w_{F,i,t}, \mathcal{E}_{F,i,t})$ form u_i's

backward privacy requirement at time slot t and *forward privacy requirement at time slot t*, respectively. Although the forward requirement can be equivalently rewritten as a shifted backward requirement, we keep the forward form because it directly captures the user's requirement declared at time slot t for the future window starting from t.

3.4 DPWEPP-IDS

Given a data stream S, the analyst aims to obtain the data stream count (i.e., original count) publishing as $\boldsymbol{c} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots]$. To protect user privacy, however, the analyst only receives the obfuscated data stream count (i.e., estimation count) $\boldsymbol{r} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots]$. The goal of the problem is to minimize the difference between the estimation count and the original count while meeting the personalized privacy requirement. We present our problem definition as follows.

Definition 10 (Dynamic Personalized w-Event Private Publishing for Infinite Data Streams). Given a user set $U = \{u_1, u_2, \ldots, u_n\}$ where each u_i holds a data collection $(w_{B,i,t}, \mathcal{E}_{B,i,t}, w_{F,i,t}, \mathcal{E}_{F,i,t}, \boldsymbol{x}_{i,t})$ at time slot t. All $\boldsymbol{x}_{i,t}$ for $u_i \in U$ at time slot t form D_t. All D_t consist an infinite data stream $D = [D_1, D_2, \ldots]$. Dynamic Personalized w-Event Private Publishing for Infinite Data Streams (or DPWEPP-IDS) is to release an obfuscated histogram $\boldsymbol{r} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots]$ of D in each timestamp t achieving $(t, \boldsymbol{w}_{B,t}, \boldsymbol{w}_{F,t}, \boldsymbol{\mathcal{E}}_{B,t}, \boldsymbol{\mathcal{E}}_{F,t})$-EPDP with the difference between $\boldsymbol{r}$ and $\boldsymbol{c}$ minimized where $\boldsymbol{w}_{B,t} = [w_{B,1,t}, \ldots, w_{B,n,t}]$, $\boldsymbol{\mathcal{E}}_{B,t} = [\mathcal{E}_{B,1,t}, \ldots, \mathcal{E}_{B,n,t}]$, $\boldsymbol{w}_{F,t} = [w_{F,1,t}, \ldots, w_{F,n,t}]$, $\boldsymbol{\mathcal{E}}_{F,t} = [\mathcal{E}_{F,1,t}, \ldots, \mathcal{E}_{F,n,t}]$. Namely,

$$\min_{\epsilon_\theta} \sum_{t\in[T]} \|\boldsymbol{r}_t - \boldsymbol{c}_t\|_2^2$$

$$s.t. \sum_{k=t-w_{B,i,t}+1}^{t} \epsilon_{i,k} \le \mathcal{E}_{B,i,t}, \quad \forall u_i \in U$$

$$\sum_{k=t}^{t+w_{F,i,t}-1} \epsilon_{i,k} \le \mathcal{E}_{F,i,t}, \quad \forall u_i \in U$$

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

4 Personalized Window Size Mechanism

In this section, we consider the fixed personalized setting, where each user maintains the same privacy requirement across all time slots. Although this setting is simpler than the dynamic case, it provides the core release components needed by our general framework. In particular, it allows us to introduce the basic mechanism for transforming heterogeneous budgets into a system-level release decision. We refer to this fixed problem as Dynamic Personalized w-Event Private Publishing for Infinite Data Streams (DPWEPP-IDS).

To maximize estimation accuracy at each time slot, we first analyze the reporting error and then introduce Optimal Budget Selection (OBS), a basic component for determining a release threshold under heterogeneous privacy budgets. Based on OBS, we develop the Personalized Window Size Mechanism (PWSM), which transforms heterogeneous personalized privacy requirements into a system-level release decision for the fixed personalized setting.

4.1 Problem Simplifying

Definition 11 (PWEPP-IDS). Given a user set $U = \{u_1, u_2, \ldots, u_n\}$, each u_i holds a privacy requirement $(w_i, \mathcal{E}_i)$ 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 $S = [D_1, D_2, \ldots]$. PWEPP-IDS is to publish an obfuscated histogram $\boldsymbol{r} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots]$ of S at each time slot t achieving $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$-EPDP with the distance between $\boldsymbol{r}$ and $\boldsymbol{c}$ minimized, namely $\forall T \in \mathbb{N}^+$:

$$\min_{\epsilon_\theta} \sum_{t\in[T]} \|\boldsymbol{r}_t - \boldsymbol{c}_t\|_2^2$$

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

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

Proposition 1 *The fixed personalized setting is a special case of the dynamic personalized setting when, for every user u_i and every time slot t, the privacy requirements remain constant over time. That is, $(w_{B,i,t}, \mathcal{E}_{B,i,t}, w_{F,i,t}, \mathcal{E}_{F,i,t})$ reduce to fixed user-specific parameters $(w_i, \mathcal{E}_i)$. Moreover, when all users share the same fixed privacy requirement, the model further reduces to the classical homogeneous w-event setting.*

Proof (sketch). When privacy requirements do not vary over time, the dynamic feasibility constraints become fixed personalized budget constraints, and DPWEPP-IDS reduces to PWEPP-IDS. If all users further share the same privacy budget and event window, the personalized constraints collapse into a single homogeneous constraint, which recovers the classical w-event setting.

4.2 Reporting Errors

Privacy budget allocation can be determined for any type of privacy requirement at each time slot. For time slot t with privacy budget allocation $\boldsymbol{\epsilon} = \{\epsilon_1, \ldots \epsilon_n\}$, we use the Sampling Mechanism (SM) [24] to satisfy all users' privacy requirements (i.e., achieving $\boldsymbol{\epsilon}$-PDP). SM operates in two

steps: *sample* (SM_s) and *disturb* (SM_d). In SM_s, the server sets a privacy budget threshold ϵ_θ and constructs a sampling subset D_S. Specifically, it adds items x_i with $\epsilon_i \geq \epsilon_\theta$ directly to D_S, while sampling other items x_i with $\epsilon_i < \epsilon_\theta$ at a probability of $p_i = \frac{e^{\epsilon_i}-1}{e^{\epsilon_\theta}-1}$. In SM_d, the server uses a DP mechanism (e.g., the Laplace Mechanism) to generate an obfuscated result that achieves ϵ_θ-DP.

SM introduces two types of errors: *sampling error* (err_s) and *noise error* (err_{dp}). Given a privacy budget threshold ϵ_θ, $err_s(\epsilon_\theta)$ occurs when sampling users with privacy budgets below ϵ_θ, while $err_{dp}(\epsilon_\theta)$ results from adding noise to achieve ϵ_θ-DP. The sum of these two errors constitutes the total reporting error. Next, we introduce these sampling and noise errors in detail.

Definition 12 (Sampling Error [24]). Given a privacy budget threshold ϵ_θ and m distinct privacy budgets $\tilde{\epsilon}_1, \tilde{\epsilon}_2, \ldots, \tilde{\epsilon}_m$ 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 $\left(\sum_{i=1}^{m} n_i = n\right)$, the sampling error $err_s(\epsilon_\theta)$ is defined as

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

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

Definition 13 (Noise Error [24]). The noise error $err_{dp}(\epsilon_\theta)$ is defined as the error of the Laplace mechanism, namely,

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

Various metrics exist to measure the errors of Laplace mechanisms for noise error, including variance [24, 34], scale [25, 19], and (α, β)-usefulness [19, 5]. In this work, we employ variance as our metric.

4.3 Optimal Budget Selection

Given the budget allocation $(\epsilon_{1,t}, \epsilon_{2,t}, \ldots, \epsilon_{n,t})$ of n users, we can determine the frequency of each distinct budget 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 first counts the distinct privacy budgets (Lines 1-2). It then finds the minimum reporting error err_{min} (lines 4-8). Specifically, it iterates through all distinct privacy budgets $\tilde{\epsilon}_k \in \tilde{\boldsymbol{\epsilon}}$ and identifies the value $\tilde{\epsilon}_k$ that produces the smallest total error $err = err_s(\tilde{\epsilon}_k) + err_{dp}(\tilde{\epsilon}_k)$. This value and its error are returned as the optimal privacy budget ϵ_{opt} and the minimum 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 errors: $err_1 = 0 + \frac{2}{0.1^2} = 200$, $err_2 = 2 \times \frac{e^{0.1}-1}{e^{0.4}-1} \times \left(1 - \frac{e^{0.1}-1}{e^{0.4}-1}\right) + \left(2 \times \left(1 - \frac{e^{0.1}-1}{e^{0.4}-1}\right)\right)^2 + \frac{2}{0.4^2} = 15.31$ and $err_3 = 2 \times \frac{e^{0.1}-1}{e^{0.8}-1} \times \left(1 - \frac{e^{0.1}-1}{e^{0.8}-1}\right) + 5 \times \frac{e^{0.4}-1}{e^{0.8}-1} \times \left(1 - \frac{e^{0.4}-1}{e^{0.8}-1}\right) + \left(2 \times \left(1 - \frac{e^{0.1}-1}{e^{0.8}-1}\right) + 5 \times \left(1 - \frac{e^{0.4}-1}{e^{0.8}-1}\right)\right)^2 + \frac{2}{0.8^2} = 89.74$. Finally, OBS returns 0.4 with the minimum error 15.31.

Algorithm 1: Optimal Budget Selection (OBS)

Input: personalized privacy budget list $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)$
Output: ϵ_{opt}, err_{min}
1 Extract distinct budget set $\tilde{\boldsymbol{\epsilon}} = (\tilde{\epsilon}_1, \tilde{\epsilon}_2, \ldots, \tilde{\epsilon}_{\tilde{n}})$ from $\boldsymbol{\epsilon}$;
2 Count the frequency n_k of all $\tilde{\epsilon}_k \in \tilde{\boldsymbol{\epsilon}}$;
3 Initialize err_{min} as the upper bound of error value;
4 **for** $\tilde{\epsilon}_k \in \tilde{\boldsymbol{\epsilon}}$ **do**
5 $err \leftarrow err_s(\tilde{\epsilon}_k) + err_{dp}(\tilde{\epsilon}_k)$;
6 **if** $err < err_{min}$ **then**
7 $err_{min} \leftarrow err$;
8 $\epsilon_{opt} \leftarrow \tilde{\epsilon}_k$;
9 **return** ϵ_{opt}, err_{min}

4.4 Personalized Window Size Mechanism

In real applications, to realize personalized privacy protection, the system need to get the collect users' privacy requirements. To accomplish this, 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 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. Once users submit these selections, the server converts them into the corresponding values.

After receiving all users' privacy requirements, the system must determine how to allocate privacy budgets within each user's feasible window while maximizing estimation accuracy. Classical budget-division methods [25, 34] are designed for homogeneous w-event privacy, where all users share the same privacy budget and the same event window. In our setting, however, users may specify different window sizes and privacy budgets, whereas the system still publishes one shared aggregate result at each time slot.

Therefore, the key issue is no longer how to allocate a single global privacy budget sequence, but how to transform heterogeneous feasible budgets into a valid system-level release decision. To address this challenge, we propose the Personalized Window Size Mechanism (PWSM). The core

idea is to use OBS together with the sampling mechanism to convert heterogeneous user-specific privacy budgets into a common release threshold, and then use that threshold to perform personalized dissimilarity estimation and adaptive release.

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 publication $\boldsymbol{r}_l$, namely,

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

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:

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

where Lap denotes the Laplace noise in the Laplace mechanism [19].

Based on the above observation, we introduce PWSM as a framework for unifying heterogeneous personalized privacy budgets into a shared release decision. As shown in Algorithm 2, the PWSM algorithm takes the historical estimation *His* and the current personalized privacy requirement $(\boldsymbol{w}_t, \boldsymbol{\mathcal{E}}_t)$ as the input. PWSM first calculates all users' budget allocations $\boldsymbol{\epsilon}_t$ at the current time slot t on the premise of satisfying $(\boldsymbol{w}_t, \boldsymbol{\mathcal{E}}_t)$-EPDP (line 1). It then divides $\boldsymbol{\epsilon}_t$ into two parts: calculation budget $\boldsymbol{\epsilon}_t^{(1)}$ and publication budget $\boldsymbol{\epsilon}_t^{(2)}$ (line 2). Using $\boldsymbol{\epsilon}_t^{(1)}$, PWSM calculates the personalized private dissimilarity dis between the current count value and the last reported one (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).

Next, we present two methods based on PWSM: Personalized Budget Distribution (PBD) and Personalized Budget Absorption (PBA), each designed to handle different types of data streams.

Algorithm 2: Personalized Window Size Mechanism (PWSM)

Input: historical estimation *His*, privacy requirement $(\boldsymbol{w}_t, \boldsymbol{\mathcal{E}}_t)$ at time slot t
Output: $\boldsymbol{r}$

1 Calculate the current privacy budgets $\boldsymbol{\epsilon}_t$ of all users according to $\boldsymbol{\mathcal{E}}_t$ and $\boldsymbol{w}_t$;
2 Split $\boldsymbol{\epsilon}_t$ into two components: the calculation budget $\boldsymbol{\epsilon}_t^{(1)}$ and the publication budget $\boldsymbol{\epsilon}_t^{(2)}$ satisfying $\boldsymbol{\epsilon}_t = \boldsymbol{\epsilon}_t^{(1)} + \boldsymbol{\epsilon}_t^{(2)}$;
3 Calculate dissimilarity dis between current estimation and the last estimation by $\text{SM}(\boldsymbol{\epsilon}_t^{(1)})$;
4 Calculate the reporting error err of current estimation by $\text{OBS}\left(\boldsymbol{\epsilon}_t^{(2)}\right)$;
5 **if** dis $> \sqrt{\text{err}}$ **then**
6 Calculate current estimation $\boldsymbol{r}$ by $\text{SM}\left(\boldsymbol{\epsilon}_t^{(2)}\right)$;
7 **else**
8 Set current estimation $\boldsymbol{r}$ as the last reporting value;
9 **return** $\boldsymbol{r}$.

4.5 Personalized Budget Distribution and Personalized Budget Absorption

Basic notations. Before describing our personalized methods, we first need to declare some important notations specific to these methods.

Definition 14 (Null/Non-null Publication). Given a sequence of publications $(r_1, r_2, \ldots, r_t)$, a *null publication* refers to approximating a historical value without consuming any privacy budget in Part_{NOP}, while a *non-null publication* represents a new publication that consumes privacy budget in Part_{NOP}.

For any time slot $2 \le \tau \le t$, we refer to $r_{\tau-1}$ as the last reporting value (or last publication) of time slot τ. In the sequence $(r_1, r_2, \ldots, r_\tau)$, we define the most recent non-null publication r_l where $l < \tau$ as *the last non-null publication*.

For example in Figure 2, the publications at time slots $\tau, \tau+1, \tau+4$ are non-null publications, while those at $\tau+2$ and $\tau+3$ are null publications. The last non-null publication at time slot $\tau+4$ is the publication at time slot $\tau+1$.

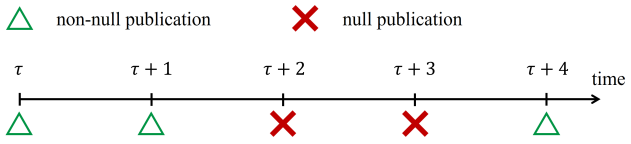


Fig. 2: A null/non-null publication example.

Definition 15 (Skipped/Nullified Publication). The skipped publications are those null publications with $dis \le \sqrt{err}$. Given a privacy budget requirement $\mathcal{E}$ and a window size w, a budget share $\bar{\epsilon} = \mathcal{E}/w$ is defined as the average privacy budget per time slot. When publishing new obfuscated data consumes x budget shares ($x > 1$), in order to maintain the average value, the following $x-1$ time slots values are approximated by the last publications. These $x-1$ time slots are defined as nullified time slots.

We can see both skipped and nullified publications are non publications. Figure 3 illustrates an example for skipped and nullified publications. With a privacy budget $\mathcal{E}$ of 4 and

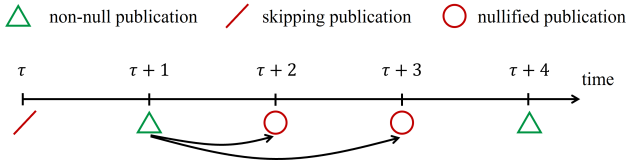


Fig. 3: A skipped/nullified publication example.

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 $\epsilon_{opt} \leftarrow \text{OBS}(\boldsymbol{\epsilon}_t)$;
2 $\tilde{D}_t \leftarrow \text{SM}_s(D_t, \boldsymbol{\epsilon}_t, \epsilon_{opt})$;
3 $\tilde{\boldsymbol{c}}_t \leftarrow Q(\tilde{D}_t)$;
4 Get the last non-null publication $\boldsymbol{r}_l$ from $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$;
5 **return** dis $\leftarrow \frac{1}{d}\sum_{j=1}^{d} |\tilde{\boldsymbol{c}}_t[j] - \boldsymbol{r}_l[j]| + Lap(1/(d \cdot \epsilon_{opt}))$;

Algorithm 4: Personalized Budget Distribution (PBD)

Input: D_t, privacy requirement set $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$, historical data publication $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$
Output: $\boldsymbol{r}_t$

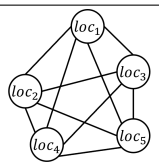
1 Calculate the current window average budget $\bar{\epsilon}_i \leftarrow \mathcal{E}_i/w_i$ for each $i \in [n]$;
2 Set $\boldsymbol{\epsilon}_t^{(1)} \leftarrow (\bar{\epsilon}_1/2, \bar{\epsilon}_2/2, \ldots, \bar{\epsilon}_n/2)$;
3 $dis \leftarrow \text{DC}\left(D_t, \boldsymbol{\epsilon}_t^{(1)}, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1}\right)$ by **Algorithm 3**;
4 Set $\epsilon_{rm,i} \leftarrow \mathcal{E}_i/2 - \sum_{k=t-w_i+1}^{t-1} \epsilon_{i,k}^{(2)}$ for each $i \in [n]$;
5 $\boldsymbol{\epsilon}_t^{(2)} \leftarrow (\epsilon_{rm,1}/2, \epsilon_{rm,2}/2, \ldots, \epsilon_{rm,n}/2)$;
6 $\epsilon_{opt}^{(2)}, err_{opt}^{(2)} \leftarrow \text{OBS}\left(\boldsymbol{\epsilon}_t^{(2)}\right)$ by **Algorithm 1**;
7 **if** dis $> \sqrt{\text{err}_{opt}^{(2)}}$ **then**
8 $\quad \tilde{D}_t^{(2)} \leftarrow \text{SM}_s\left(D_t, \boldsymbol{\epsilon}_t^{(2)}, \epsilon_{opt}^{(2)}\right)$;
9 $\quad \tilde{\boldsymbol{c}}_t^{(2)} \leftarrow Q\left(\tilde{D}_t^{(2)}\right)$;
10 $\quad$ **return** $\boldsymbol{r}_t \leftarrow SM_d\left(\tilde{\boldsymbol{c}}_t^{(2)}, \epsilon_{opt}^{(2)}\right)$;
11 **else**
12 $\quad \boldsymbol{\epsilon}_t^{(2)} \leftarrow (0, 0, \ldots, 0)$;
13 $\quad$ **return** $\boldsymbol{r}_t \leftarrow \boldsymbol{r}_{t-1}$;

a window size of 4, the budget share $\bar{\epsilon}$ equals $\mathcal{E}/w = 1$. When time slot $\tau + 1$ uses 3 shares, the publications at time slots $\tau + 2$ and $\tau + 3$ become nullified publications.

