LDP-IDS: Local Differential Privacy for Infinite Data Streams

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

[1]Xi'an Jiaotong University, Xi'an, China
[2]University of Warwick, Coventry, United Kingdom
[3]Imperial College London, London, United Kingdom

xuebinren@mail.xjtu.edu.cn,sl1624@stu.xjtu.edu.cn,weiren.yu@warwick.ac.uk
shusenyang@mail.xjtu.edu.cn,c.zhao@imperial.ac.uk,zbxu@mail.xjtu.edu.cn

ABSTRACT

Streaming data collection is essential to real-time data analytics in various IoTs and mobile device-based systems, which, however, may expose end users' privacy. Local differential privacy (LDP) is a promising solution to privacy-preserving data collection and analysis. However, existing few LDP studies over streams are either applicable to finite streams only or suffering from great utility loss due to simply adopting the budget division method in centralized differential privacy. In this paper, we study this problem by first proposing LDP-IDS, a novel LDP paradigm for infinite streams, and designing baseline approaches under the budget division framework. Particularly, we develop two budget division methods that are adaptive to sparsity changes in streams, with better data utility and communication efficiency. To improve the poor utility in budget division-based LDP, we then propose a population division framework that can not only avoid the high sensitivity of LDP noise to the budget division but also require significantly less communication. Under the population division framework, we also present two data-adaptive methods with theoretical analysis to further improve the estimation accuracy by leveraging the sparsity of data streams. We conduct extensive experiments on synthetic and real-world datasets to evaluate the effectiveness of utility of LDP-IDS. Experimental results demonstrate that, compared to the budget division-based solutions, our population division-based and data-adaptive algorithms for LDP-IDS can significantly reduce the utility loss and communication cost.

1 INTRODUCTION

The proliferation of smart devices and 5G technologies has been greatly boosting data streaming applications, such as event monitoring, log stream analysis, and video querying. These applications often adopt a client/server distributed architecture, where massive users' devices continuously produce data reports and the back-end server conducts real-time analytics over the aggregate data stream. Despite offering valuable information, the continuous collection of streaming data casts severe privacy risks. For example, call detail records of mobile phones can be collected for crowd analysis but

potentially reveal users' location [1]. Smart metering data can be mined to improve utility services, which, however, may expose users' daily activities [22].

Differential privacy (DP) has emerged as the de-facto standard for private data analysis with rigorous mathematical proof. DP for data streams has also attracted extensive interests. According to the granularity of privacy protection, these studies can be broadly classified into three categorizes: *event-level*, *user-level* and *w-event privacy*. Early researches mainly focus on *event-level privacy* for *finite streams* [4–6, 14, 15] and *user-level privacy* for *infinite streams* [19–21]. However, the former that hides a single event in streams is insufficient for protecting users' privacy, while the latter that protects a user's occurrence in infinite streams is impractical for most realistic scenarios [27]. To break the dilemma, *w-event privacy* for infinite streams is proposed [27], which aims to guarantee ϵ-DP for any time window consisting of w consecutive time instances (or time interval or timestamps for simplicity). Due to meaningful protection and applicability, *w-event privacy* has become the research trend and achieved fruitful results [37, 42, 43]. Nonetheless, these studies are based on *central/centralized differential privacy* (CDP), which relies on a trusted aggregator[1] and are prone to honest-but-curious adversaries.