Personalized Budget Distribution (PBD). As shown in Algorithm 4, PBD inputs the current user data, all users' fixed privacy requirements, and historical data publication. The fixed privacy budget requirement $\mathcal{E}_i$ of u_i is split into two parts: 1) Part_{DC} for calculating the dissimilarity between the current data and the last publication (Lines 2-3); 2) Part_{NOP} for calculating the new obfuscated publication at the current time slot (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., $\frac{\mathcal{E}_i}{2w_i}$ 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 [24] to compute the dissimilarity *dis* (Lines 2-5). Notice that the remaining budget calculation in Line 4 is a standard sliding-window sum and can be maintained incrementally.

	u_1	u_2	u_3
$\boldsymbol{\mathcal{E}}$	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$
$\boldsymbol{w}$	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

Fig. 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 publication instead.

Example 3 Suppose there are 3 users distributed across 5 locations, forming a complete graph. Figure 4 illustrates the fixed personalized privacy 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 $\mathcal{E}_i/2$. The first part is allocated for dissimilarity calculation, while the second is for publication noise calculation. For instance, $\mathcal{E}_1$ is divided into $\boldsymbol{\epsilon}_1^{(1)}(u_1) = \mathcal{E}_1/2$ and $\boldsymbol{\epsilon}_1^{(2)}(u_1) = \mathcal{E}_1/2$. We compute the privacy budget usage $\epsilon_{i,t}^{(1)}$ for dissimilarity and $\epsilon_{i,t}^{(2)}$ for obfuscated 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, $\epsilon_{1,t}^{(1)} = \boldsymbol{\epsilon}_1^{(1)}(u_1)/w_1 = \mathcal{E}_1/8$. At time slot 1, $\epsilon_{1,1}^{(2)} = \boldsymbol{\epsilon}_1^{(2)}(u_1)/2 = \mathcal{E}_1/4$. The algorithm calculates the dissimilarity *dis* at time slot 1 using all $\epsilon_{i,1}^{(1)}$, and the error $err_{opt}^{(2)}$ using all $\epsilon_{i,1}^{(2)}$. 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

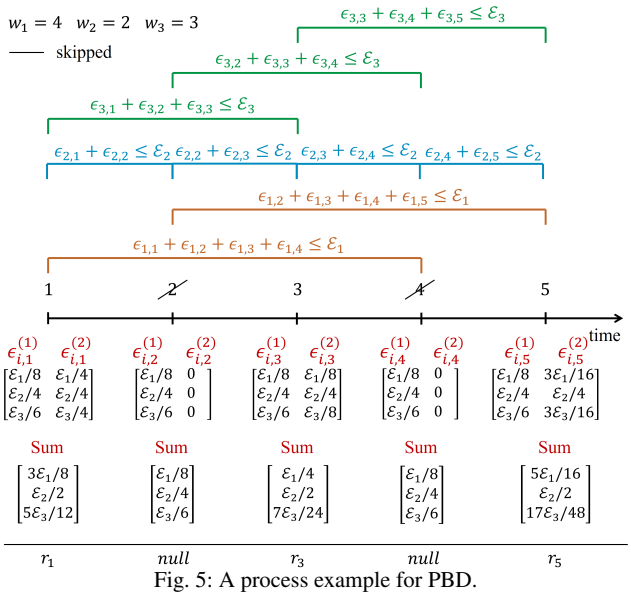


Fig. 5: A process example for PBD.

for all users. At time slot 3, $\epsilon_{1,3}^{(2)} = (\mathcal{E}_1/2 - \epsilon_{1,1}^{(2)})/2 = \mathcal{E}_1/8$. 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 $\epsilon_{1,1}^{(1)} + \epsilon_{1,1}^{(2)} = 3\mathcal{E}_1/8$.

Personalized Budget Absorption (PBA). Algorithm 5 outlines the process of PBA. The dissimilarity calculation (Part_{DC}) in PBA is identical to that of PBD. However, PBA and PBD differ significantly in their strategies on allocating the publication privacy budget (Part_{NOP}).

For Part_{NOP} in PBA, we assume an average privacy budget of $\frac{\mathcal{E}_i}{2w_i}$ (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 6 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 ($t - l \le \tilde{t}_N$). If so, we approximate the publication using the last publication. Otherwise, we calculate the remaining budget shares from the precursor time slots (i.e., $t_{A,i}$ in Line 12) and set the current publication budget as the total absorbed shares (Line 13). 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., $\mathcal{E}_i/(2w_i)$) to publish a new obfuscated statistic result. Assume time slot 2 is skipped $\left(\text{i.e., } dis \le \sqrt{err_{opt}^{(2)}}\right)$. At time slot 3, $t_{1,N} = t_{2,N} = t_{3,N} = 0$. Thus, the nullified bound $\tilde{t}_N$ is 0. Since $t - l = 3 - 1 = 2 > \tilde{t}_N$, a new obfuscated statistic result is reported. The publication budget set

Algorithm 5: Personalized Budget Absorption (PBA)

Input: D_t, fixed personalized privacy requirement set $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$, historical data publication $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$
Output: $\boldsymbol{r}_t$

1 Calculate the current window average budget $\bar{\epsilon}_i = \mathcal{E}_i/w_i$ for each $i \in [n]$;
2 $\boldsymbol{\epsilon}_t^{(1)} \leftarrow (\bar{\epsilon}_1/2, \bar{\epsilon}_2/2, \ldots, \bar{\epsilon}_n/2)$;
3 $dis \leftarrow \text{DC}\left(D_t, \boldsymbol{\epsilon}_t^{(1)}, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1}\right)$ by **Algorithm 3**;
4 **for** $i \in [n]$ **do**
5 Initialize nullified time slots $t_{i,N}$ as 0;
6 Set $t_{i,N} \leftarrow \frac{\epsilon_{i,l}^{(2)}}{\mathcal{E}_i/(2w_i)} - 1$ if l exists where l is the last non-null publication time slot;
7 Set nullified time slot bound $\tilde{t}_N \leftarrow \max_{i \in [n]} t_{i,N}$;
8 **if** $t - l \le \tilde{t}_N$ **then**
9 **return** $\boldsymbol{r}_t \leftarrow \boldsymbol{r}_{t-1}$;
10 **else**
11 **for** $i \in [n]$ **do**
12 Set absorbed time slots $t_{A,i} \leftarrow \max(t - l - t_{i,N}, 0)$;
13 Set publication budget $\epsilon_{i,t}^{(2)} \leftarrow \frac{\mathcal{E}_i}{2w_i} \cdot \min(t_{A,i}, w_i)$;
14 $\boldsymbol{\epsilon}_t^{(2)} \leftarrow \left(\epsilon_{1,t}^{(2)}, \epsilon_{2,t}^{(2)}, \ldots, \epsilon_{n,t}^{(2)}\right)$;
15 $\epsilon_{opt}^{(2)}, err_{opt}^{(2)} \leftarrow \text{OBS}\left(\boldsymbol{\epsilon}_t^{(2)}\right)$;
16 **if** $\text{dis} > \sqrt{\text{err}_{opt}^{(2)}}$ **then**
17 $\tilde{D}_t^{(2)} \leftarrow \text{SM}_s\left(D_t, \boldsymbol{\epsilon}_t^{(2)}, \epsilon_{opt}^{(2)}\right)$;
18 $\tilde{\boldsymbol{c}}_t^{(2)} \leftarrow Q\left(\tilde{D}_t^{(2)}\right)$;
19 **return** $\boldsymbol{r}_t \leftarrow SM_d\left(\tilde{\boldsymbol{c}}_t^{(2)}, \epsilon_{opt}^{(2)}\right)$;
20 **else**
21 $\boldsymbol{\epsilon}_t^{(2)} \leftarrow (0, 0, \ldots, 0)$;
22 **return** $\boldsymbol{r}_t \leftarrow \boldsymbol{r}_{t-1}$;

is calculated as $\boldsymbol{\epsilon}_3^{(2)} = (\mathcal{E}_1/4, \mathcal{E}_2/2, \mathcal{E}_3/3)$. At time slot 4, $t_{1,N} = t_{2,N} = t_{3,N} = 1$. As $t - l = 4 - 3 = 1 \le \tilde{t}_N$, the publication is approximated as the one at time slot 3. 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 publication budget set is $\boldsymbol{\epsilon}_5^{(2)} = (\mathcal{E}_1/8, \mathcal{E}_2/4, \mathcal{E}_3/6)$.

4.6 Analyses

Time Cost Analysis. Let m be the number of distinct privacy requirements $(w_i, \mathcal{E}_i)$, where $m \le n$. Then we have Theorem 1 as follows.

Theorem 1 *The time complexities of PBD and PBA are both $O(n)$.*

Proof 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 $O(n)$.

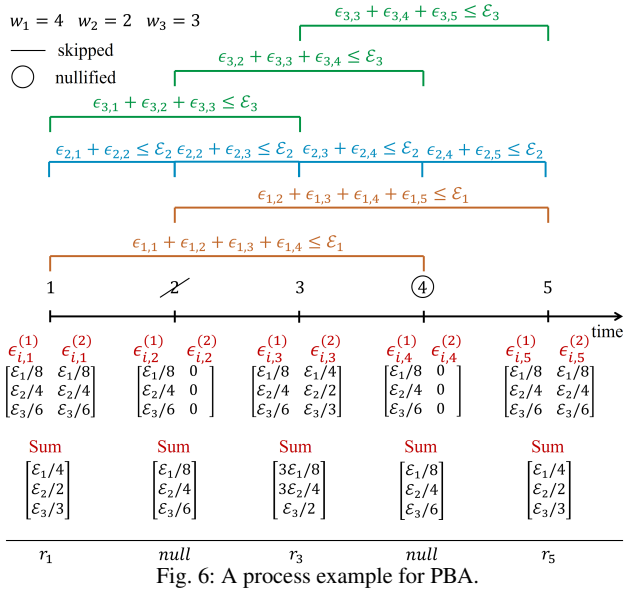


Fig. 6: A process example for PBA.

Memory Complexity Analysis. For PBD and PBA, we have Theorem 2 as follows.

Theorem 2 *Both PBD and PBA have memory complexity* $O(n \cdot w_{max})$.

Proof For the process of OBS, the memory complexity is $O(m)$. For each one of the n users in both PBD and PBA, each user requires storing at most w_{max} window-related states. Thus, the memory complexity is $O(n \cdot w_{max})$.

Scalability Discussion. The above time and memory bounds indicate that PBD and PBA are scalable to large user populations. At each time slot, the computation only requires processing the current per-user privacy requirements and maintaining a limited amount of historical state within the relevant window(s), rather than revisiting the entire stream history. Therefore, the per-time-slot cost grows linearly with the number of users n, while the memory usage is bounded by the maintained budget/publication states associated with active windows. This makes the method practical for long-running streams with a large user population, provided that the maximum window size remains moderate.

Privacy Analysis. As for the privacy analysis of PBD and PBA, we have Theorem 3 as follows.

Theorem 3 *PBD and PBA satisfy* $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$*-EPDP.*

Proof Please refer to details of Theorem 3 in Appendix 8.4.1.

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 $(w_i, \mathcal{E}_i)$, let $\epsilon_L = \min_{i\in[n]} \frac{\mathcal{E}_i}{w_i}$ and $\epsilon_R = \max_{i\in[n]} \frac{\mathcal{E}_i}{w_i}$ be the minimum and maximum values of $\frac{\mathcal{E}_i}{w_i}$, respectively. Let n_A be the number of times ϵ_R appears among all users. We assume that at most $\tilde{s} \le w_L$ non-null 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 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:*

$$\min\left(\frac{2I^2}{\min_j \epsilon_j^2}, (n - n_A)\left(n - n_A + \frac{1}{4}\right) + \frac{2I^2}{\max_j \epsilon_j^2}\right), \tag{3}$$

where $n_A = n_k$ *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 $2 \cdot \left(\frac{I}{\min_j \epsilon_j}\right)^2 = \frac{2I^2}{\min_j \epsilon_j^2}$. 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:

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

The noise error err_{dp} in this case is $2 \cdot \left(\frac{I}{\max_j \epsilon_j}\right)^2 = \frac{2I^2}{\max_j \epsilon_j^2}$. Thus, the total error of M_R is $err_{M_R} = (n - n_M)\left(n - n_M + \frac{1}{4}\right) + \frac{2I^2}{\max_j \epsilon_j^2}$. 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,

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

To ensure the robustness of Lemma 1, we analyze the behavior of the proposed mechanism under extreme conditions. Specifically, we consider two extreme cases: (1) *Uniform Privacy Budget*. When all users possess the same privacy budget, i.e., $\epsilon_j \equiv \epsilon$, the system reduces to a single budget-quantity pair (ϵ, n). Here, the error equals the standard CDP bound, i.e., $2I^2/\epsilon^2$. Given $n_M = n$, Equation (3) evaluates to $\min\left(\frac{2I^2}{\epsilon^2}, 0 + \frac{2I^2}{\epsilon^2}\right) = \frac{2I^2}{\epsilon^2}$. Thus, the equation consistently recovers the standard CDP error in the uniform setting; (2) *Highly Disparate Privacy Budgets*. When

some users have significantly larger privacy budget than others (e.g., $\epsilon_1 \ll \epsilon_2 \ll \cdots \ll \epsilon_n$), the error is no more than $2I^2/\min_j \epsilon_j^2$, which is exactly the dominant error term err_{M_L} in Equation (3). Thus, Lemma 1 still holds in these two extreme cases.

For PBD we present Theorem 4 to declare its error upper bound as follows.

Theorem 4 *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* $Z=(n-n_A)\left(n-n_A+\frac{1}{4}\right)$*, if at most* $\tilde{s}$ *non-null publications occur in any window with size* w_L*.*

Proof Please refer to details of Theorem 4 in Appendix 8.5.1.

PBD achieves low error when the number of non-null 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 budgets 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 set as the last non-null 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 5 as follows.

Theorem 5 *The average error per time slot in PBA is at most* $\min\left(\frac{8}{d^2\epsilon_L}, Z+\frac{8}{d^2\epsilon_R}\right)+\frac{1}{2\alpha+1}\left(\widetilde{\text{err}}_{\text{NOP}}^{(s,p)}+\alpha\cdot\overline{\text{err}}_{nlf}\right)$ *where* $\widetilde{\text{err}}_{\text{NOP}}^{(s,p)}$ *is* $\min\left(\frac{2}{\epsilon_L^2}H_{\alpha+1}^2, (\alpha+1)Z+\frac{2}{\epsilon_R^2}H_{\alpha+1}^2\right)$ *when* $\alpha \le w_L$ *and* $\min\left(\frac{2}{\epsilon_L^2}H_{w_L}^2, w_L Z+\frac{2}{\epsilon_R^2}H_{w_L}^2\right)+(\alpha-w_L+1)\min\left(\frac{2}{\epsilon_{\tilde{L}}^2}, Z+\frac{2}{\epsilon_{\tilde{R}}^2}\right)$ *when* $\alpha > w_L$ *and* $Z=(n-n_A)\left(n-n_A+\frac{1}{4}\right)$ *and* H_x^2 *is the* x*-th square harmonic number, if there are* α *skipped publications occur in average before each publication.*

Proof Please refer to details of Theorem 5 in Appendix 8.5.2.

Discussion on Frequent-Update Regimes. The bound above shows that the utility of PBA may deteriorate when the number of skipped and nullified publications becomes large. This does not weaken the formal privacy guarantee of PBA, which is still ensured by the budget-composition analysis in Theorem 3, but it may reduce estimation accuracy when the stream changes too frequently. Therefore, PBA is more suitable for relatively smooth streams, whereas PBD is preferable when the stream exhibits persistent rapid changes.

5 Dynamic Personalized Window Size Mechanism

In this section, we consider the dynamic personalized setting, where each user may specify different backward and forward privacy requirements at different time slots. Unlike the fixed setting, the privacy budget allocation at the current time slot cannot be determined solely from the current requirement. Instead, it must remain consistent with previously consumed privacy budgets and, at the same time, preserve feasibility for future privacy requirements. Therefore, the dynamic setting introduces an online feasibility problem under heterogeneous time-varying personalized privacy requirements.

To address this problem, we generalize the fixed-setting framework to a dynamic framework called Dynamic Personalized Window Size Mechanism (DPWSM). DPWSM focuses on online feasibility maintenance and shared release-budget construction under dynamic personalized privacy requirements, rather than merely substituting personalized parameters into existing w-event mechanisms. The key idea of DPWSM is to compute feasible privacy budget upper bounds at each time slot under both backward and forward privacy requirements, and then use these feasible budgets to make a shared release decision for the current time slot.

5.1 Feasibility Conditions for Privacy Budget Requirements

Backward Feasibility Condition. *At time slot* t*, user* u_i *may declare a desired backward privacy requirement* $(w_{B,i,t}, \mathcal{E}_{B,i,t})$*. Since the historical privacy budget consumption* $\{\epsilon_{i,k}\}_{k<t}$ *is maintained internally by the trusted curator, the feasibility of this declared backward budget is checked by the curator rather than by the user. A backward requirement is feasible only if*

$$0 \le \epsilon_{i,t} \le \mathcal{E}_{B,i,t} - \sum_{k=\max(t-w_{B,i,t}+1,1)}^{t-1} \epsilon_{i,k} \quad \text{for} \quad i \in [n].$$

If the declared backward budget is infeasible, the curator either rejects it or projects it to the minimal feasible value before computing the current release budget.

The backward feasibility condition has two parts. The first rule requires users to propose valid backward privacy budgets that exceed their historical privacy usage within the current backward window. We assume the trusted curator enforces this feasibility condition before budget allocation at each time slot. The second rule establishes an upper bound for each user's privacy budget usage at the current time slot.

Forward Feasibility Condition. *Let* $T_{B,i,t} = \{\tau | \tau \leq t \leq \tau + w_{F,i,\tau} - 1\}$ *be the set of backward time slots whose forward windows cover time slot t. The privacy budget usage* $\epsilon_{i,t}$ *must not exceed the minimal remaining forward privacy budget among all the time slots in* $T_{B,i,t}$*. Namely,*

$$0 \leq \epsilon_{i,t} \leq \min_{\tau \in T_{B,i,t}} \left(\mathcal{E}_{F,i,\tau} - \sum_{k=\tau}^{t-1} \epsilon_{i,k} \right) \quad \text{for} \quad i \in [n].$$

The forward feasibility condition ensures that the privacy budget usage at the current time slot does not violate the forward privacy requirements of all historical time slots. We illustrate this forward feasibility condition in Example 5.

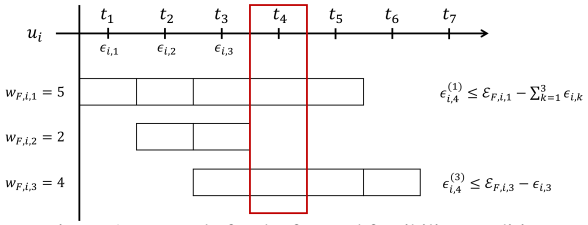


Fig. 7: An example for the forward feasibility condition.

Example 5 As shown in Figure 7, assume u_i spends privacy budgets $\epsilon_{i,1}$, $\epsilon_{i,2}$ and $\epsilon_{i,3}$ at time slots t_1, t_2 and t_3, respectively. The forward window sizes of u_i at these time slots are 5, 2 and 4. Let $\epsilon^{(k)}_{i,j}$ represent u_i's upper bound of the budget usage at time slot t_j constrained by the forward budget requirement at t_k. Based on the requirement at t_1, for the budget usage at t_4, we have $\epsilon^{(1)}_{i,4} \leq \mathcal{E}_{F,i,1} - \sum_{k=1}^{3} \epsilon_{i,k}$. Since the forward window at t_2 does not cover t_4, its requirement does not affect $\epsilon_{i,4}$. Based on the requirement at t_3, for the budget usage at t_4, we have $\epsilon^{(3)}_{i,4} \leq \mathcal{E}_{F,i,3} - \sum_{k=3}^{3} \epsilon_{i,k} = \mathcal{E}_{F,i,3} - \epsilon_{i,3}$. Therefore, the final budget upper bound is $\epsilon_{i,4} = \min\left(\epsilon^{(1)}_{i,4}, \epsilon^{(3)}_{i,4}\right)$.

5.2 Solution for DPWEPP-IDS

In this subsection, we instantiate DPWSM with two mechanisms: Dynamic Personalized Budget Distribution (DPBD) and Dynamic Personalized Budget Absorption (DPBA). Both mechanisms follow the same feasibility principle: the privacy budget used at each time slot must satisfy both backward and forward personalized privacy requirements. They differ in how the feasible publication budget is scheduled across time slots: DPBD follows a distribution-based strategy, whereas DPBA follows an absorption-based strategy.

Dynamic Personalized Budget Distribution. DPBD extends PBD to satisfy users' variable privacy level demands at different time slots. Similar to PBD, the process in DPBD is divided into Part_{DC} for dissimilarity calculation and Part_{NOP} for publication calculation. To satisfy both backward and forward feasibility conditions, the privacy budget usages in Part_{DC} and Part_{NOP} are required to comply with these feasibility conditions using half of the backward and forward privacy budgets. Specifically, for Part_{DC}: $\epsilon^{(1)}_{i,t} \leq \mathcal{E}_{B,i,t}/2 - \sum_{k=\max(t-w_{B,i,t}+1,1)}^{t-1} \epsilon^{(1)}_{i,k}$ (backward feasibility condition), and $\epsilon^{(1)}_{i,t} \leq \min_{\tau \in T_{B,i,t}} \left(\mathcal{E}_{F,i,\tau}/2 - \sum_{k=\tau}^{t-1} \epsilon^{(1)}_{i,k} \right)$ (forward feasibility condition). For Part_{NOP}: $\epsilon^{(2)}_{i,t} \leq \mathcal{E}_{B,i,t}/2 - \sum_{k=\max(t-w_{B,i,t}+1,1)}^{t-1} \epsilon^{(2)}_{i,k}$ (backward feasibility condition), and $\epsilon^{(2)}_{i,t} \leq \min_{\tau \in T_{B,i,t}} \left(\mathcal{E}_{F,i,\tau}/2 - \sum_{k=\tau}^{t-1} \epsilon^{(2)}_{i,k} \right)$ (forward feasibility condition).

The process of DPBD is shown in Algorithm 6. For each user u_i, we obtain the historical time slot set $T_{B,i,t}$ where each elmement's forward window covers the current time slot (Line 2).

In Part_{DC} process, for each historical time slot τ, the total forward privacy budget is set as $\mathcal{E}_{F,i,\tau}/2$ and allocated evently among all the time slot within the forward window. Thus each time slot in the forward window with size $w_{F,i,\tau}$ holds forward privacy budget as $\mathcal{E}_{F,i,\tau}/(2w_{F,i,\tau})$. Therefore, the privacy budget upper bound based on the forward feasibility condition at time slot t is $\min_{\tau \in T_{B,i,t}} \frac{\mathcal{E}_{F,i,\tau}}{2w_{F,i,\tau}}$ (Line 3). Based on the backward feasibility condition, the privacy budget upper bound at time slot t is set as $\mathcal{E}_{B,i,t}/2 - \sum_{\tau=t-w_{B,i,t}+1}^{t-1} \epsilon^{(1)}_{i,\tau}$ (Line 4). To satisfy both feasibility conditions, each user u_i's budget usage for Part_{DC} is set to the minimum of these two upper bounds: $\epsilon^{(1)}_{i,t} = \min\left(\epsilon^{(1)}_{F,i,t}, \epsilon^{(1)}_{B,i,t}\right)$ (Line 5). The subsequent steps in Part_{DC} follow Algorithm 4 (PBD).

In Part_{NOP} process, the forward remaining budget $\epsilon^{(2)}_{F,i,t}$ for each u_i is set to half of the minimum forward remaining budgets across all time slots in $T_{B,i,t}$ (Line 9). The backward remaining budget $\epsilon^{(2)}_{B,i,t}$ for each u_i is set to the remaining budget within the front $w_{B,i,t}$ time slots. The final publication budget for each u_i is determined by taking the minimum value between $\epsilon^{(2)}_{F,i,t}$ and $\epsilon^{(2)}_{B,i,t}$. The subsequent steps in Part_{NOP} follow those in Algorithm 4 (PBD).

Example 6 Consider a system with users' privacy requirements shown in Table 3. Privacy requirements are denoted as $B : (a, b)$ and $F : (c, d)$, where a is the backward window size $w_{B,i,t}$, b is the backward privacy budget $\mathcal{E}_{B,i,t}$, c is the forward window size $w_{F,i,t}$, and d is the forward privacy budget $\mathcal{E}_{F,i,t}$. We analyze the first 5 time slots with privacy settings shown in Figure 8. The status is recorded as $\left[\epsilon^{(1)}_{B,i,t}, \epsilon^{(1)}_{F,i,t}; \epsilon^{(2)}_{B,i,t}, \epsilon^{(2)}_{F,i,t}\right]$ with non-null publications occuring at time slots $t = 1$ and $t = 3$. At each time slot, we first compute the calculation budgets by Lines 2-5 of Algorithm 6, and then compute the publication budgets by

Algorithm 6: Dynamic Personalized Budget Distribution (DPBD)

Input: D_t, dynamic personalized privacy requirement set $(\boldsymbol{w}_{B,t}, \boldsymbol{\mathcal{E}}_{B,t}, \boldsymbol{w}_{F,t}, \boldsymbol{\mathcal{E}}_{F,t})$, historical data publication $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$

Output: $\boldsymbol{r}_t$

1 **for** $i \in [n]$ **do**
2 Calculate $T_{B,i,t} \leftarrow \{\tau | \tau \le t \le \tau + w_{F,i,\tau} - 1\}$;
3 Calculate $\epsilon^{(1)}_{F,i,t} \leftarrow \min_{\tau \in T_{B,i,t}} \frac{\mathcal{E}_{F,i,\tau}}{2w_{F,i,\tau}}$;
4 Calcualte $\epsilon^{(1)}_{B,i,t} \leftarrow \mathcal{E}_{B,i,t}/2 - \sum_{\tau=t-w_{B,i,t}+1}^{t-1} \epsilon^{(1)}_{i,\tau}$;
5 Set $\epsilon^{(1)}_{i,t} \leftarrow \min\left(\epsilon^{(1)}_{F,i,t}, \epsilon^{(1)}_{B,i,t}\right)$;
6 $\boldsymbol{\epsilon}^{(1)}_t \leftarrow \left(\epsilon^{(1)}_{1,t}, \epsilon^{(1)}_{2,t}, \ldots, \epsilon^{(1)}_{n,t}\right)$;
7 Estimate $dis \leftarrow \mathrm{DC}\left(D_t, \boldsymbol{\epsilon}^{(1)}_t, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1}\right)$ by **Algorithm 3**;
8 **for** $i \in [n]$ **do**
9 Calculate $\epsilon^{(2)}_{F,i,t} \leftarrow \frac{1}{2}\min_{\tau \in T_{B,i,t}}\left(\mathcal{E}_{F,i,\tau}/2 - \sum_{k=\tau}^{t-1} \epsilon^{(2)}_{i,k}\right)$;
10 Calculate $\epsilon^{(2)}_{B,i,t} \leftarrow \left(\mathcal{E}_{B,i,t}/2 - \sum_{\tau=t-w_{B,i,t}+1}^{t-1} \epsilon^{(2)}_{i,\tau}\right)$;
11 Set $\epsilon^{(2)}_{i,t} \leftarrow \min\left(\epsilon^{(2)}_{F,i,t}, \epsilon^{(2)}_{B,i,t}\right)$;
12 $\boldsymbol{\epsilon}^{(2)}_t \leftarrow \left(\epsilon^{(2)}_{1,t}, \epsilon^{(2)}_{2,t}, \ldots, \epsilon^{(2)}_{n,t}\right)$;
13 Same as Lines 6-13 in **Algorithm 4**

Table 3: Privacy requirements in DPBD, where B and F denote backward privacy requirements and forward privacy budget requirements.

Time	1	2	3	4	5
u_1	$B:(1,1.0)$ $F:(4,2.4)$	$B:(2,2.4)$ $F:(4,3.2)$	$B:(2,2.8)$ $F:(3,4.2)$	$B:(2,2.4)$ $F:(3,2.4)$	$B:(5,3.0)$ $F:(2,0.8)$
u_2	$B:(1,0.6)$ $F:(2,1.6)$	$B:(2,1.6)$ $F:(2,2.4)$	$B:(2,3.2)$ $F:(2,2.8)$	$B:(3,4.2)$ $F:(2,2.8)$	$B:(3,3.6)$ $F:(2,2.0)$
u_3	$B:(1,2.0)$ $F:(3,1.2)$	$B:(2,1.2)$ $F:(3,3.0)$	$B:(3,1.8)$ $F:(2,1.2)$	$B:(2,3.2)$ $F:(3,0.6)$	$B:(4,2.4)$ $F:(3,1.8)$

Lines 9-11. Finally, the release decision follows Line 13 of Algorithm 6 together with Lines 6-13 of Algorithm 4.

At time slot 1, for u_1, the forward and backward privacy budgets in Part_{DC} are $\epsilon^{(1)}_{F,1,1} = \frac{2.4}{2\times4} = 0.3$ and $\epsilon^{(1)}_{B,1,1} = \frac{1.0}{2} = 0.5$ according to Lines 2-4. The calculation budget is therefore $\epsilon^{(1)}_{1,1} = \min(0.5, 0.3) = 0.3$ in Line 5. In Part_{NOP}, the forward and backward budgets are $\epsilon^{(2)}_{F,1,1} = \frac{2.4}{2}/2 = 0.6$ and $\epsilon^{(2)}_{B,1,1} = \frac{1.0}{2} = 0.5$ according to Lines 9 and 10. Thus, the publication budget is $\epsilon^{(2)}_{1,1} = \min(0.5, 0.6) = 0.5$ in Line 11. The backward and forward budgets in Part_{DC} and Part_{NOP} are recorded as $[0.5, 0.3; 0.5, 0.6]$, while the calculation and publication budgets are recorded as $[0.3; 0.5]$. The budgets for u_2 and u_3 are shown below u_1's.

At time slot 2, take u_1 as an example, the forward and backward privacy budgets in Part_{DC} are $\epsilon^{(1)}_{F,1,2} = \min\left(\frac{3.2}{2\times4}, 0.3\right) = 0.3$ and $\epsilon^{(1)}_{B,1,2} = \frac{2.4}{2} - 0.3 = 0.9$ (according to Lines 2-4). The calculation budget is therefore $\epsilon^{(1)}_{1,2} = \min(0.9, 0.3) = 0.3$ in Line 5. According to the comparison in Line 13 in Algorithm 6 together with Lines 6-13 in Algorithm 4, there is no new publication occurs at this time slot. Thus, the publication budget usage is 0.

The budgets for the remaining three time slots are also recorded in Figure 8.

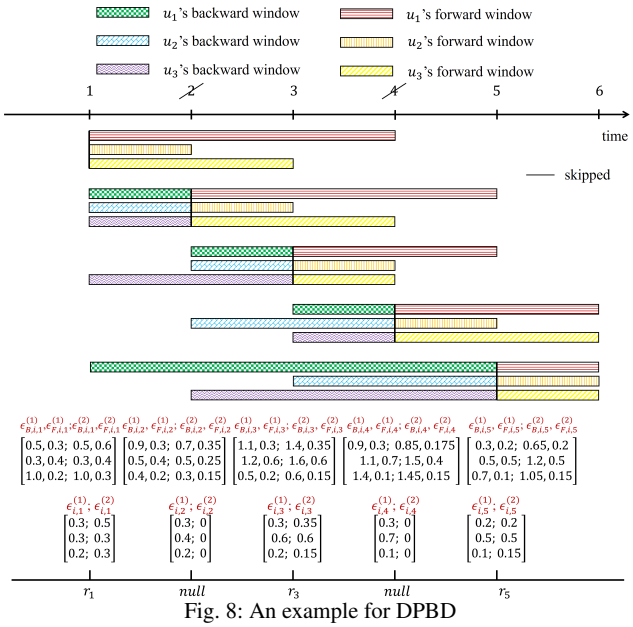


Fig. 8: An example for DPBD

Dynamic Personalized Budget Absorption. DPBA enhances PBA by supporting dynamic privacy requirements for users. Similar to DPBD, DPBA consists of two sub-mechanisms: Part_{DC} and Part_{NOP}. The private dissimilarity calculation in Part_{DC} remains identical to DPBD. In Part_{NOP}, the system determines whether to nullify the current time slot t for each user u_i based on publication budget usage at the relevant historical publication time slots. For each historical time slot $\tau \in T_{B,i,t}$ influencing time slot t, we calculate the nullified right time slot border $t_{FN,i,\tau}$ with total publication budget shares $\left(\text{where one share equals } \frac{\mathcal{E}_{F,i,\tau}}{2w_{F,i,\tau}}\right)$ from τ to $(t-1)$ (Line 7). For user u_i at time slot t, we determine the final forward nullified time slot right border $R_{FN,i}$ as the maximum value of these time slot borders (Line 8). We then obtain the forward nullified right border $\tilde{R}_{FN}$ as the maximum value among all $R_{FN,i}$ (Line 11). If the current time slot t is no larger than $\tilde{R}_{FN}$, t is nullified and skipped. Otherwise, for each u_i, we calculate the budget absorption $\epsilon_{AF,i,t}$ as the maximum absorption budgets among all historically influencing time slots (Line 16). We also determine the minimum remaining forward budgets $\epsilon_{UF,i,t}$ across all historical time slots as the remaining forward budget upper bound (Line 17). The forward absorption budget $\epsilon_{FA,i,t}$ is set as the minimum between $\epsilon_{AF,i,t}$ and $\epsilon_{UF,i,t}$ (Lines 16-18). Finally, we calculate the backward budget upper bound $\epsilon_{UB,i,t}$ (Line 19) and set the publication budget as the minimum between $\epsilon_{FA,i,t}$ and $\epsilon_{UB,i,t}$ (Line 20). The subsequent steps follow those in PBD.

Algorithm 7: Dynamic Personalized Budget Absorption (DPBA)

Input: D_t, dynamic personalized privacy requirement set $(\boldsymbol{w}_{B,1,t}, \boldsymbol{\mathcal{E}}_{B,t}, \boldsymbol{w}_{F,t}, \boldsymbol{\mathcal{E}}_{F,t})$, historical data publication $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1})$

Output: $\boldsymbol{r}_t$

1 **for** $i \in [n]$ **do**
2 Calculate $T_{B,i,t} \leftarrow \{\tau | \tau \le t \le \tau + w_{F,i,\tau} - 1\}$;
3 Calculate $\epsilon^{(1)}_{F,i,t} \leftarrow \min_{\tau \in T_{B,i,t}} \frac{\mathcal{E}_{F,i,\tau}}{2w_{F,i,\tau}}$;
4 Calcualte $\epsilon^{(1)}_{B,i,t} \leftarrow \mathcal{E}_{B,i,t}/2 - \sum_{\tau=t-w_{B,i,t}+1}^{t-1} \epsilon^{(1)}_{i,\tau}$;
5 Set $\epsilon^{(1)}_{i,t} \leftarrow \min\left(\epsilon^{(1)}_{F,i,t}, \epsilon^{(1)}_{B,i,t}\right)$;
6 **for** $\tau \in T_{B,i,t}$ **do**
7 Calculate u_i's forward nullified time slot right border from τ as $t_{FN,i,\tau} \leftarrow \frac{\sum_{\mu=\tau}^{t-1} \epsilon^{(2)}_{i,\mu}}{\mathcal{E}_{F,i,\tau}/(2 \cdot w_{F,i,\tau})} + \tau - 1$
8 Set u_i's forward nullified time slot right border as $R_{FN,i} \leftarrow \max_{\tau \in T_{B,i,t}} t_{FN,i,\tau}$;
9 $\boldsymbol{\epsilon}^{(1)}_t \leftarrow \left(\epsilon^{(1)}_{1,t}, \epsilon^{(1)}_{2,t}, \ldots, \epsilon^{(1)}_{n,t}\right)$;
10 Estimate $dis \leftarrow \mathrm{DC}\left(D_t, \boldsymbol{\epsilon}^{(1)}_t, \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{t-1}\right)$ by **Algorithm 3**;
11 Set forward nullified right border $\tilde{R}_{FN} \leftarrow \max_{i \in [n]} R_{FN,i}$;
12 **if** $t \le \tilde{R}_{FN}$ **then**
13 **return** $\boldsymbol{r}_t \leftarrow \boldsymbol{r}_{t-1}$;
14 **else**
15 **for** $i \in [n]$ **do**
16 Calcualte allocated forward absorption budget $\epsilon_{AF,i,t} \leftarrow \max_{\tau \in T_{B,i,t}} \left((t - t_{FN,i,\tau}) \cdot \frac{\mathcal{E}_{F,i,\tau}}{2 \cdot w_{F,i,\tau}}\right)$;
17 Calculate remaining forward budget upper bound $\epsilon_{UF,i,t} \leftarrow \min_{\tau \in T_{B,i,t}} \left(\frac{\mathcal{E}_{F,i,\tau}}{2} - \sum_{\mu=\tau}^{t-1} \epsilon^{(2)}_{i,\mu}\right)$;
18 Set forward absorption budget $\epsilon_{FA,i,t} \leftarrow \min\left(\epsilon_{AF,i,t}, \epsilon_{UF,i,t}\right)$;
19 Calculate remaining backward budget upper bound $\epsilon_{UB,i,t} \leftarrow \mathcal{E}_{B,i,t}/2 - \sum_{\tau=t-w_{B,i,t}+1}^{t-1} \epsilon^{(2)}_{i,\tau}$;
20 Set publication budget $\epsilon^{(2)}_{i,t} \leftarrow \min\left(\epsilon_{FA,i,t}, \epsilon_{UB,i,t}\right)$;
21 $\boldsymbol{\epsilon}^{(2)}_t \leftarrow \left(\epsilon^{(2)}_{1,t}, \epsilon^{(2)}_{2,t}, \ldots, \epsilon^{(2)}_{n,t}\right)$;
22 Same as Lines 15-22 in **Algorithm 5**

Example 7 Figure 9 illustrates the execution of DPBA. At each time slot, we first compute the candidate forward window set and the calculation budgets by Lines 2-5 of Algorithm 7. Then, the forward nullified right border is computed by Lines 6-8. If $t \le \tilde{R}_{FN}$, the publication is nullified according to Line 13; otherwise, we compute the forward absorption budget, the remaining backward budget upper bounds, and the final publication budget by Lines 15-20.