Recently, *local differential privacy* (LDP) [11, 12, 26] has demonstrated a great potential in accomplishing analytic tasks without relying on a trusted aggregator. Unlike CDP, LDP has the advantage of guaranteeing massive end users' privacy locally, and thereby has been successfully deployed by many well-known corporations, e.g., Google [18], Microsoft [10], Apple [9], and Uber [23]. Contemporary studies on LDP mainly focus on static (non-streaming) data analysis, including frequency [18, 25, 38] and mean estimation [36, 44]. For evolving (streaming) data analytics, there are only a few work, including event-level LDP for infinite streams [24, 39] and user-level LDP for finite streams [3, 17]. Under event-level privacy, Joseph *et al.* [24] propose THRESH, which aims at reducing privacy loss at time slots with no significant population-wide updates. Despite being compatible to infinite streams, event-level LDP cannot protect a user's coarse-grained data. For user-level LDP, Bao *et al.* [3] present a correlated Gaussian mechanism CGM via utilizing autocorrelations in streams. However, under the analytic Gaussian mechanism, CGM achieves only approximate LDP (i.e., (ϵ, δ)-LDP) rather than pure LDP, and limited to finite streams only, meaning that the service has to be restarted periodically for infinite stream scenarios. Table 1 summarizes DP studies on data streams from both aspects of privacy granularity and applicable architecture. To the best of our knowledge, there is no prior work on *w-event*

---

[1]Or server, we use both terms interchangably in the paper.

arXiv:2204.00526v1 [cs.DB] 1 Apr 2022

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

Table 1: Summary of DP research on data streams

	Event-level	User-level	w-event level
CDP	[4–6, 14, 15]	[19–21]	[27, 37, 42, 43]
LDP	[24, 39]	[3, 17]	Our work

LDP for infinite streams, which can persistently provide strong and practical protection for indefinitely streaming data collection.

In this paper, we propose LDP-IDS, a pure ϵ-LDP based paradigm over *infinite streams* under the framework of *w-event privacy*. There are three technical challenges for LDP-IDS:

• **No access to raw data**. In CDP studies [27], to reduce overall noise, it is a common technique to mainly update at remarkable timestamps or assign different budget at different timestamps according to the sparsity in raw streams. However, this is difficult in LDP protocols, since the aggregator no longer has access to the raw data streams, which have to be perturbed at the user's end locally.

• **Utility loss in budge division**. Even if some methods can be formulated to mine the characteristics of raw streams underlying the LDP perturbed streams, the budget division methodology, commonly used in CDP, is not efficient in LDP. This is because the budget division, incurs only quadratic utility degradation in CDP, would incur an approximately exponential utility degradation in LDP [38, 41].

• **High communication cost**. As streaming data generates, massive end users persistently release their perturbed data to the aggregator at each timestamp. That often causes high communication cost for resource-constrained devices. It is desirable to consider the communication efficiency in designing LDP mechanisms.

To address the above challenges, we propose LDP-IDS, an LDP algorithm for infinite data streams. Specifically, in this paper, we make the following contributions:

- We first formulate the problem of infinite streaming data collection with LDP, which aims at realizing statistical analysis over LDP perturbed streams while providing meaningful privacy protection (i.e., w-event LDP).

- We construct a unified distortion analysis for streaming data analytics with LDP. Based on the analysis, we present two budge division-based baseline solutions, which dynamic allocates budget according to the non-deterministic sparsity in data streams. Compared to naive methods that evenly divide the privacy budget and enforce the same LDP in each time window of size w, the two baselines can effectively improve the utility via leveraging the stream characteristics.

- We propose a novel population division-based framework for streaming data collection with LDP, which achieves significantly higher data utility and less communication overheads. By building an analogy between budget division and population division, we design several population division-based solutions with much better utility and communication reduction than the baselines.

We implemented all proposed LDP algorithms for streams and conducted extensive experiment evaluation on both synthetic and real-world datasets. Experimental results show that, compared with the budget division algorithms, population division-based algorithms achieve significant reduction in utility loss and communication overhead, respectively. To further demonstrate the real-time

event monitoring performance, we evaluated the above-threshold detection performance of our LDP algorithms. Results shows that our methods effectively detect changes in LDP protected streams.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the background knowledge. Section 4 formulates the problem. Section 5 provides LDP solutions via the budget division methodology. Section 6 proposes the novel framework of population division, gives two detailed algorithms with extensive analysis. Section 7 presents an extensive set of experiment results. Finally, Section 8 concludes the paper.