At time slot 1, $T_{B,i,1}$ for each u_i contains only the current time slot, resulting in $R_{FN,i} = 0$ for all u_i. Following Lines 16-18 of Algorithm 7, for u_1, we calculate $\epsilon_{AF,1,1} = \frac{\mathcal{E}_{F,1,1}}{2w_{F,i,1}} = \frac{2.4}{2\times 4} = 0.3$ and $\epsilon_{UF,1,1} = \frac{\mathcal{E}_{F,1,1}}{2} = 1.2$, leading to a forward absorption budget of $\epsilon_{FA,1,1} = 0.3$. For u_2 and u_3, we obtain $\epsilon_{FA,2,1} = 0.4$ and $\epsilon_{FA,3,1} = 0.2$. With backward remaining budgets $\epsilon_{UB,i,1}$ of 0.5, 0.3 and 1.0 for u_1, u_2 and u_3 respectively in Line 19, their final publication budgets are 0.3, 0.3, and 0.2 according Line 20.

At time slot 2, the publication is skipped, resulting in 0 publication budget usage for all users in Line 22 together with Lines 15-22 in Algorithm 5.

At time slot 3, following Line 2, $T_{B,1,3} = \{1, 2, 3\}$ for u_1. The forward nullified time slot right border for each time slot in $T_{B,1,3}$ is $\left\{\frac{0.3}{0.3} + 1 - 1, \frac{0}{0.4} + 2 - 1, \frac{0}{0.7} + 3 - 1\right\} = \{1, 1, 2\}$ according to Line 7. This yields an allocated forward absorption budget of $\epsilon_{AF,1,3} = \max((3-1) \times 0.3, (3-1) \times 0.4, (3-2) \times 0.7) = 0.8$ in Line 16. For u_2 and u_3, we calculate $\epsilon_{AF,2,3} = 1.2$ and $\epsilon_{AF,3,3} = 0.6$. The remaining forward budget upper bounds are $\epsilon_{UF,1,3} = \min\left(\frac{2.4}{2} - 0.3, \frac{3.2}{2}, \frac{4.2}{2}\right) = 0.9$ for u_1, $\epsilon_{UF,2,3} = 1.2$ for u_2, and $\epsilon_{UF,3,3} = 1.2$ for u_3 according to Line 17. This results in forward absorption budgets of $\epsilon_{FA,1,3} = \min(0.8, 0.9) = 0.8$, $\epsilon_{FA,2,3} = \min(1.2, 1.2) = 1.2$, and $\epsilon_{UF,3,3} = \min(0.6, 1.2) = 0.6$ in Line 18. The remaining backward budget upper bounds are $\epsilon_{UB,1,3} = \frac{2.8}{2} = 1.4$, $\epsilon_{UB,2,3} = \frac{3.2}{2} = 1.6$, and $\epsilon_{UB,3,3} = \frac{1.8}{2} - 0.2 = 0.7$ in Line 19, leading to final publication budgets of 0.8, 1.2 and 0.6 for u_1, u_2, and u_3 respectively in Line 20.

At time slot 4, according to Line 8, we calculate the forward nullified time slot right borders as $R_{FN,1} = 3.67$, $R_{FN,2} = 3.71$ and $R_{FN,3} = 4$. Since the current time slot $t = 4 \le \tilde{R}_{FN} = \max(3.67, 3.71, 4)$, following Lines 12-13, the publication is nullified, resulting in 0 publication budget usage for all users.

At time slot 5, following the same process as time slot 3, we obtain publication budget usage of 0.4, 0.6 and 0.3 for u_1, u_2, and u_3 in Line 20.

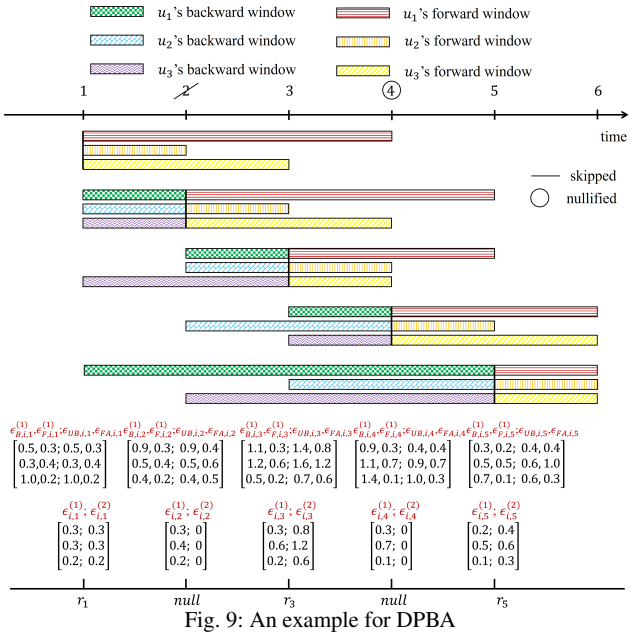


Fig. 9: An example for DPBA

5.3 Analysis

In this subsection, we analyze the time cost and privacy levels of our DPBD and DPBA.

Time Cost Analysis. For the time cost of DPBD and DPBA, we have Theorem 6 as follows.

Theorem 6 *The time complexities of DPBD and DPBA are both* $O\left(\left(w_{max}^{(F)} + w_{max}^{(B)}\right) \cdot n\right)$.

Proof For each u_i's $T_{B,i,t}$ at time slot t, we maintain a queue to record $T_{B,i,t}$ and update it by adding the next time slot $t+1$ and removing the time slots that no longer influence time slot $t+1$ to obtain $T_{B,i,t+1}$. The update time cost of all users' queues is $O\left(n \cdot w_{max}^{(F)}\right)$, where $w_{max}^{(F)}$ represents the maximum window size among all users' forward windows. For each user's historical publication budgets, we maintain another queue of size $w_{max}^{(B)}$, where $w_{max}^{(B)}$ represents the maximum window size among all users' backward windows. The time cost of calculating remaining backward budget upper bounds for all users is $O\left(n \cdot w_{max}^{(B)}\right)$. Thus, the time complexity of calculating $\boldsymbol{\epsilon}_t^{(1)}$ and $\boldsymbol{\epsilon}_t^{(2)}$ for each time slot is $O\left(\left(w_{max}^{(F)} + w_{max}^{(B)}\right) \cdot n\right)$. The sample mechanism time complexity is $O(n)$. Therefore, both DPBD and DPBA have a time complexity of $O\left(\left(w_{max}^{(F)} + w_{max}^{(B)}\right) \cdot n\right)$.

The remaining backward/forward budget computation is implemented here using a direct sliding-window summation for clarity. In practice, this step can be maintained incrementally via prefix sums or queue-based sliding-window data structures, reducing the amortized update cost from $O(w_{max})$ to $O(1)$ per user per time slot.

Memory Complexity Analysis. For DPBD and DPBA, we have Theorem 7 as follows.

Theorem 7 *Both DPBD and DPBA have complexity* $O\left(\left(w_{max}^{(F)} + w_{max}^{(B)}\right) \cdot n\right)$.

Proof For the process of OBS, the memory complexity is $O(m)$. For each one of the n users in both DPBD and DPBA, there are at most $w_{max}^{(F)}$ forward window size data and $w_{max}^{(B)}$ backward window size data need to be stored. Thus, the memory complexity is $O\left(\left(w_{max}^{(F)} + w_{max}^{(B)}\right) \cdot n\right)$.

Scalability Discussion. DPBD and DPBA remain scalable for large user populations for the same reason as the fixed case (PBD and PBA), while incurring only additional bookkeeping for backward/forward privacy requirements. Thus, their per-time slot cost still depends on the current users and bounded window states rather than the full stream history.

Privacy Analysis. The privacy analysis for DPBD and DPBA is presented in Theorem 8.

Theorem 8 *DPBD and DPBA satisfy* $(t, \boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$*-EPDP at each time slot* t*, where* $\boldsymbol{w}_B = (w_{B,1,t}, \ldots, w_{B,n,t})$ *and* $\boldsymbol{w}_F = (w_{F,1,t}, \ldots, w_{F,n,t})$ *represent the requirement sets for all users' backward and forward window sizes at time stamp* t*, and* $\boldsymbol{\mathcal{E}}_B = (\mathcal{E}_{B,1,t}, \ldots, \mathcal{E}_{B,n,t})$ *and* $\boldsymbol{\mathcal{E}}_F = (\mathcal{E}_{F,1,t}, \ldots, \mathcal{E}_{F,n,t})$ *represent the requirement sets for all users' backward and forward privacy budgets at time slot* t*.*

Proof Please refer to details of Theorem 8 in Appendix 8.4.2.

Utility Analysis under Periodic Dynamic Requirements. To obtain a closed-form average error bound, we analyze a common recurrent setting in which each user's privacy requirement sequence is periodic with period Y. That is for $t > Y$, each user's privacy requirement at time slot t matches that at time slot $t - Y$. This periodicity assumption is introduced only for the utility analysis; the mechanisms DPBD and DPBA, as well as their privacy guarantees, do not require periodic privacy requirements. Such periodic requirements arise naturally in applications where user privacy preferences follow daily or weekly routines. For example, a commuter may request stronger privacy protection during regular commuting hours and weaker protection during working hours, leading to a daily repeated requirement pattern. Similarly, drivers or delivery workers may exhibit recurring work/rest schedules that induce periodic changes in privacy requirements.

Besides, assume there are at most $\hat{s} \leq Y$ non-null publications occurring at time slots $t_1, t_2, \ldots, t_{\hat{s}}$. Assume each stream approximates the same number (ρ_{sk}) of skipped publications and the same number (ρ_{nu}) of nullified publications. Let $\mathcal{E}_L^{(F)}(i) = \min_t \mathcal{E}_{F,i,t}$ be the minimal proposed forward privacy budget among all time slots for each u_i. We define $\epsilon_L^{(B,M)}(i) = \frac{\mathcal{E}_L^{(F)}(i)}{2^{\beta_i}}$ as the lower bound of $\epsilon_{B,i,t}^{(1)}$ and $\epsilon_{B,i,t}^{(2)}$, where β_i is the parameter to be determined. We denote $\epsilon_{BL} = \min_{i\in[n]} \epsilon_L^{(B,M)}(i)$. Besides, we define $\epsilon_R^{(B,M)}(i) = \frac{\mathcal{E}_L^{(F)}(i)}{2^{\eta_i}}$ as the upper bound of $\epsilon_{B,i,t}^{(1)}$ and $\epsilon_{B,i,t}^{(2)}$, where η_i is the parameter to be determined. We also denote $\epsilon_{BR} = \max_{i\in[n]} \epsilon_R^{(B,M)}(i)$. Let $\epsilon_{FL}(i) = \min_t \frac{\mathcal{E}_{F,i,t}}{2w_{F,i,t}}$ be the half of minimal forward privacy budget share for u_i among all time slots. Let $\epsilon_{FLL} = \min_{i\in[n]} \epsilon_{FL}(i)$. Let $\epsilon_{FLR} = \max_{i\in[n]} \epsilon_{FL}(i)$. Let $\gamma_L = \min_{i\in[n]} \left(2^{\eta_i - 1} - 1\right)$ and $\gamma_R = \max_{i\in[n]} \left(2^{\beta_i - 1} - 1\right)$. Let $Z' = (n - n_B)\left(n - n_B + \frac{1}{4}\right)$ be the sampling error upper bound, where n_B is the quantity of $\max_{i\in[n],t\in[T]} \frac{\mathcal{E}_{F,i,t}}{w_{F,i,t}}$. For DPBD, we have Theorem 9 as follows.

Theorem 9 *The average error per time slot in DPBD is at most* $\min\left(\frac{2}{d^2(\min(\epsilon_{FLL}, \epsilon_{BL}))^2}, Z' + \frac{2}{d^2(\max(\epsilon_{FLR}, \epsilon_{BR}))^2}\right) + \min\left(\frac{2(4^{\hat{s}-\gamma_R+1}+3\gamma_R-4)}{3\hat{s}\epsilon_{BL}^2}, Z' + \frac{2(4^{\hat{s}-\gamma_L+1}+3\gamma_L-4)}{3\hat{s}\epsilon_{BR}^2}\right)$,

where $\gamma_L = \min_{i\in[n]}\left(2^{\eta_i-1}-1\right)$ *and* $\gamma_R = \max_{i\in[n]}\left(2^{\beta_i-1}-1\right)$, *if at most* $\hat{s}$ *non-null publications occur in any period* Y.

Proof Please refer to details of Theorem 9 in Appendix 8.5.3.

Interpretation of Theorem 9. The bound in Theorem 9 consists of the error from Part_{DC} and the accumulated publication error from Part_{NOP} over one period Y. The parameter $\hat{s}$ denotes the maximum number of non-null publications in a period, while γ_L and γ_R characterize the transition range where the forward publication budget decreases from being above the backward-budget interval to below it. Hence, the bound decomposes the average error into several stages of the DPBD process, rather than treating it as a single opaque expression.

Discussion on Consistency. When the dynamic forward and backward privacy requirements degenerate to fixed personalized privacy requirements, $\epsilon_{FL}(i) = \epsilon_L^{(B,M)}(i) = \epsilon_R^{(B,M)}(i) = \mathcal{E}_i/(2w_i)$. Besides, $\gamma_L = \gamma_R = 0$, $(\min(\epsilon_{FLL},\epsilon_{BL}))^2 = \min_{i\in[n]}(\mathcal{E}_i/(2w_i))^2$, $(\max(\epsilon_{FLR},\epsilon_{BR}))^2 = \max_{i\in[n]}(\mathcal{E}_i/(2w_i))^2$. For that matter, the error bound of DPBD reduces to the corresponding bound of the fixed personalized mechanism (PBD). This confirms that the dynamic analysis is consistent with the fixed-case analysis.

Discussion on Tightness. The bound in Theorem 9 is tight up to constant factors in the following sense. When the number of non-null publications in each period is small, the Part_{DC} term dominates, which matches the actual behavior of DPBD since most time slots reuse previous releases. When non-null publications occur close to the upper limit $\hat{s}$, the accumulated Part_{NOP} terms dominate, again matching the mechanism behavior because more fresh releases consume more privacy budget and incur more noise. Therefore, the bound captures the correct dominant error source in both sparse-update and frequent-update regimes.

For DPBA, we have Theorem 10 as follows.

Theorem 10 *The average error per time slot in DPBA is at most* $\min\left(\frac{2}{d^2(\min(\epsilon_{FLL},\epsilon_{BL}))^2}, Z' + \frac{2}{d^2(\max(\epsilon_{FLR},\epsilon_{BR}))^2}\right) + \frac{\widetilde{err}^{(s,p)}_{\text{Part}_{\text{NOP}}} + \rho_{nu}\overline{err}_{nlf}}{\rho_{sk}+\rho_{nu}+1}$ *where the value of* $\widetilde{err}^{(s,p)}_{\text{Part}_{\text{NOP}}}$ *is shown in Equation* (4).

Proof Please refer to details of Theorem 10 in Appendix 8.5.4.

Discussion on Frequent-Update Regimes. Similar to PBA, the error bound of DPBA becomes larger when skipped or nullified publications occur frequently. This effect concerns utility rather than privacy: the formal privacy guarantee of DPBA is still ensured by the budget-feasibility and composition analysis, whereas what may deteriorate in frequent-update regimes is estimation accuracy. In practice, DPBA is therefore more suitable for smoother streams in which reuse of previous releases is effective, while DPBD is more appropriate for streams with persistent rapid changes. This interpretation is also consistent with our experimental observations that the absorption-based methods perform better on smoother synthetic streams, whereas the distribution-based methods are more competitive on rapidly changing real datasets. Designing an adaptive switching strategy between these two mechanism families is an interesting direction for future work.

6 Experiments

6.1 Datasets

We evaluate our solutions on both real and synthetic datasets.

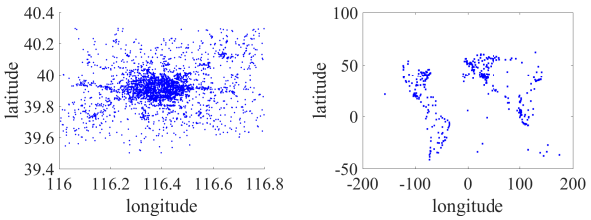
Real datasets. We use two real-world datasets, *Taxi* [47,48] and *Foursquare* [45,44], to evaluate the performance of our algorithms.

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 $\left([116, 116.8] \times [39.5, 40.3]\right)$ as A_E. Figure 10(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 10(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. Among the three synthetic sequence models, only TLNS involves randomness due to the Gaussian perturbation term, while the Sin and Log sequences are deterministic once their

$$\widetilde{err}^{(s,p)}_{\text{Part}_{\text{NOP}}} = \begin{cases} \min\left(\frac{2H^2_{\rho_{sk}+1}}{\epsilon^2_{FLL}}, Z'(\rho_{sk}+1) + \frac{2H^2_{\rho_{sk}+1}}{\epsilon^2_{FLR}}\right) & \text{if } \lambda_{LR} \geq \lambda_{RL}; \\ \min\left(\frac{2}{\epsilon^2_{FLL}} H^2_{\rho_{sk}+1}, Z'(\rho_{sk}+1) + \frac{2(\rho_{sk}+1)}{\epsilon^2_{BR}}\right) & \text{if } \lambda_{LR} < \lambda_{RL} \text{ and } \lambda_L < \rho_{sk}+1 \leq \lambda_R; \\ \min\left(\frac{2}{\epsilon^2_{FLL}} H^2_{\lambda_L}, Z'\lambda_L + \frac{2\lambda_L}{\epsilon^2_{BR}}\right) + (\lambda_R - \lambda_L)\min\left(\frac{2}{\epsilon^2_{BL}}, n\left(n+\frac{1}{4}\right)\right. & \\ \left. + \frac{2}{\epsilon^2_{BR}}\right) + \min\left(\frac{2(\rho_{sk}-\lambda_R+1)}{\epsilon^2_{BL}}, (\rho_{sk}-\lambda_R+1)Z' + \frac{2}{\epsilon^2_{FLR}} H^2_{\rho_{sk}-\lambda_R+1}\right) & \text{otherwise.} \end{cases} \tag{4}$$



(a) Taxi (b) Foursquare

Fig. 10: Illustration of Real datasets.

parameters are fixed. 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 $p_t = 1$. For reproducibility, the Gaussian perturbation in TLNS is generated using Java Random with a fixed seed of 1 in the revised implementation. Under the default setting with sequence length $T = 10,000$, clipping occurs in 8 out of $10,000$ time slots (0.08%). All clipping events correspond to values falling below 0, while no values exceed 1. This indicates that clipping is very rare and has negligible impact on the temporal trend and statistical properties of the generated sequence.
- 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 $b = 0.01$.

6.2 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, PBA, DPBD and DPBA with three non-personalized methods: Budget Distribution (BD), Budget Absorption (BA) [25] and SPAS [29]. We also compare against a simple personalized LDP method, Personalized LDP Budget Uniform (PLBU), which extends LDP Budget Uniform (LBU) [34] by replacing the inner CDP mechanism with an LDP mechanism.

Let $\mathcal{E}$ and w be the privacy budget and window size in non-personalized static methods (BD, BA and SPAS). For non-personalized static methods, we set the $\mathcal{E}$ to vary from 0.2 to 1.0 and w to vary from 40 to 200. To make our PBD and PBA comparable with the non-personalized static methods, we set the lower bound of each user's privacy budget as $\mathcal{E}$ and the upper bound of each user's window size as w in PBD and PBA to match the requirement of privacy level. Similarly, to make DPBD and DPBA comparable with PBD and PBA, we set the lower bound of each user's forward privacy budget as $\mathcal{E}_i$ and the upper bound of each user's forward window size as w_i. According to the design of DPBD and DPBA, the backward privacy requirement (backward privacy budget and window size) is independent of the forward privacy requirement. To study how the forward privacy requirement affects accuracy, we set the backward privacy level to a value that does not impact the publication budget decision, namely, $\epsilon^{(2)}_{F,i,t} = \min\left(\epsilon^{(2)}_{F,i,t}, \epsilon^{(2)}_{B,i,t}\right)$ in DPBD and $\epsilon_{FA,i,t} = \min(\epsilon_{FA,i,t}, \epsilon_{UB,i,t})$ in DPBA. Therefore, we set the backward privacy budget sufficiently large (i.e., $\mathcal{E}_{B,i,t} = 10$) and the backward window size sufficiently small (i.e., $w_{B,i,t} = 1$).

Table 4: Experimental settings.

Parameters	Values
static privacy budget $\mathcal{E}$	0.2, 0.4, **0.6**, 0.8, 1.0
static window size w	40, 80, **120**, 160, 200
personalized privacy budget $\mathcal{E}_i$	$\mathcal{E}, \ldots, 0.8, 1.0$
personalized window size w_i	$40, 80, \ldots, w$
users' quantity ratio o	0.1, 0.3, **0.5**, 0.7, 0.9
forward privacy budget $\mathcal{E}_{F,i,t}$	$\mathcal{E}_i, \ldots, 0.8, 1.0$
forward window size $w_{F,i,t}$	$40, 80, \ldots, w_i$
backward privacy budget $\mathcal{E}_{B,i,t}$	10
backward window size $w_{B,i,t}$	1

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 $\mathcal{E}_i = 0.5$ and $w_i = 10$ from 0.1 to 0.9.

The parameters are shown in Table 4, 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.

6.3 Measures

We evaluate the performance of different mechanisms based on their 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

$$AMRE = \frac{1}{T}\sum_{\tau=1}^{T} MRE_\tau = \frac{1}{T}\sum_{\tau=1}^{T}\frac{1}{d}\|\boldsymbol{r}_\tau - \boldsymbol{c}_\tau\|_2^2. \tag{5}$$

Besides, *AJSD* is defined as the average value of Jensen-Shannon Divergence (*JSD*, D_{JS}) [30], which is based on Kullback-Leibler Divergence [27], as

$$\begin{aligned}
&\bar{D}_{JS}(\boldsymbol{r}\|\boldsymbol{c})\\
=&\frac{1}{T}\sum_{\tau=1}^{T} D_{JS}(\boldsymbol{r}\|\boldsymbol{c})\\
=&\frac{1}{T}\sum_{\tau=1}^{T}\left(\frac{1}{2}D_{KL}(\boldsymbol{r}\|\boldsymbol{v}) + \frac{1}{2}D_{KL}(\boldsymbol{c}\|\boldsymbol{v})\right)\\
=&\frac{1}{2T}\sum_{\tau=1}^{T}\sum_{j=1}^{d}\left(\boldsymbol{r}_\tau(j)\log\left(\frac{\boldsymbol{r}_\tau(j)}{\boldsymbol{v}_\tau(j)}\right) + \boldsymbol{c}_\tau(j)\log\left(\frac{\boldsymbol{c}_\tau(j)}{\boldsymbol{v}_\tau(j)}\right)\right),
\end{aligned} \tag{6}$$

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

6.4 Overall Utility Analysis

Table 5 shows the average mean relative error *AMRE* as the privacy budget $\mathcal{E}$ varies. Across most datasets, *AMRE* generally decreases as $\mathcal{E}$ increases, since a larger $\mathcal{E}$ reduces the variance of the injected noise. However, the sensitivity of *AMRE* to $\mathcal{E}$ differs across mechanisms and datasets. On real datasets, distribution-based methods such as PBD and DPBD show a more evident reduction as $\mathcal{E}$ increases, whereas on synthetic datasets, absorption-based methods such as PBA and DPBA are more sensitive to the privacy budget and achieve larger reductions. This difference is mainly caused by the temporal characteristics of the streams: real datasets contain more abrupt changes and thus benefit from more responsive budget distribution, while synthetic datasets evolve more smoothly and benefit more from absorbing budgets for fewer but more accurate releases.

We attribute this difference mainly to the **temporal characteristics of the data streams**. The real datasets exhibit stronger temporal variability and more abrupt changes across consecutive time slots, whereas the synthetic datasets are relatively smoother over time. When the stream changes rapidly, the dissimilarity between the current statistics and the previous release becomes large, thus, mechanisms that allocate budget more responsively to the current time slot are more effective. 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. Therefore, PBD achieves lower *AMRE* than PBA on the real datasets.

In contrast, when the stream evolves more smoothly, the dissimilarity between consecutive time slots remains relatively small, when the density function changes gradually, the dissimilarity at each time slot remains small. In such cases, concentrating budget on fewer but more accurate releases is more beneficial than publishing more frequently. Therefore, PBA performs significantly better than PBD on the synthetic datasets.

DPBD performs better than PBD, while DPBA performs better than PBA. This improved performance occurs because the dynamic personalized methods maintains a higher privacy budget of at least $\mathcal{E}_i$ and a small window size of at most w_i compared to the personalized methods. Both a large privacy budget and a small window size contribute to improved accuracy. 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.

The comparison with SPAS further confirms this observation. On the two real datasets, Taxi and Foursquare, SPAS generally yields much larger *AMRE* than the proposed personalized mechanisms. Although SPAS adaptively allocates privacy budgets under homogeneous w-event privacy, it still relies on a single global privacy requirement and cannot exploit heterogeneous user-specific budgets and window sizes. This limitation becomes more evident on real streams with abrupt temporal changes, where more responsive personalized budget distribution is needed. In contrast, on the synthetic datasets, SPAS is more competitive with some distribution-based methods, especially when the stream evolves smoothly. Nevertheless, the absorption-based methods, particularly PBA and DPBA, still achieve smaller *AMRE* in most cases because they can skip or nullify unnecessary publications, accumulate more budget for informative releases, and use OBS to reduce the reporting error.

For the real datasets, our PBD consistently outperforms non-personalized methods. The *AMRE* of PBD is on average 72.6% lower than that of BD on Taxi dataset and 72.0% lower on Foursquare dataset. Besides, the *AMRE* of DPBD is on average 73.5% lower than that of BD on Taxi dataset and 93.3% lower on Foursquare dataset. We note that this performance gap is not primarily explained by dimensionality alone; the additional dimensionality analysis in Fig 17 in Appendix 8.3 shows that PBD/DPBD remain better than PBA/DPBA across different dimensions on the real datasets.

Table 5: Average Mean Relative Error (*AMRE*) with $\mathcal{E}$ varied.

Datasets	Methods	$\mathcal{E}$=0.2	$\mathcal{E}$=0.4	$\mathcal{E}$=0.6	$\mathcal{E}$=0.8	$\mathcal{E}$=1
Taxi	BD	8,459.58	2,156.19	990.69	610.25	409.43
	BA	3,050.51	1,495.79	961.62	679.88	558.63
	PLBU	34,419.70	34,415.11	34,417.54	34,416.61	34,416.79
	SPAS	618,995.29	154,449.26	68,648.03	38,946.65	25,004.66
	PBD	1,203.40	**449.91**	**275.14**	**204.53**	**166.84**
	PBA	2,874.89	1,369.62	1,041.83	869.90	723.00
	PDBD	**613.25**	**327.08**	**255.70**	**223.15**	**195.87**
	PDBA	**795.28**	534.17	461.53	419.98	396.67
Foursquare	BD	13,725.14	3,544.35	1,704.59	938.09	664.04
	BA	7,162.41	3,411.68	2,225.49	1,332.70	1,159.01
	PLBU	180,722.61	180,706.57	180,700.83	180,694.79	180,687.04
	SPAS	603,754.52	149,939.73	67,711.28	37,846.80	24,564.06
	PBD	2,185.22	**725.76**	**482.02**	**347.09**	**253.75**
	PBA	7,491.81	3,681.56	2,672.96	1,990.94	1,377.30
	PDBD	**460.58**	**167.42**	**114.14**	**82.10**	**66.04**
	PDBA	**1,675.06**	1,020.44	749.20	551.41	407.95
TLNS	BD	35,214,891.91	26,932,213.28	21,900,899.91	19,202,732.21	19,317,360.81
	BA	72,479.55	13,456.32	5,228.94	2,583.01	**1,447.21**
	PLBU	9,750,998.15	9,732,708.88	9,719,230.14	9,705,975.87	9,688,944.63
	SPAS	3,276,401.46	2,774,019.75	2,506,067.88	2,359,560.64	2,246,597.48
	PBD	5,625,685.13	5,708,081.90	5,761,339.31	5,761,668.46	6,168,585.37
	PBA	**15,734.77**	**7,593.93**	**4,031.24**	**2,415.89**	1,449.54
	PDBD	4,108,894.40	4,001,623.91	4,254,805.65	4,059,568.08	4,156,445.29
	PDBA	**3,249.53**	**2,096.95**	**1,309.70**	**1,097.65**	**860.35**
Sin	BD	14,809,297.94	3,868,423.13	2,822,001.85	2,166,052.12	2,875,381.82
	BA	61,604.76	21,010.09	9,193.50	4,678.60	**2,659.74**
	PLBU	18,127,404.96	18,103,836.01	18,082,416.37	18,052,150.71	18,028,552.72
	SPAS	234,146.85	208,385.79	190,648.17	185,012.37	179,534.63
	PBD	1,065,153.09	891,132.17	796,008.09	753,028.43	689,846.46
	PBA	**26,390.31**	**12,807.53**	**7,805.06**	**4,123.15**	2,661.26
	PDBD	376,035.19	331,161.83	321,423.27	315,928.35	312,774.69
	PDBA	**8,258.50**	**4,668.58**	**3,350.10**	**2,269.24**	**1,974.88**
Log	BD	12,598,827.10	3,106,749.11	1,520,101.37	1,159,412.40	912,889.07
	BA	25,313.21	7,067.12	3,701.09	2,351.14	**1,763.29**
	PLBU	6,334,895.72	6,323,757.64	6,316,533.20	6,306,368.84	6,298,869.11
	SPAS	23,030.09	21,985.85	21,573.75	21,520.27	20,983.06
	PBD	580,200.16	438,397.59	438,337.35	362,035.74	320,952.59
	PBA	**9,856.66**	**4,316.13**	**3,365.33**	**2,339.70**	1,837.17
	PDBD	75,571.46	59,450.90	50,179.75	51,748.11	46,332.37
	PDBA	**1,788.03**	**1,415.64**	**1,405.98**	**1,174.66**	**1,178.10**

For synthetic datasets, our PBA consistently outperforms other non-personalized methods and our DPBA further improves upon PBA. Compared to BA, the *AMRE* of PBA is lower on average of 30.2% on TLNS , 24.6% on Sin, and 21.1% on Log. Besides, the *AMRE* of DPBA is lower on average by 70.6% on TLNS, 61.0% on Sin, and 63.6% on Log. Moreover, our PBD consistently outperforms BD.

Table 6 shows the average mean relative error *AMRE* as the window size w varies. As w increases, *AMRE* generally increases. 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 most datasets, since LDP methods achieve lower accuracy than CDP methods under equivalent privacy budgets. SPAS is sensitive to the window size. Although it achieves very small *AMRE* on the real datasets when $w = 40$, its error increases sharply when the window size becomes larger. This suggests that the homogeneous adaptive strategy of SPAS becomes less robust when the privacy budget must be allocated over longer windows. In comparison, the proposed personalized mechanisms show more stable performance across different window sizes. On real datasets, PBD and DPBD achieve lower *AMRE* than the corresponding non-personalized baselines, while on synthetic datasets, PBA and DPBA remain more effective due to their budget absorption strategy.

For real datasets, our DPBD achieves the lowest error compared to others methods. The *AMRE* of PBD is on average 63.3% lower than that of BD on Taxi dataset and 65.8% on Foursquare dataset. Besides, the *AMRE* of DPBD is on average 62.7% lower than that of BD on Taxi dataset and 85.8% on Foursquare dataset. For synthetic datasets,

Table 6: Average Mean Relative Error (*AMRE*) with w varied.

Datasets	Methods	w=40	w=80	w=120	w=160	w=200
Taxi	BD	163.02	447.18	990.69	1,853.31	3,060.57
	BA	294.60	608.41	961.62	1,333.79	1,664.90
	PLBU	34,410.72	34,415.09	34,417.54	34,415.33	34,416.53
	SPAS	**29.37**	39,224.91	68,648.03	84,345.03	69,863.51
	PBD	100.42	**206.72**	**275.14**	**477.71**	**673.23**
	PBA	303.99	747.23	1,041.83	1,549.01	1,879.14
	PDBD	157.04	**177.12**	**255.70**	**275.78**	**297.00**
	PDBA	**98.77**	333.17	461.53	642.48	784.93
Foursquare	BD	241.39	687.78	1,704.59	2,910.86	4,783.33
	BA	255.27	1,240.90	2,225.49	3,112.29	3,758.29
	PLBU	180,654.54	180,701.20	180,700.83	180,713.97	180,722.91
	SPAS	**36.42**	37,353.95	67,711.28	81,441.64	68,097.75
	PBD	120.07	**297.72**	**482.02**	**819.29**	**1,022.65**
	PBA	981.55	1,988.55	2,672.96	3,715.36	4,636.11
	PDBD	**102.81**	**105.90**	**114.14**	**112.99**	**128.03**
	PDBA	109.47	388.57	749.20	1,245.98	1,634.49
TLNS	BD	10,321,505.61	16,427,785.34	21,900,899.91	22,192,516.74	22,368,956.59
	BA	387.96	1,948.21	5,228.94	9,421.72	16,065.42
	PLBU	9,629,126.83	9,695,808.71	9,719,230.14	9,732,079.75	9,741,718.28
	SPAS	4,002,347.34	1,766,335.27	2,506,067.88	3,060,865.41	3,527,531.79
	PBD	4,177,406.45	4,841,586.74	5,761,339.31	5,943,316.00	6,268,467.82
	PBA	**381.45**	**1,835.74**	**4,031.24**	**6,285.94**	**7,870.36**
	PDBD	3,579,033.68	3,799,459.12	4,254,805.65	4,320,520.18	4,344,081.28
	PDBA	**225.51**	**846.94**	**1,309.70**	**2,073.61**	**2,985.93**
Sin	BD	769,929.84	1,285,735.11	2,822,001.85	3,444,317.33	6,933,306.34
	BA	**611.27**	3,322.22	9,193.50	16,455.49	25,231.48
	PLBU	17,907,365.61	18,032,304.84	18,082,416.37	18,107,904.18	18,118,392.15
	SPAS	226,588.50	170,583.07	190,648.17	207,628.80	217,991.62
	PBD	378,448.77	572,317.51	796,008.09	1,144,032.83	1,339,405.34
	PBA	649.14	**3,298.99**	**7,805.06**	**13,784.27**	**17,342.63**
	PDBD	260,132.70	300,682.17	321,423.27	345,762.45	365,804.39
	PDBA	**445.32**	**1,764.71**	**3,350.10**	**5,505.09**	**7,833.50**
Log	BD	110,158.58	750,581.32	1,520,101.37	3,170,285.91	8,658,570.30
	BA	**770.17**	2,006.90	3,701.09	5,943.62	7,973.24
	PLBU	6,256,580.11	6,298,876.91	6,316,533.20	6,325,047.93	6,327,726.59
	SPAS	32,948.08	22,013.63	21,573.75	24,442.50	22,909.65
	PBD	47,909.95	207,973.60	438,337.35	582,733.67	759,567.65
	PBA	820.44	**1,946.36**	**3,365.33**	**4,270.21**	**6,007.43**
	PDBD	31,028.42	47,844.35	50,179.75	60,542.04	66,902.55
	PDBA	**633.30**	**1,162.11**	**1,405.98**	**1,617.49**	**2,122.58**

our DPBA demonstrates the lowest error among all non-dynamic methods. Compared to BA, the *AMRE* of PBA is lower by an average of 22.9% for TLNS, 11.4% for Sin, and 11.7% for Log, respectively. DPBA further reduces the *AMRE* by an average of 66.6% for TLNS, 54.6% for Sin, and 53.6% for Log, respectively. Moreover, our PBD consistently outperforms BD across all datasets.

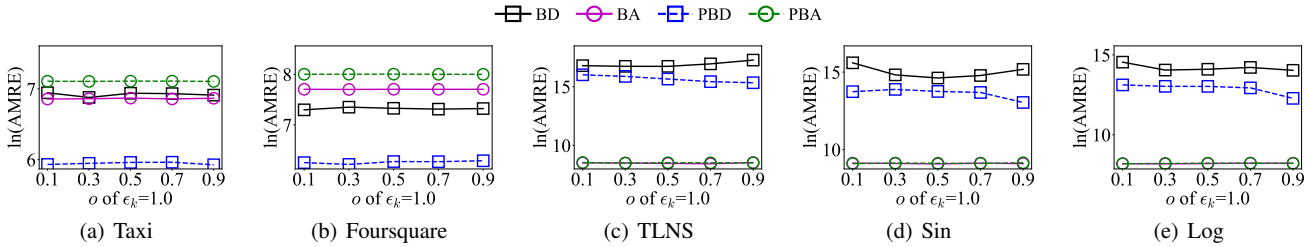
In summary, our PBD and DPBD demonstrate superior performance on real datasets, with an *AMRE* at least 63.3% and 62.7% lower than BD, respectively. For synthetic datasets, our PBA and DPBA outperform BA with at least 11.4% and 53.6% smaller *AMRE*, respectively.