2 RELATED WORK

Differential Privacy on Streams. *Centralized Differential Privacy* (CDP) [13] on streams or temporal data originally focuses on two notions: event-level DP and user-level DP [15]. The former aims to hide a single event in a stream while the latter tries to hide all events of a single user. For example, Dwork *et al.* [15] initiate the study and propose a binary tree technique based event-level DP algorithm for finite streams. Chan *et al.* [5] adopt the same technique to produce partial sum for binary counting in both finite and infinite data streams. Dwork *et al.* [14] further propose a cascade buffers counter with event-level DP to adaptively update the counter according to stream density. Bolot *et al.* [4] introduce the notion of decayed privacy and gradually reduce the privacy cost for past data. Then, Chen *et al.* [6] propose PeGaSus to answer multiple queries while satisifying event-level DP for data streams in a framework of perturb-group-smooth. Nonetheless, event-level DP is usually insufficient for privacy protection while user-level DP can only be achieved on a finite stream. For instance, Fan *et al.* [20] present FAST for realizing user-level DP on finite streams with a framework of sampling-and-filtering. To address the dilemma, Kellaris *et al.* [27] propose a notion of w-event DP for infinite streams, which ensures ϵ-DP for any time window including w consecutive timestamps. Based on a *sliding window* methodology, they further propose two methods satisfying w-event privacy, *Budget Distribution* (BD) and *Budget Absorption* (BA) to effectively allocate privacy budget considering that the statistics on streams may not change significantly in successive timestamps. Moreover, Wang *et al.* [37] propose a multi-dimensional stream release mechanism RescueDP by applying the idea of w-event DP to FAST and grouping the dimensions with similar trends. All in all, these CDP solutions cannot be directly applied to LDP settings since the untrusted server can no longer observe the raw data. Besides, LDP noise is generally larger than CDP noise under the same privacy parameter.

Local Differential Privacy (LDP). In the *local setting* where the server may be untrustworthy, LDP is proposed to perturb data at end users [11, 12, 26], which has also been extensively studied and applied into many analytic applications [7, 18, 29–33, 35, 38, 40, 41, 44, 45]. Nonetheless, most LDP studies concentrate on batch data analysis but seldom consider the stream settings. It has been demonstrated in [34] that, in Apple's LDP implementation, privacy loss accumulated in a short period would be too large to provide meaningful protection. To this end, Erlingsson *et al.* [18] introduce a memoization mechanism to provide longitudinal (i.e., long-term) LDP guarantee in cases when underlying true value changes in an uncorrelated fashion. Arcolezi *et al. et al.* [1] adopt

the same memoiazation technique to avoid average attack in login-tudinal analysis. Inspired by the binary tree technique in the CDP case [15], Erlingsson *et al.* [17] further propose an online protocol that guarantees longitudinal LDP regardless of whether the true value is independent or correlated. However, the construction of binary tree mainly applies to a finite stream, thus limiting its applications to infinite streams. Joseph *et al.* [24] propose an LDP algorithm THRESH for evolving data, which merely consumes privacy budget at global update timeslots that are selected via users' LDP voting. THRESH relies on the assumption of the number of global updates, and therefore is not applicable to infinite streams either. Besides, Wang *et al.* [39] extends a threshold-based data release algorithm from CDP to LDP for real-valued streams. Nevertheless, this work focuses on event-level LDP and lacks sufficient protection for infinite streams. Bao *et al.* [3] propose an (ϵ, δ) user-level LDP algorithm for finite streaming data collection using the analytic Gaussian mechanism, which focuses on approximate DP and has to renew privacy budget periodically. Wang *et al.* [42] propose a pattern-aware stream data collection mechanism with a metric based w-event LDP, which is not comparable to our work. More importantly, all these approaches enforce LDP over streams via a budget division methodology, which causes severe utility loss as reporting with low LDP budget is rather noisy.