For the experimental results under the AJSD metric, please refer to Appendix 8.2 for details.

6.5 Impact of User Requirement Type

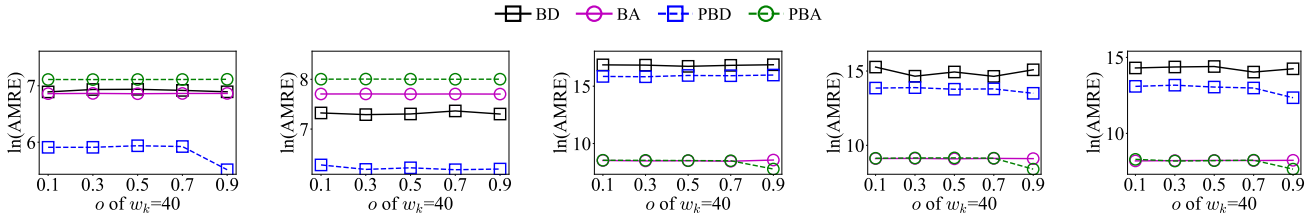
We define a set of users with privacy requirement as $(w_k, \mathcal{E}_k)$-*requirement type*. In this subsection, we examine the impact of user type on the utility. For our analysis, we set $\mathcal{E}_k$ candidate set as $\{0.6, 1.0\}$ with a default value of 0.6, and the w_k candidate set as $\{40,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

Figure 11 illustrates the change in users' quantity ratio for $\mathcal{E}_k = 1.0$ from 0.1 to 0.9, with a fixed window size of $w_k = 120$. Figure 12 shows the effect on changing users' quantity for $w_k = 40$ from 0.1 to 0.9, with a fixed privacy



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

Fig. 11: Average Mean Relative Error (*AMRE*) with ratio for privacy budget varied.



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

Fig. 12: Average Mean Relative Error (*AMRE*) with ratio for window size varied.

budget of $\mathcal{E}_k = 0.6$. We observe that as the users' quantity ratio increases, the *AMRE* remains relatively stable. However, when the users' quantity ratio of $\mathcal{E}_k = 1.0$ 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 $\mathcal{E}$, resulting in lower error.

7 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. Besides, we propose two dynamic solutions called DPBD and DPBA to solve this problem in scenarios with dynamic personalized privacy budget and window sizes. We also compare our PBD, PBA, DPBD and DPBA with recent solutions to demonstrate their efficiency and effectiveness.

Our future work will focus on developing local differential privacy mechanisms that adapt to users' evolving privacy preferences in local settings while preserving data utility. Besides, we plan to optimize our algorithms for high-speed data streams and expand their application to real-world scenarios like mobile crowd sensing and social media analysis.

References

1. Alaggan, M., Gambs, S., and Kermarrec, A. Heterogeneous differential privacy. *J. Priv. Confidentiality 7*, 2 (2016).
2. Andrés, M. E., Bordenabe, N. E., Chatzikokolakis, K., and Palamidessi, C. 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* (2013), A. Sadeghi, V. D. Gligor, and M. Yung, Eds., ACM, pp. 901–914.
3. Bao, E., Yang, Y., Xiao, X., and Ding, B. CGM: an enhanced mechanism for streaming data collection with local differential privacy. *Proc. VLDB Endow. 14*, 11 (2021), 2258–2270.
4. Bassily, R., and Smith, A. D. 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* (2015), R. A. Servedio and R. Rubinfeld, Eds., ACM, pp. 127–135.
5. Blum, A., Ligett, K., and Roth, A. 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* (2008), pp. 609–618.
6. Bolot, J., Fawaz, N., Muthukrishnan, S., Nikolov, A., and Taft, N. Private decayed predicate sums on streams. In *Joint 2013 EDBT/ICDT Conferences, ICDT '13 Proceedings, Genoa, Italy, March 18-22, 2013* (2013), pp. 284–295.
7. Chan, T. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur. 14*, 3 (2011), 26:1–26:24.
8. Chen, Y., Machanavajjhala, A., Hay, M., and Miklau, G. 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* (2017), pp. 1375–1388.
9. Cheu, A. Differential privacy in the shuffle model: A survey of separations. *CoRR abs/2107.11839* (2021).
10. Cheu, A., Smith, A. D., Ullman, J. R., Zeber, D., and Zhilyaev, M. Distributed differential privacy via shuffling. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I* (2019), Y. Ishai and V. Rijmen, Eds., vol. 11476 of *Lecture Notes in Computer Science*, Springer, pp. 375–403.

11. CUMMINGS, R., FELDMAN, V., MCMILLAN, A., AND TALWAR, K. 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* (2022).
12. DONG, W., LUO, Q., AND YI, K. Continual observation under user-level differential privacy. In *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023* (2023), pp. 2190–2207.
13. DU, L., CHENG, P., CHEN, L., SHEN, H. T., LIN, X., AND XI, W. Infinite stream estimation under personalized w-event privacy. *Proc. VLDB Endow. 18*, 6 (2025), 1111–1123.
14. DU, L., CHENG, P., ZHENG, L., XI, W., LIN, X., ZHANG, W., AND FANG, J. Dynamic private task assignment under differential privacy. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023* (2023), pp. 2740–2752.
15. DVIJOTHAM, K. D., MCMAHAN, H. B., PILLUTLA, K., STEINKE, T., AND THAKURTA, A. Efficient and near-optimal noise generation for streaming differential privacy. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024* (2024), pp. 2306–2317.
16. DWORK, C. Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II* (2006), M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of *Lecture Notes in Computer Science*, Springer, pp. 1–12.
17. DWORK, C. 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* (2010), pp. 174–183.
18. DWORK, C., NAOR, M., PITASSI, T., AND ROTHBLUM, G. N. 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* (2010), pp. 715–724.
19. DWORK, C., AND ROTH, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci. 9*, 3-4 (2014), 211–407.
20. ERLINGSSON, Ú., PIHUR, V., AND KOROLOVA, A. 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* (2014), G. Ahn, M. Yung, and N. Li, Eds., ACM, pp. 1054–1067.
21. FAN, L., AND XIONG, L. An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Trans. Knowl. Data Eng. 26*, 9 (2014), 2094–2106.
22. FENG, S., MOHAMMADY, M., WANG, H., LI, X., QIN, Z., AND HONG, Y. DPI: ensuring strict differential privacy for infinite data streaming. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024* (2024), pp. 1009–1027.
23. GUO, P., JIANG, T., ZHANG, Q., AND ZHANG, K. Sleep scheduling for critical event monitoring in wireless sensor networks. *IEEE Trans. Parallel Distributed Syst. 23*, 2 (2012), 345–352.
24. JORGENSEN, Z., YU, T., AND CORMODE, G. Conservative or liberal? personalized differential privacy. In *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015* (2015), pp. 1023–1034.
25. KELLARIS, G., PAPADOPOULOS, S., XIAO, X., AND PAPADIAS, D. Differentially private event sequences over infinite streams. *Proc. VLDB Endow. 7*, 12 (2014), 1155–1166.
26. KOTSOGIANNIS, I., DOUDALIS, S., HANEY, S., MACHANAVAJJHALA, A., AND MEHROTRA, S. One-sided differential privacy. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020* (2020), pp. 493–504.
27. KULLBACK, S., AND LEIBLER, R. A. On information and sufficiency. *The annals of mathematical statistics 22*, 1 (1951), 79–86.
28. LI, X., CAO, Y., AND YOSHIKAWA, M. Locally private streaming data release with shuffling and subsampling. In *39th IEEE International Conference on Data Engineering, ICDE 2023 - Workshops, Anaheim, CA, USA, April 3-7, 2023* (2023), IEEE, pp. 125–131.
29. LI, X., LI, T., CHENG, Y., GONG, C., REN, K., QIN, Z., AND WANG, T. SPAS: continuous release of data streams under w-event differential privacy. *Proc. ACM Manag. Data 3*, 1 (2025), 78a:1–78a:27.
30. LIN, J. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theory 37*, 1 (1991), 145–151.
31. LIU, J., LOU, J., XIONG, L., LIU, J., AND MENG, X. Projected federated averaging with heterogeneous differential privacy. *Proc. VLDB Endow. 15*, 4 (2021), 828–840.
32. MOON, W., HYUN, S., PARK, S., PARK, D., AND HEO, J. 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* (2023), pp. 23023–23033.
33. MURAKAMI, T., AND KAWAMOTO, Y. 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* (2019), pp. 1877–1894.
34. REN, X., SHI, L., YU, W., YANG, S., ZHAO, C., AND XU, Z. 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* (2022), pp. 1064–1077.
35. SUN, D., DONG, W., QIU, Y., AND YI, K. Personalized truncation for personalized privacy. *Proc. ACM Manag. Data 2*, 6 (2024), 249:1–249:25. SIGMOD 2025.
36. TENENBAUM, J., KAPLAN, H., MANSOUR, Y., AND STEMMER, U. Concurrent shuffle differential privacy under continual observation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA* (2023), A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 33961–33982.
37. WANG, Q., ZHANG, Y., LU, X., WANG, Z., QIN, Z., AND REN, K. 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* (2016), pp. 1–9.
38. WANG, S., LI, J., PENG, Y., CHEN, K., YANG, W., JIANG, H., AND LI, J. Differential private data stream analytics in the local and shuffle models. *IEEE Trans. Mob. Comput. 24*, 7 (2025), 6701–6717.
39. WANG, T., CHEN, J. Q., ZHANG, Z., SU, D., CHENG, Y., LI, Z., LI, N., AND JHA, S. 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* (2021), pp. 1237–1253.

40. WANG, Z., HU, J., LV, R., WEI, J., WANG, Q., YANG, D., AND QI, H. Personalized privacy-preserving task allocation for mobile crowdsensing. *IEEE Trans. Mob. Comput. 18*, 6 (2019), 1330–1341.
41. WANG, Z., LIU, W., PANG, X., REN, J., LIU, Z., AND CHEN, Y. 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* (2020), pp. 109–118.
42. XIE, Y., PAN, Z., MA, J., JIE, L., AND MEI, Q. 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* (2023), pp. 3892–3902.
43. XUE, Q., YE, Q., HU, H., ZHU, Y., AND WANG, J. DDRM: A continual frequency estimation mechanism with local differential privacy. *IEEE Trans. Knowl. Data Eng. 35*, 7 (2023), 6784–6797.
44. YANG, D., ZHANG, D., CHEN, L., AND QU, B. Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *J. Netw. Comput. Appl. 55* (2015), 170–180.
45. YANG, D., ZHANG, D., AND QU, B. 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.
46. YE, Q., HU, H., HUANG, K., AU, M. H., AND XUE, Q. Stateful switch: Optimized time series release with local differential privacy. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications, New York City, NY, USA, May 17-20, 2023* (2023), pp. 1–10.
47. YUAN, J., ZHENG, Y., XIE, X., AND SUN, G. 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* (2011), pp. 316–324.
48. YUAN, J., ZHENG, Y., ZHANG, C., XIE, W., XIE, X., SUN, G., AND HUANG, Y. 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* (2010), pp. 99–108.

8 Appendix

8.1 Running time Analysis

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

Figure 13 shows the average running time per time slot as the privacy budget varies from 0.2 to 1. The running time remains statble across different privacy budgets. This stability occurs because the privacy level does not impact on the running time. DPBD requires the highest computation time among all methods, while BD requires the least time. What's more, each of the personalized budget distribution methods (PBD, and DPBD) runs slower than its relative personalized budget absorption methods (PBA and DPBA). It is because personalized budget absorption methods are more likely to skip a publication than personalized budget distribution methods, which leads to fewer non-null publication calculations.

Figure 14 shows the average running time per time slot as the window size changes from 40 to 200. The running time decreases as the window size increases. This occurs because large window sizes result in more skipped time slots or nullified time slots, reducing the overall calculation time. Similar to Figure 13, BD requires the least running time among all methods, while personalized budget distribution methods (PBA and DPBA) require the more running time. It is because personalized methods introduce an optimal budget selection step that increases the running time. Compared to personalized budget absorption methods, the dissimilarity of personalized budget distribution methods increases more rapidly than the error, resulting in fewer skips or nullifications.

8.2 Experiments under *AJSD* Metric

Figure 15 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 is broadly consistent with the *AMRE* trend in Section 6.4. PLBU performs worse than CDP-based methods on most datasets, since LDP methods generally provide lower utility under the same privacy budget. Both PBD and PBA outperform BD across all datasets. DPBD achieves the lowest *AJSD* on the two real datasets, while DPBA performs best with the three synthetic datasets.

Notably, the *AJSD* ranking is not always identical to the *AMRE* ranking. This is expected because *AMRE* measures pointwise estimation error, whereas *AJSD* evaluates similarity between the released and true distributions. Therefore, *AJSD* captures a different aspect of utility and should be viewed as a complementary distribution-level metric, rather than as direct evidence for the same causal explanation used for *AMRE*.

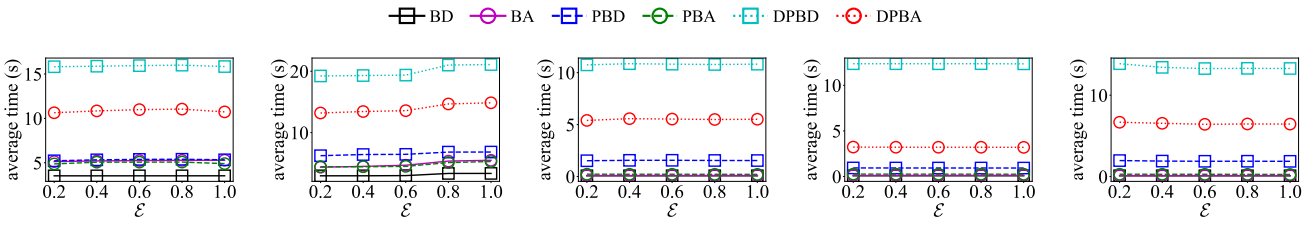
Figure 16 shows the results of *AJSD* as the window size w varies from 20 to 200. *AJSD* generally increases with larger window sizes for all methods, since a larger window reduces the effective privacy budget available at each time slot. PLBU again shows lower utility than the CDP-based methods on most datasets. Consistent with Figure 15, both PBD and PBA outperform BD. DPBD achieves the lowest *AJSD* on the two real datasets, while DPBA leads on the three synthetic datasets.

Similar to Figure 15, these results should be interpreted together with *AMRE* rather than in isolation. In particular, the discrepancy between *AMRE* and *AJSD* on Foursquare suggests that distribution-level similarity and pointwise estimation accuracy may favor different mechanisms. For this reason, we avoid attributing the *AJSD* behavior to a single factor such as dimensionality or sparsity.

The comparison with SPAS further supports the above observations under the *AJSD* metric. On the real datasets, SPAS usually yields larger *AJSD* than the proposed methods, which indicates that the proposed personalized mechanisms preserve distribution-level similarity more effectively than this homogeneous w-event baseline. On the synthetic datasets, SPAS can outperform some distribution-based methods, especially when the stream changes smoothly. However, PBA and DPBA still achieve the best or near-best *AJSD* in most cases, showing the advantage of budget absorption for smooth streams. Since *AMRE* measures pointwise relative error whereas *AJSD* evaluates the similarity between the released and true distributions, the results under these two metrics provide complementary evidence that the proposed personalized mechanisms improve data utility over a recent homogeneous w-event baseline while supporting a more general privacy model.

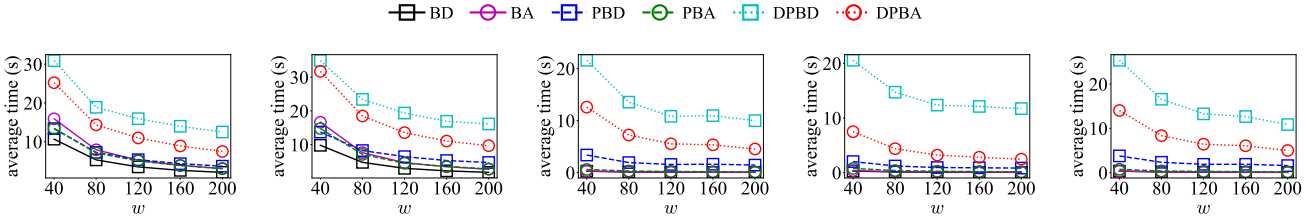
8.3 Experiments for Dimension Change

Figure 17 reports the utility under different dimensional settings. For AMRE, we observe that PBD/DPBD consistently outperform PBA/DPBA on the real datasets across all tested dimensions. This suggests that the performance gap is not primarily caused by dimensionality. Instead, the main reason is the temporal characteristics of the real data streams. Since real datasets usually exhibit more rapid and irregular changes across consecutive time slots, the absorption-based strategy in PBA/DPBA becomes less effective, because skipped or nullified updates are more difficult to approximate accurately using previous



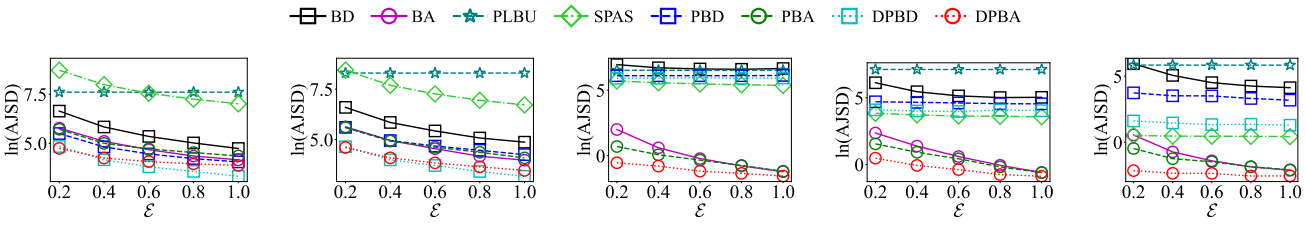
(a) Taxi (b) Foursquare (c) TLNS (d) Sin (e) Log

Fig. 13: The average running time per time slot with $\mathcal{E}$ varied.



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

Fig. 14: The average running time per time slot with w varied.



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

Fig. 15: Average Jensen-Shannon Divergencee (*AJSD*) with $\mathcal{E}$ varied.

releases. In contrast, PBD/DPBD allocate privacy budgets in a more responsive manner, which leads to lower pointwise estimation error. For AJSD, the trends are not always identical to those observed for AMRE. This is expected because AMRE measures pointwise estimation accuracy, whereas AJSD evaluates similarity between the released and true distributions. Therefore, we treat AJSD as a complementary utility metric that captures a different aspect of data quality. In the revised manuscript, we avoid attributing the AJSD behavior to a single factor such as dimensionality or sparsity, since the current results do not provide sufficient evidence for such a causal conclusion.

8.4 Proof for Privacy Analysis

8.4.1 Proof for Theorem 3

Proof We record $\max(t-w_i+1,1)$ as t_L for short.

(1) PBD satisfies $(\boldsymbol{w},\boldsymbol{\mathcal{E}})$-EPDP.

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

$$\sum_{k=t_L}^{t}\epsilon_{i,k}^{(1)}=\frac{\mathcal{E}_i}{2}. \tag{7}$$

In Part_{NOP}, for each user u_i at time slot t, only half of the publication budget is used when non-null publication occurs: $\epsilon_{i,t}^{(2)}=\left(\mathcal{E}_i/2-\sum_{k=t_L}^{t-1}\epsilon_{i,k}^{(2)}\right)/2$. For any time slot $t\in[1,w_i]$, the summation publication budgets used for u_i is at most $\sum_{k=1}^{w_i}\mathcal{E}_i/\left(2\cdot 2^k\right)\le(\mathcal{E}_i/2)\cdot(1-1/2^{w_i})\le\mathcal{E}_i/2$.

Assume $\sum_{k=t_L}^{t}\epsilon_{i,k}^{(2)}\le\mathcal{E}_i/2$ for $t=w_i+s$ (i.e., $\sum_{k=\max(s+1,1)}^{w_i+s}\epsilon_{i,k}^{(2)}\le\mathcal{E}_i/2$). Then for $t=w_i+s+1$, we have:

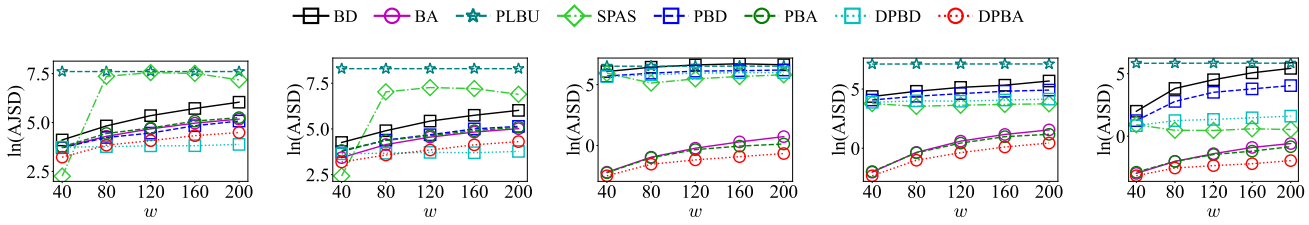
$$\sum_{k=\max(s+2,1)}^{w_i+s+1}\epsilon_{i,k}^{(2)}=\sum_{k=\max(s+2,1)}^{w_i+s}\epsilon_{i,k}^{(2)}+\epsilon_{i,w_i+s+1}^{(2)}. \tag{8}$$

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

$$\epsilon_{i,w_i+s+1}^{(2)}\le\frac{\frac{\mathcal{E}_i}{2}-\sum_{k=\max(s+2,1)}^{w_i+s}\epsilon_{i,k}^{(2)}}{2}. \tag{9}$$

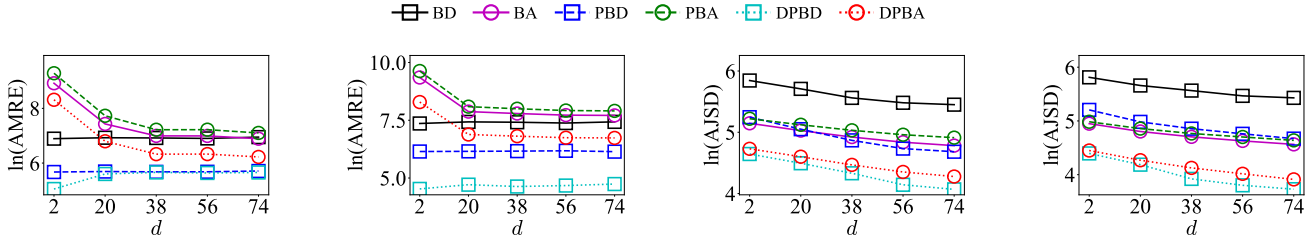
Let $k^*=\max(s+2,1)$. According to Equations (8) and (9), we have:

$$\begin{aligned}\sum_{k=k^*}^{w_i+s+1}\epsilon_{i,k}^{(2)}&\le\sum_{k=k^*}^{w_i+s}\epsilon_{i,k}^{(2)}+\frac{\frac{\mathcal{E}_i}{2}-\sum_{k=k^*}^{w_i+s}\epsilon_{i,k}^{(2)}}{2}\\&=\frac{\mathcal{E}_i}{4}+\frac{\sum_{k=k^*}^{w_i+s}\epsilon_{i,k}^{(2)}}{2}\\&\le\frac{\mathcal{E}_i}{4}+\frac{\mathcal{E}_i}{4}\\&=\frac{\mathcal{E}_i}{2}.\end{aligned}$$



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

Fig. 16: Average Jensen-Shannon Divergencee (*AJSD*) with w varied.



(a) Taxi (b) Foursquare (c) Taxi (d) Foursquare

Fig. 17: AMRE and AJSD with d varied.

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

$$\sum_{k=t_L}^{t} \epsilon_{i,k}^{(2)} \leq \frac{\mathcal{E}_i}{2}. \tag{10}$$

According to the Composition Theorems [19] and Equation (7) and Equation (10), we have:

$$\sum_{k=t_L}^{t} \epsilon_{i,k} = \sum_{k=t_L}^{t} \epsilon_{i,k}^{(1)} + \sum_{k=t_L}^{t} \epsilon_{i,k}^{(2)} \leq \mathcal{E}_i.$$

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 $\text{PBD}(S_t[t]) = o_t \in \mathcal{O}$, for any $O \subseteq \mathcal{O}$, we have:

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

Therefore, PBD satisfies $(\boldsymbol{w}, \boldsymbol{\mathcal{E}})$-EPDP where $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ and $\boldsymbol{\mathcal{E}} = (\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n)$.

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

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

$$\sum_{k=t_L}^{t} \epsilon_{i,k}^{(1)} = \mathcal{E}_i/2. \tag{11}$$

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 18 illustrates an example where $s_i = 3$ and $w_i = 9$.

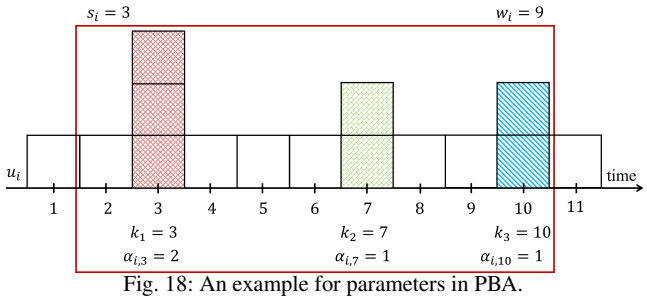


Fig. 18: An example for parameters in PBA.

Based on Algorithm 5, we have:

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

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

$$\begin{aligned}
\sum_{k=t_L}^{t} \epsilon_{i,k}^{(2)} &\leq \frac{\mathcal{E}_i}{2w_i} \cdot \sum_{j=1}^{s_i} (1 + \alpha_{i,k_j}) \\
&\leq \frac{\mathcal{E}_i \cdot \sum_{j=1}^{s_i} (1 + \alpha_{i,k_j})}{2\sum_{j=1}^{s_i} (1 + 2\alpha_{i,k_j}) - 2\alpha_{i,k_1} - 2\alpha_{i,k_{s_i}}} \\
&= \frac{\mathcal{E}_i \cdot \sum_{j=1}^{s_i} (1 + \alpha_{i,k_j})}{2\sum_{j=1}^{s_i} (1 + \alpha_{i,k_j}) + 2\sum_{j=2}^{s_i - 1} \alpha_{i,k_j}} \\
&\leq \frac{\mathcal{E}_i}{2}.
\end{aligned} \tag{12}$$

Based on Equations (11) and (12), and applying the Composition Theorems [19], we obtain:

$$\sum_{k=t_L}^{t} \epsilon_{i,k} = \sum_{k=t_L}^{t} \epsilon_{i,k}^{(1)} + \sum_{k=t_L}^{t} \epsilon_{i,k}^{(2)} \leq \mathcal{E}_i.$$

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

8.4.2 Proof for Theorem 8

Proof We analyze the privacy guarantees of DPBD and DPBA separately. Let $\hat{\mathcal{E}}_{B,i,t} = \hat{\mathcal{E}}^{(1)}_{B,i,t} + \hat{\mathcal{E}}^{(2)}_{B,i,t}$ represent the backward budget usage from $\max(t-w_{B,i,t}+1)$ to t for u_i, where $\hat{\mathcal{E}}^{(1)}_{B,i,t}$ and $\hat{\mathcal{E}}^{(2)}_{B,i,t}$ are the budget usages in sub-mechanism Part_{DC} and Part_{NOP} respectively. Let $\hat{\mathcal{E}}_{F,i,t} = \hat{\mathcal{E}}^{(1)}_{F,i,t} + \hat{\mathcal{E}}^{(2)}_{F,i,t}$ represent the forward budget usage from $\max(t-w_{B,i,t}+1)$ to t for u_i, where $\hat{\mathcal{E}}^{(1)}_{F,i,t}$ and $\hat{\mathcal{E}}^{(2)}_{F,i,t}$ are the budget usages in sub-mechanism Part_{DC} and Part_{NOP} respectively.

(1) DPBD satisfies $(t, \boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-EPDP at each time slot $t \in [T]$.

For the backward privacy budget usage of each u_i at time slot t, we have

$$\begin{aligned}\hat{\mathcal{E}}^{(1)}_{B,i,t} &= \sum_{k=\max(t-w_{B,i,t}+1,1)}^{t} \epsilon^{(1)}_{i,k} \\ &= \epsilon^{(1)}_{i,t} + \sum_{k=\max(t-w_{B,i,t}+1,1)}^{t-1} \epsilon^{(1)}_{i,k} \\ &\le \epsilon^{(1)}_{B,i,t} + \frac{\mathcal{E}_{B,i,t}}{2} - \epsilon^{(1)}_{B,i,t} \\ &= \frac{\mathcal{E}_{B,i,t}}{2}.\end{aligned} \tag{13}$$

Besides, it holds

$$\begin{aligned}\hat{\mathcal{E}}^{(2)}_{B,i,t} &= \sum_{k=\max(t-w_{B,i,t}+1,1)}^{t} \epsilon^{(2)}_{i,k} \\ &= \epsilon^{(2)}_{i,t} + \sum_{k=\max(t-w_{B,i,t}+1,1)}^{t-1} \epsilon^{(2)}_{i,k} \\ &\le \epsilon^{(2)}_{B,i,t} + \frac{\mathcal{E}_{B,i,t}}{2} - \epsilon^{(2)}_{B,i,t} \\ &= \frac{\mathcal{E}_{B,i,t}}{2}.\end{aligned} \tag{14}$$

From Equations (13) and (14), we have $\hat{\mathcal{E}}_{B,i,t} = \hat{\mathcal{E}}^{(1)}_{B,i,t} + \hat{\mathcal{E}}^{(2)}_{B,i,t} \le \mathcal{E}_{B,i,t}/2 + \mathcal{E}_{B,i,t}/2 = \mathcal{E}_{B,i,t}$.

For the forward privacy budget usage of each u_i at time slot t, in sub-mechanism Part_{DC} we have

$$\begin{aligned}\hat{\mathcal{E}}^{(1)}_{F,i,t} &= \sum_{k=t}^{t+w_{F,i,t}-1} \epsilon^{(1)}_{i,k} \\ &\le \sum_{k=t}^{t+w_{F,i,t}-1} \epsilon^{(1)}_{F,i,k} \\ &\le \sum_{k=t}^{t+w_{F,i,t}-1} \frac{\mathcal{E}_{F,i,t}}{2w_{F,i,t}} \\ &= \frac{\mathcal{E}_{F,i,t}}{2}.\end{aligned} \tag{15}$$

In sub-mechanism Part_{NOP}, given two time slot t and τ with $t \le \tau$, according to the calculation process of $\epsilon^{(2)}_{i,\tau}$, we have $0 \le \epsilon^{(2)}_{i,\tau} \le \epsilon^{(2)}_{F,i,\tau} \le \frac{1}{2}\left(\frac{\mathcal{E}_{F,i,t}}{2} - \sum_{k=t}^{\tau-1}\epsilon^{(2)}_{i,k}\right)$, thus $\sum_{k=t}^{\tau-1}\epsilon^{(2)}_{i,k} \le \frac{\mathcal{E}_{F,i,t}}{2} - 2\epsilon^{(2)}_{i,\tau}$. Therefore, it holds

$$\begin{aligned}\hat{\mathcal{E}}^{(2)}_{F,i,t} &= \sum_{k=t}^{t+w_{F,i,t}-1} \epsilon^{(2)}_{i,k} \\ &\le \frac{\mathcal{E}_{F,i,t}}{2} - 2\epsilon^{(2)}_{i,t+w_{F,i,t}} \\ &\le \frac{\mathcal{E}_{F,i,t}}{2}.\end{aligned} \tag{16}$$

From Equations (15) and (16), we have $\hat{\mathcal{E}}_{F,i,t} = \hat{\mathcal{E}}^{(1)}_{F,i,t} + \hat{\mathcal{E}}^{(2)}_{F,i,t} \le \mathcal{E}_{F,i,t}/2 + \mathcal{E}_{F,i,t}/2 = \mathcal{E}_{F,i,t}$.

For any user u_i at any time slot t and any two stream prefixes S_t, S'_t satisfying S_t and S'_t are $w_{B,i,t}$-neighboring (i.e., $S_t \sim_{B,t,w} S'_t$) and S_t and S'_t are $w_{F,i,t}$-neighboring (i.e., $S_t \sim_{F,t,w} S'_t$). Let t_s be the minimal time slot with $S_t[t_s] \ne S'_t[t_s]$ and t_e be the maximal time slot with $S_t[t_s] \ne S'_t[t_s]$. Then we have $t - t_s + 1 \le w_{B,i,t}$ and $t_e - t + 1 \le w_{F,i,t}$. Let $\epsilon_{i,t}$ be the privacy budget usage of u_i at time slot t. Let the output of our DPBD as $\text{DPBD}(S_t[t]) = o_t \in \mathcal{O}$. For any $O \subseteq \mathcal{O}$ we have

$$\begin{aligned}\frac{\Pr[\text{DPBD}(S_t) \in O]}{\Pr[\text{DPBD}(S'_t) \in O]} &\le \prod_{k=t_s}^{t_e} \frac{\Pr[\text{DPBD}(S_t[k]) = o_k]}{\Pr[\text{DPBD}(S'_t[k]) = o_k]} \\ &\le e^{\sum_{k=t_s}^{t_e}\epsilon_{i,k}} \\ &\le e^{\sum_{k=t_s}^{t}\epsilon_{i,k} + \sum_{k=t}^{t_e}\epsilon_{i,k}} \\ &= e^{\hat{\mathcal{E}}_{B,i,t} + \hat{\mathcal{E}}_{F,i,t}} \\ &\le e^{\mathcal{E}_{B,i,t} + \mathcal{E}_{F,i,t}}\end{aligned}$$

Let $\boldsymbol{w}_B = (w_{B,1,t}, \ldots, w_{B,n,t})$, $\boldsymbol{w}_F = (w_{F,1,t}, \ldots, w_{F,n,t})$, $\boldsymbol{\mathcal{E}}_B = (\mathcal{E}_{B,1,t}, \ldots, \mathcal{E}_{B,n,t})$ and $\boldsymbol{\mathcal{E}}_F = (\mathcal{E}_{F,1,t}, \ldots, \mathcal{E}_{F,n,t})$. Then, DPBD satisfies $(t, \boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-EPDP.

(2) DPBA satisfies $(t, \boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-EPDP at each time slot $t \in [T]$.

In DPBA, the backward privacy budget usage for each u_i at time slot t matches that of DPBD, which means $\hat{\mathcal{E}}_{B,i,t} \le \mathcal{E}_{B,i,t}$. Similarly, the forward privacy budget usage in Part_{DC} at time slot t is identical to DPBD, resulting in $\hat{\mathcal{E}}^{(1)}_{F,i,t} \le \mathcal{E}_{F,i,t}/2$.

Next, we consider the forward privacy budget usage of each u_i at time slot t in Part_{NOP}. For any two time slots t and τ where $t \le \tau$, based on the calculation process of $\epsilon^{(2)}_{i,\tau}$, we have $0 \le \epsilon^{(2)}_{i,\tau} \le \epsilon^{(2)}_{FA,i,\tau} \le \epsilon^{(2)}_{UF,i,\tau} \le \frac{\mathcal{E}_{F,i,t}}{2} - \sum_{k=t}^{\tau-1}\epsilon^{(2)}_{i,k}$. Thus, $\sum_{k=t}^{\tau-1}\epsilon^{(2)}_{i,k} \le \frac{\mathcal{E}_{F,i,t}}{2} - \epsilon^{(2)}_{i,\tau}$. Therefore, we have

$$\begin{aligned}\hat{\mathcal{E}}^{(2)}_{F,i,t} &= \sum_{k=t}^{t+w_{F,i,t}-1} \epsilon^{(2)}_{i,k} \\ &\le \frac{\mathcal{E}_{F,i,t}}{2} - \epsilon^{(2)}_{i,t+w_{F,i,t}} \\ &\le \frac{\mathcal{E}_{F,i,t}}{2}.\end{aligned}$$

Thus, $\hat{\mathcal{E}}_{F,i,t} = \hat{\mathcal{E}}^{(1)}_{F,i,t} + \hat{\mathcal{E}}^{(2)}_{F,i,t} \le \mathcal{E}_{F,i,t}$. The subsequent steps follow the same proof process as shown in DPBD. Therefore, DPBA satisfies $(t, \boldsymbol{w}_B, \boldsymbol{w}_F, \boldsymbol{\mathcal{E}}_B, \boldsymbol{\mathcal{E}}_F)$-EPDP.

8.5 Proof for Utility Analysis

8.5.1 Proof for Theorem 4

Proof Given a privacy budget-quantity pair set P and 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 fixed personalized privacy requirement $(w_i, \mathcal{E}_i)$, we calculate their average budget per window as $\frac{\mathcal{E}_i}{w_i}$. We denote the set of all average budgets as $\bar{\epsilon} = \left\{\frac{\mathcal{E}_i}{w_i} | i \in [n]\right\}$. We then construct the privacy budget-quantity pair set of each type of average budget as $P_A = \{(\epsilon_j, n_j) | \epsilon_j \in \bar{\epsilon}\}$. Let $Z = (n - n_A)\left(n - n_A + \frac{1}{4}\right)$ be the sampling error upper bound, where n_A is the quantity of $\max_{i\in[n]} \frac{\mathcal{E}_i}{w_i}$ in $\bar{\epsilon}$.