Recently, several LDP studies [8, 30, 38] have shown that population division is generally better than budget division in LDP, which can be seen as the effect of amplification via subsampling [2]. Wang *et al.* [38, 41] point out that one can partition users to answer multiple questions with LDP, which still satisfy the same LDP under the parallel composition but can achieve much higher accuracy than splitting privacy budget and adopting the sequential composition. In particular, they further adopt this idea in the marginal release problem in LDP [45], where users are divided into different groups to report on different marginals but with the entire privacy budget. The similar user partition idea is actually adopted in [17], where each user randomly reports with the entire privacy budget on the nodes at a fixed level of the binary tree. Despite these pioneering studies, the idea of population division cannot be directly extended to infinite stream collection and analytics with LDP.

3 PRELIMINARIES

In this section, we first present the background about w-event privacy and existing methods in centralized setting. Then, we introduce LDP and its building block, frequency oracle.

3.1 w-event Privacy in Centralized Setting

On data streams, w-event privacy can strike a nice balance between event-level privacy for infinite streams and user-level privacy for finite streams. Therefore, we follow such a privacy definition in this paper and present its definition here.

We first give the notion about *stream prefix* and *neighboring streams*. A stream prefix of an infinite series $S = (D_1, D_2, ...)$ at timestamp t is defined as $S_t = (D_1, D_2, ..., D_t)$, where D_i is a snapshot of the stream at i. Let w be a positive integer, two stream prefixes S_t, S'_t are called w-*neighboring*, if for each $S_t[i], S'_t[i]$ such

that $i \in [t]$ and $S_t[i] \neq S'_t[i]$, it holds that $S_t[i], S'_t[i]$ are neighboring; and for each $S_t[i_1], S_t[i_2], S'_t[i_1], S'_t[i_2]$ with $i_1 \leq i_2, S_t[i_1] \neq S'_t[i_1]$ and $S_t[i_2] \neq S'_t[i_2]$, it holds that $i_2 - i_1 + 1 \leq w$.

Definition 3.1 (w-event Privacy [27]). Let $\mathcal{M}$ be a mechanism that takes as input a stream prefix of arbitrary size and O denotes the set of all possible outputs of $\mathcal{M}$. $\mathcal{M}$ satisfies w-event ϵ-DP (or, simply, w-event privacy) if for all sets $O \subseteq \mathcal{O}$, all w-neighboring stream prefixes S_t, S'_t, and all t, it holds that $\Pr[\mathcal{M}(S_t) \in O] \leq e^\epsilon \cdot \Pr[\mathcal{M}(S'_t) \in O]$.

A mechanism satisfies w-event privacy can provide ϵ-DP guarantee in any sliding window of size w. Or, for any mechanism with w-event privacy, ϵ can be viewed as the total available privacy budget in any sliding window of size w [27].

3.2 Existing Methods with w-event CDP

By properly allocating with different portions of the total budget ϵ, a mechanism composed of a series sub mechanisms over the timestamps can satisfy w-event privacy [27]. A naive method is to evenly apply ϵ/w-DP histogram release mechanism at every timestamp. Unfortunately, with the increase of w, the allocated budget becomes much small, which causes large perturbation noise at each timestamp. Another simple method is to release an ϵ-DP fresh histogram at one timestamp while other timestamps in a window is directly approximated with this result. However, the fixed sampling strategy cannot accurately follow the update patterns in the dynamic stream, thus leading to large errors.