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

$$\begin{aligned}\overline{err}_{\text{NOP}} &= \frac{1}{w_L}\sum_{k\in[\tilde{s}]}\frac{w_L}{\tilde{s}}\cdot\widetilde{err}_O\left(\frac{1}{2^{k+1}}P_A\right)\\ &< \frac{1}{\tilde{s}}\sum_{k\in[\tilde{s}]}\min\left(\frac{2}{\left(\frac{\epsilon_L}{2^{k+1}}\right)^2}, Z+\frac{2}{\left(\frac{\epsilon_R}{2^{k+1}}\right)^2}\right)\\ &< \frac{1}{\tilde{s}}\min\left(\sum_{k\in[\tilde{s}]}\frac{8\cdot 4^k}{\epsilon_L^2}, \tilde{s}\cdot Z+\sum_{k\in[\tilde{s}]}\frac{8\cdot 4^k}{\epsilon_R^2}\right)\\ &= \min\left(\frac{32\cdot(4^{\tilde{s}}-1)}{3\tilde{s}\epsilon_L^2}, Z+\frac{32\cdot(4^{\tilde{s}}-1)}{3\tilde{s}\epsilon_R^2}\right).\end{aligned} \tag{17}$$

When Part_{DC} is private, the error from Part_{DC} can lead to two scenarios: (1) falsely skipping a publication or (2) falsely performing a non-null 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 $\overline{err}_{\text{DC}}$ of each time slot in window size w_L, according to Lemma 1, we have

$$\begin{aligned}\overline{err}_{\text{DC}} &< \min\left(\frac{2}{d^2\min_{i\in[n]}\left(\frac{\mathcal{E}_i}{2w_i}\right)^2}, Z+\frac{2}{d^2\max_{i\in[n]}\left(\frac{\mathcal{E}_i}{2w_i}\right)^2}\right)\\ &= \min\left(\frac{8}{d^2\epsilon_L^2}, Z+\frac{8}{d^2\epsilon_R^2}\right).\end{aligned} \tag{18}$$

Based on Equations (18) and (17), we can get the average error upper bound as $\overline{err}_{\text{DC}}+\overline{err}_{\text{NOP}}$.

8.5.2 *Proof for Theorem 5*

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 non-null publication corresponds to α skipped publications preceding it and α nullified publications succeeding it.

For each u_i's skipped publication, the publication privacy budget lower bound doubles with each time slot increasing until it reaches $\mathcal{E}_i/2$ or a non-null publication occurs. For example, in Figure 19, assume $\alpha=5$, the non-null publication time slot is t_6. At time slot t_1, each u_i's publication budget lower bound is $\mathcal{E}_i/(2w_i)$. Take u_1 as an example: it reaches $\mathcal{E}_1/2$ at time slot t_4. The publication lower bound for u_1 remains at $\mathcal{E}_1/2$ until time slot t_6. Let the publication budget lower bound set for all users at skipped publication 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 19, the error upper bound at t_3 is the error of publishing a new obfuscated statistic result using $\{\frac{3\mathcal{E}_1}{8},\frac{\mathcal{E}_2}{2},\frac{3\mathcal{E}_3}{16},\frac{\mathcal{E}_4}{4}\}$.

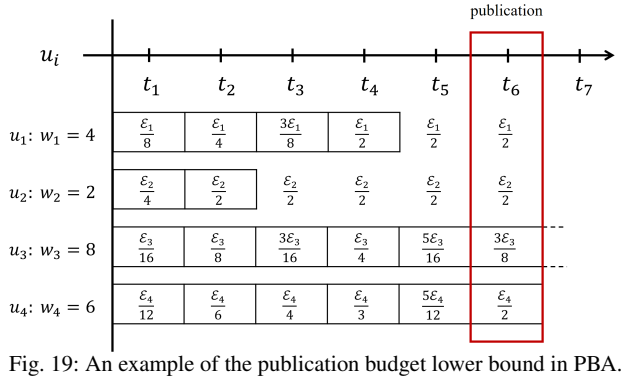


Fig. 19: An example of the publication budget lower bound in PBA.

Let $Z=(n-n_A)\left(n-n_A+\frac{1}{4}\right)$ be the sampling error upper bound, where n_A is the number of users with maximum value of $\frac{\mathcal{E}_i}{w_i}$. We now consider the following two cases:

Case $\alpha\le w_L$. In this case, the publication budget lower bound doubles with each time slot increasing. Let $err^{(sk)}_{\text{NOP}}(\alpha)$ and $err^{(pb)}_{\text{NOP}}$ be the total error upper bounds of the α skipped publications and the non-null publication in Part_{NOP}, respectively. Let $err^{(s,p)}_{\text{NOP}}$ be the error of all skipped publications and the publication in Part_{NOP}. According to Lemma 1, we have

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

and

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

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

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

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

Case $\alpha>w_L$. In this case, we have

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

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

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

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

When Part_{DC} is private, its error is identical to that in PBD:

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

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

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

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

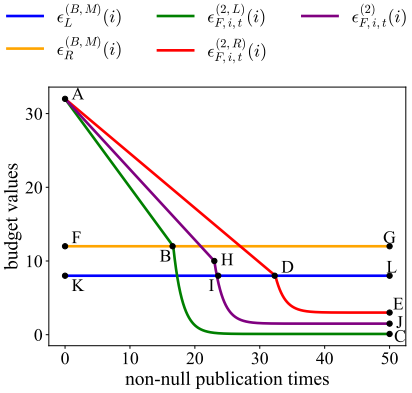


Fig. 20: An example of backward privacy budget bounds, where $\epsilon_L^{(B,M)}(i)$ and $\epsilon_R^{(B,M)}(i)$ represent the lower and upper bounds of $\epsilon_{B,i,t}^{(2)}$ respectively, $\epsilon_{F,i,t}^{(2,L)}(i)$ and $\epsilon_{F,i,t}^{(2,R)}(i)$ represent the lower and upper bounds of budget values, and $\epsilon_{F,i,t}^{(2)}(i)$ shows an example of budget values.

8.5.3 Proof for Theorem 9

Proof When Part_{DC} is not private, the error of DPBD is from the process of Part_{NOP}. This error is determined by $\boldsymbol{\epsilon}_t^{(2)} = \left(\epsilon_{1,t}^{(2)}, \ldots, \epsilon_{n,t}^{(2)}\right)$ at each time slot t, where each element $\epsilon_{i,t}^{(2)}$ depends on $\epsilon_{B,i,t}^{(2)}$ and $\epsilon_{F,i,t}^{(2)}$. For any $\epsilon_{B,i,t}^{(2)}$, there are two possible cases: either $\epsilon_{F,i,t}^{(2)} \geq \epsilon_L^{(B,M)}(i)$ (as shown by Curve *AHI* in Figure 20) or $\epsilon_{F,i,t}^{(2)} < \epsilon_L^{(B,M)}(i)$ (as shown by Curve *IJ* in Figure 20). Let γ_i be the number of non-null publications where $\epsilon_{F,i,t}^{(2)} \geq \epsilon_L^{(B,M)}(i)$. Since the current backward privacy budget can be any value within $[\epsilon_L^{(B,M)}(i), \epsilon_R^{(B,M)}(i)]$, we bound the transition point γ_i by considering the two extreme decay cases of the forward budget: the fastest decay with per-publication consumption $\epsilon_R^{(B,M)}(i)$, and the slowest decay with per-publication consumption $\epsilon_L^{(B,M)}(i)$. Therefore, after γ_i non-null publications, the forward budget needs to be no greater than $\epsilon_L^{(B,M)}(i)$ in the fastest-decay case, and no smaller than $\epsilon_R^{(B,M)}(i)$ in the slowest-decay case. Thus, we have

$$\frac{\mathcal{E}_L^{(F)}(i)}{2} - \gamma_i \cdot \epsilon_R^{(B,M)}(i) \leq \epsilon_R^{(B,M)}(i);$$
$$\frac{\mathcal{E}_L^{(F)}(i)}{2} - \gamma_i \cdot \epsilon_L^{(B,M)}(i) \geq \epsilon_L^{(B,M)}(i).$$

Thus,

$$2^{\eta_i - 1} - 1 \leq \gamma_i \leq 2^{\beta_i - 1} - 1.$$

Let $\hat{\boldsymbol{\epsilon}}_k^{(2)}$ be the privacy budget usage at time slot k in Part_{NOP}. Then for the average error $\overline{err}_{\text{Part}_{\text{NOP}}}$ of Part_{NOP} in DPBD, we have

$$\begin{aligned}
&\overline{err}_{\text{Part}_{\text{NOP}}} \\
=&\frac{1}{Y} \cdot \frac{Y}{\hat{s}} \sum_{k=1}^{\hat{s}} \widetilde{err}_O\left(\hat{\boldsymbol{\epsilon}}_k^{(2)}\right) \\
=&\frac{1}{\hat{s}}\left(\sum_{k=1}^{\gamma_L} \widetilde{err}_O\left(\hat{\boldsymbol{\epsilon}}_k^{(2)}\right) + \sum_{k=\gamma_L+1}^{\gamma_R} \widetilde{err}_O\left(\hat{\boldsymbol{\epsilon}}_k^{(2)}\right) + \sum_{k=\gamma_R+1}^{\hat{s}} \widetilde{err}_O\left(\hat{\boldsymbol{\epsilon}}_k^{(2)}\right)\right) \\
\leq& \min\left(\frac{2\gamma_L}{\hat{s}\epsilon_{BL}^2}, \frac{Z'\gamma_L}{\hat{s}} + \frac{2\gamma_L}{\hat{s}\epsilon_{BR}^2}\right) \\
&+ \min\left(\frac{2(\gamma_R - \gamma_L)}{\hat{s}\epsilon_{BL}^2}, \frac{Z'(\gamma_R - \gamma_L)}{\hat{s}} + \frac{32(4^{\gamma_R - \gamma_L} - 1)}{3\hat{s}\epsilon_{BR}^2}\right) \\
&+ \min\left(\frac{32(4^{\hat{s}-\gamma_R} - 1)}{3\hat{s}\epsilon_{BL}^2}, \frac{Z'(\hat{s} - \gamma_R)}{\hat{s}} + \frac{32(4^{\hat{s}-\gamma_L} - 4^{\gamma_R - \gamma_L})}{3\hat{s}\epsilon_{BR}^2}\right) \\
\leq& \min\left(\frac{2(4^{\hat{s}-\gamma_R+2} + 3\gamma_R - 16)}{3\hat{s}\epsilon_{BL}^2}, Z' + \frac{2(4^{\hat{s}-\gamma_L+2} + 3\gamma_L - 16)}{3\hat{s}\epsilon_{BR}^2}\right).
\end{aligned} \tag{24}$$

When Part_{DC} is private, the error from Part_{DC} will lead to falsely skipping or publishing a non-null publication. Both cases are bounded by the error in Part_{DC}. Let $\hat{\boldsymbol{\epsilon}}_k^{(1)}$ be the privacy budget usage at time slot k in Part_{DC}. For the error in Part_{DC}, we have

$$\begin{aligned}
\overline{err}_{\text{Part}_{\text{DC}}} =& \frac{1}{Y} \cdot \frac{Y}{\hat{s}} \sum_{k=1}^{\hat{s}} \widetilde{err}_O\left(\hat{\boldsymbol{\epsilon}}_k^{(1)}\right) \\
\leq& \min\left(\frac{2}{d^2(\min(\epsilon_{FLL}, \epsilon_{BL}))^2}, Z' + \frac{2}{d^2(\max(\epsilon_{FLR}, \epsilon_{BR}))^2}\right).
\end{aligned} \tag{25}$$

Thus according to Equation (25) and (24), we can get the error of DPBD as $err_{\text{DPBD}} = \overline{err}_{\text{Part}_{\text{DC}}} + \overline{err}_{\text{Part}_{\text{NOP}}}$.

8.5.4 Proof for Theorem 10

Proof Let ρ_{sk} be the number of skipped publications before a non-null publication and ρ_{nu} be the number of nullified publications after a non-null publication.

When the process of Part_{DC} is not private, the average error is only from Part_{NOP} which can further divided into skipped publication error, non-null publication error and nullified publication error. Let $\epsilon_{FL}(i) = \min_t \frac{\mathcal{E}_{F,i,t}}{2w_{F,i,t}}$, $\epsilon_{FLL} = \min_{i\in[n]} \epsilon_{FL}(i)$ and $\epsilon_{FLR} = \max_{i\in[n]} \epsilon_{FL}(i)$. Similar as that in the proof of Theorem 5, the publication lower bound doubles with the time slot increases. For each u_i, let $\lambda_L(i) = \frac{\epsilon_L^{(B,M)}(i)}{\epsilon_{FL}(i)/2}$ be the number of skipped time slots whose publication lower bound are no more than $\epsilon_L^{(B,M)}(i)$. Let $\lambda_R(i) = \frac{\epsilon_R^{(B,M)}(i)}{\epsilon_{FL}(i)/2}$ be the number of skipped time slots whose publication lower bound are no more than $\epsilon_R^{(B,M)}(i)$. Let $\lambda_{LR} = \max_{i\in[n]} \lambda_L(i)$ be the maximal value among all $\lambda_L(i)$. Let $\lambda_{RL} = \min_{i\in[n]} \lambda_R(i)$ be the minimal value among all $\lambda_R(i)$. If $\lambda_{LR} \geq \lambda_{RL}$, then the publication privacy budget lower bounds and upper bounds are determined by the forward publication budgets. Thus, for the error $err_{\text{Part}_{\text{NOP}}}^{(s,p)}$ in skipped publications and non-null publication, we have

$$\begin{aligned}
&err_{\text{Part}_{\text{NOP}}}^{(s,p)} \\
<& \sum_{k\in[\rho_{sk}+1]} \min\left(\frac{2}{(k\epsilon_{FLL})^2}, Z' + \frac{2}{(k\epsilon_{FLR})^2}\right) \\
<& \min\left(\frac{2}{\epsilon_{FLL}^2} H_{\rho_{sk}+1}^2, Z'(\rho_{sk}+1) + \frac{2}{\epsilon_{FLR}^2} H_{\rho_{sk}+1}^2\right).
\end{aligned}$$

If $\lambda_{LR} < \lambda_{RL}$, $err_{\text{Part}_{\text{NOP}}}^{(s,p)}$ can be classified into three cases that $\rho_{sk} + 1 \leq \lambda_L$, $\lambda_L < \rho_{sk} + 1 \leq \lambda_R$ and $\rho_{sk} + 1 > \lambda_R$. We denote the $err_{\text{Part}_{\text{NOP}}}^{(s,p)}$ as $err_{\text{Part}_{\text{NOP}}}^{(s,p,1)}$, $err_{\text{Part}_{\text{NOP}}}^{(s,p,2)}$ and $err_{\text{Part}_{\text{NOP}}}^{(s,p,3)}$ respectively.

(1) $\rho_{sk} + 1 \leq \lambda_L$. In this case, the publication privacy budget lower bounds are determined by the forward publication privacy budget lower bounds. The publication privacy budget upper bounds are determined by the backward publication privacy budget upper bounds. Thus, we have

$$\begin{aligned}
&err_{\text{Part}_{\text{NOP}}}^{(s,p,1)} \\
<& \sum_{k\in[\rho_{sk}+1]} \min\left(\frac{2}{(k\epsilon_{FLL})^2}, Z' + \frac{2}{\epsilon_{BR}^2}\right) \\
<& \min\left(\frac{2}{\epsilon_{FLL}^2} H_{\rho_{sk}+1}^2, Z'(\rho_{sk}+1) + \frac{2(\rho_{sk}+1)}{\epsilon_{BR}^2}\right).
\end{aligned}$$

(2) $\lambda_L < \rho_{sk} + 1 \leq \lambda_R$. In this case, the first λ_L publication privacy budget errors are the same as those in case (1). For the remaining errors, the publication privacy budget lower bounds and upper bounds are determined by the backward publication privacy budget lower bounds and upper bounds. Thus, we have

$$
\begin{aligned}
&err^{(s,p,2)}_{\text{Part}_{\text{NOP}}} \\
&< err^{(s,p,1)}_{\text{Part}_{\text{NOP}}}(\lambda_L) + \sum_{k=\lambda_L+1}^{\rho_{sk}+1} \min\left(\frac{2}{\epsilon^2_{BL}}, Z' + \frac{2}{\epsilon^2_{BR}}\right) \\
&< \min\left(\frac{2}{\epsilon^2_{FLL}} H^2_{\lambda_L}, Z'\lambda_L + \frac{2\lambda_L}{\epsilon^2_{BR}}\right) \\
&+ (\rho_{sk} - \lambda_L + 1) \min\left(\frac{2}{\epsilon^2_{BL}}, Z' + \frac{2}{\epsilon^2_{BR}}\right).
\end{aligned}
$$

(3) $\rho_{sk} + 1 > \lambda_R$. In this case, the first λ_R publication privacy budget errors are the same as those in case (2). For the remaining errors, the publication privacy budget lower bounds are determined by the backward publication privacy budget lower bounds. The publication privacy budget upper bounds are determined by the forward publication privacy budget upper bounds. Thus, we have

$$
\begin{aligned}
&err^{(s,p,3)}_{\text{Part}_{\text{NOP}}} \\
&< err^{(s,p,2)}_{\text{Part}_{\text{NOP}}}(\lambda_R) + \sum_{k=\lambda_R+1}^{\rho_{sk}+1} \min\left(\frac{2}{\epsilon^2_{BL}}, Z' + \frac{2}{(k\epsilon_{FLR})^2}\right) \\
&< \min\left(\frac{2}{\epsilon^2_{FLL}} H^2_{\lambda_L}, Z'\lambda_L + \frac{2\lambda_L}{\epsilon^2_{BR}}\right) \\
&\quad + (\lambda_R - \lambda_L) \min\left(\frac{2}{\epsilon^2_{BL}}, Z' + \frac{2}{\epsilon^2_{BR}}\right) \\
&\quad + \min\Bigg(\frac{2(\rho_{sk} - \lambda_R + 1)}{\epsilon^2_{BL}}, (\rho_{sk} - \lambda_R + 1)Z' \\
&\quad + \frac{2}{\epsilon^2_{FLR}} H^2_{\rho_{sk} - \lambda_R + 1}\Bigg).
\end{aligned}
$$

We denote the upper bound of $err^{(s,p)}_{\text{Part}_{\text{NOP}}}$ as $\widetilde{err}^{(s,p)}_{\text{Part}_{\text{NOP}}}$. Therefore, we can get the average error upper bound $\overline{err}_{\text{Part}_{\text{NOP}}}$ of each time slot in Part_{NOP} as

$$
\overline{err}_{\text{Part}_{\text{NOP}}} < \frac{1}{\rho_{sk} + \rho_{nu} + 1} \left(\widetilde{err}^{(s,p)}_{\text{Part}_{\text{NOP}}} + \rho_{nu}\overline{err}_{nlf}\right). \tag{26}
$$

When Part_{DC} is private, the average error $\overline{err}_{\text{Part}_{\text{DC}}}$ from Part_{DC} is the same as that in Theorem 9 (same as Equation (25)). From Equations (25) and (26), we can get the average error of DPBA as the sum of Equations (25) and (26).