BD (budget distribution) and BA (budget absorption) are benchmark adaptive methods for infinite stream release with w-event CDP[27]. Both BD and BA can be summarized into three components: *private dissimilarity calculation*, *private strategy determination*, and *privacy budget allocation*. In *private dissimilarity calculation*, a dissimilarity dis between the current $\mathbf{c}_t$ and the last update $\mathbf{c}_l$ is computed and perturbed with some fixed *dissimilarity budget* $\epsilon_{t,1}$. In *private strategy determination*, some *publication budget* $\epsilon_{t,2}$ is assigned (How to assign is designed in *privacy budget allocation*) for potential publication of noisy statistic, which can derive a potential publication error err. Then, dis and err is compared to decide the private strategy for current release. If $err < dis$, publish with perturbation (i.e., $\mathbf{r}_t = \mathbf{c}_t + \langle Lap(1/\epsilon_{t,2})\rangle^d$); otherwise, approximate by the previous release (i.e., $\mathbf{r}_t = \mathbf{c}_l$). In above process, $\epsilon_{t,1}$ is fixed for each timestamp, but $\epsilon_{t,2}$ is assigned based on different rules in BD and BA. In BD, $\epsilon_{t,2}$ is distributed in an exponentially decaying way to the timestamps where a publication is chosen, and reuses the budget spent in timestamps out of the current sliding window. While in BA, $\epsilon_{t,2}$ is uniformly assigned first and then unused budget is absorbed at timestamps where approximation is chosen.

3.3 Local Differential Privacy (LDP)

In the LDP paradigm, $\mathcal{M}$ is a randomized mechanism that takes each distributed user's input v and outputs a perturbed value before sending to the central aggregator, which collects the perturbed data and reconstructs the aggregated statistics.

Definition 3.2 (Local Differential Privacy). A mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy (i.e., ϵ-LDP), if and only if, for any

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]
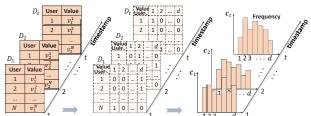

Figure 1: An illustration of streaming data release

input v and v' in domain $Dom(\mathcal{M})$, we have

$$\forall O \subseteq O, \Pr[\mathcal{M}(v) \in O] \leq e^\epsilon \cdot \Pr[\mathcal{M}(v') \in O],$$

where O is the set of all possible outputs of $\mathcal{M}$.

Despite receiving data from individuals, LDP ensures the central aggregator cannot infer the input with high confidence. As a DP variant, LDP inherits the properties of CDP, including sequential/parallel composition and post-processing theorems [16][28].

3.4 Frequency Oracle under LDP

LDP data analyses are commonly built on some frequency oracle (FO) protocols, which enable frequency estimation of any value v in a domain $\Omega = \{\omega_1, \omega_2, \ldots, \omega_d\}$ of size $d = |\Omega|$ under LDP. A common FO protocol is *Generalized Randomized Response (GRR)*. The idea of GRR method is that with a private data $v \in \Omega$, each user sends the true value to the central aggregator with probability p, and randomly sends a value in the candidate set $\Omega \setminus \{v\}$ with probability $1-p$. A GRR-based LDP mechanism $\mathcal{M}$ with the domain $Dom(\mathcal{M}) = \Omega$ is defined as follows.

$$\forall \bar{v} \in \Omega, \Pr[\mathcal{M}(v) = \bar{v}] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + d - 1}, & \text{if } \bar{v} = v \\ q = \frac{1}{e^\epsilon + d - 1}, & \text{if } \bar{v} \neq v \end{cases} \quad (1)$$

GRR-based FO works as follows. The aggregator aims to calculate the *frequency* of each distinct item v or $\omega_k \in \Omega$, denoted as $\mathbf{c}[k]$. It first counts the frequency of ω_k in perturbed data, which is denoted as $\mathbf{c}'[k]$. Then, assuming n is the number of participant users, the estimated frequency $\bar{\mathbf{c}}[k]$ of ω_k through GRR protocol can be obtained as $\bar{\mathbf{c}}_{\text{GRR}}[k] = (\mathbf{c}'[k]/n - q)/(p - q)$. We use $\text{FO}(\overline{D}_t, \epsilon)$ to denote this FO process. It is shown in [38] that this is an unbiased estimation of the true frequency, with the variance

$$\text{Var}[\bar{\mathbf{c}}_{\text{GRR}}[k]; \epsilon, n] = \frac{d - 2 + e^\epsilon}{n \cdot (e^\epsilon - 1)^2} + \frac{f_k \cdot (d - 2)}{n \cdot (e^\epsilon - 1)} \quad (2)$$

where f_v is the frequency of v and there is $\sum_{k=1}^d f_k = 1$. Considering f_v is often small, the above variance is also simply approximated as $\frac{d - 2 + e^\epsilon}{n \cdot (e^\epsilon - 1)^2}$ [45]. Although there are also other FOs (e.g., OUE) [45], the estimation variance of these FOs can all be seen as a function of parameter ϵ and population n.

For simplicity, without specifying the FO used, we use $V(\epsilon, n)$ to represent the estimation variance from n users with budget ϵ.

4 PROBLEM DEFINITION

In this subsection, we propose LDP-IDS, a novel LDP paradigm for infinite streams under the framework of w-event privacy.

We assume that there is a distributed system consisting of a central server and N distributed users $\{1, 2, \ldots, N\}$ that continuously report the value of a data item (e.g., user's location in an area, or

units of certain measurement) at discrete timestamps. For example, an area is divided into d disjoint regions and the server aims to illustrate the time-evolving population density map from users' location reports. Let v_t^j represent the report of user j at timestamp t and v_t^j come from a domain Ω with the carnality of d. Then, each user has an infinite data stream $V^j = (v_1^j, v_2^j, \ldots)$. Meanwhile, at every timestamp t, the central server receives all users' reports $\{v_t^1, \ldots, v_t^N\}$, which can be transformed into a binary database D_t with with d columns and n rows. As shown in Fig. 1, the server aims to release the statistical histogram $\mathbf{c}_t = \langle \mathbf{c}_t[1], \mathbf{c}_t[2], \ldots, \mathbf{c}_t[d] \rangle$ over all n users' data continuously at timestamp t. For example, $\mathbf{c}_t[k] = \frac{1}{n} \sum_j \mathbb{1}_{\{k|v_t^j = \omega_k\}}(k)$ is the frequency [2] of every unique value in Ω and $\mathbb{1}_X(k)$ is an indicator function that equals to 1 if $k \in X$, and 0 otherwise. However, direct data collection or release would compromise individual's privacy. Specifically, we consider that the central server is not trustworthy and assumed to be *honest-but-curious*. Instead of directly reporting v_t^i, each user would choose to send a perturbed value $\bar{v}_t^j$ with LDP or report nothing at each timestamp t. Therefore, our goal is to design an LDP solution that helps the server to collect data and release an estimated histogram $\mathbf{r}_t = \langle \mathbf{r}_t[1], \mathbf{r}_t[2], \ldots, \mathbf{r}_t[d] \rangle$ at each timestamp t where $\mathbf{r}_t[k]$ denotes the estimated frequency for each value in the domain.

Considering the infinity of streaming data, users also wish to adopt a meaningful privacy paradigm similar to w-event privacy in the centralized setting. We naturally extend the definition of w-event privacy to the local setting. Before that, we first define the notion of w-neighboring in the local setting as follows.

Definition 4.1 (w-neighboring). Let V_t and V_t' denote two stream prefixes defined on the same domain Ω^t. Let w be a positive integer. V_t and V_t' are w-neighboring, if for each $V_t[i_1], V_t[i_2], V_t'[i_1], V_t'[i_2]$ with $i_1 \leq i_2$, $V_t[i_1] \neq V_t'[i_1]$ and $V_t[i_2] \neq V_t'[i_2]$, it holds that $i_2 - i_1 + 1 \leq w$.

That is to say, if two stream prefixes are w-neighboring, then their elements are *the same* while all their *same* elements consist of a window of up to w timestamps. This is slightly different from the definition in the central setting.

Definition 4.2. [w-event LDP] Let $\mathcal{M}$ be a mechanism that takes as input stream prefix $V_t = (v_1, v_2, \ldots, v_t)$ consisting of a single user's arbitrary number of consecutive input value v_t. Also let O be the set of all possible outputs of $\mathcal{M}$. We say that $\mathcal{M}$ satisfies w-event ϵ-LDP (i.e., w-event LDP) if for any w-neighboring stream prefixes V_t, V_t', and all t, it holds that

$$\forall O \subseteq O, \Pr[\mathcal{M}(V_t) \in O] \leq e^\epsilon \Pr[\mathcal{M}(V_t') \in O].$$

In other words, a w-event LDP mechanism will provide each user ϵ-LDP for any sliding window of size w.

5 BUDGET DIVISION-BASED METHODS

In this section, we first present the budget division framework for streaming data collection with LDP. Then, based on this framework, we introduce our LDP methods for the problem defined.

---

[2]Other aggregate analyses, such as count and mean estimation, can be applicable, as the query type is orthogonal to the streaming data setting.
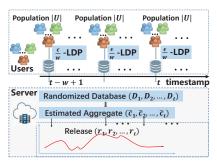
Figure 2: Illustration of budget division framework

5.1 Budget Division Framework for LDP

Inspired by the studies in the centralized setting, the following theorem can be derived for designing LDP mechanisms.

THEOREM 5.1. *Let $\mathcal{M}$ be a mechanism that takes as input stream prefix V_t consisting of a single user's arbitrary number of consecutive input value v_t, i.e., $V_t[i] = v_i$, and outputs a transcript $o = (o_1, o_2, ..., o_t) \in Range(\mathcal{M})$. Suppose that we can decompose $\mathcal{M}$ into t mechanisms $\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_t$, such that $\mathcal{M}_i(v_i) = o_i$, each $\mathcal{M}_i$ generates independent randomness and achieves ϵ_i-LDP. Then, $\mathcal{M}$ satisfies w-event ϵ-LDP if for any user and any timestamp $i \in [t]$, there is $(\sum_{\tau=i-w+1}^{i} \epsilon_\tau) \leq \epsilon$.*

PROOF. See Appendix A.1. □

This theorem enables a w-event LDP mechanism to view ϵ as the total privacy budget in any sliding window of size w, and appropriately allocate portions of it across the timestamps, as shown in Fig. 2. According to the above theorem, some straightforward approaches can be summarized to solve the problem defined in Section 4, based on different allocation methods of the LDP budget.

5.2 Baseline w-event LDP Methods

*5.2.1 **LDP Budget Uniform Method** (LBU).* One straightforward approach is to uniformly assign the LDP budget ϵ to all w timestamps in the sliding windows. At each timestamp, each user reports the perturbed value with an FO using the fixed budget ϵ/w for satisfying w-event LDP. Recall that $V(\epsilon, n)$ represents the LDP estimation variance from n users with privacy budget ϵ, without specifying the FO. Since $\mathbf{r}_t$ is an unbiased estimate of $\mathbf{c}_t$, the mean square error (MSE) between the true stream prefix $C_t = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_t)$ and the released stream prefix $R_t = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_t)$, is indeed the estimation variance of $\mathbf{r}_t$ i.e., $\text{MSE}_{\text{LBU}} = \text{Var}[\mathbf{r}_t; \epsilon/w, N] = V(\epsilon/w, N)$. If w is large, privacy budget allocated at each timestamp is very small, leading to a large noise scale.

*5.2.2 **LDP Sampling Method** (LSP).* Each user invests the entire budget ϵ on a single (sampling) timestamp within the window, while saving budget for the next $w - 1$ timestamps via approximation. At the last sampling timestamp l, the MSE of LSP equals to estimation variance $V(\epsilon, N)$. For non-sampling timestamps, it equals to the sum of the variance of last release at the sampling timestamp (i.e., $\text{Var}[\mathbf{r}_l]$ [3]), and the variance of true statisics at the current timestamp t from that at the last sampling timestamp l (i.e., $(\mathbf{c}_t - \mathbf{c}_l)^2$, which

---

[3]For simplicity, we use $\text{Var}[\mathbf{r}_l]$ to denote the average variance over d dimensions of vector$\mathbf{r}_l$, or $\text{Var}[\mathbf{r}_l] = \frac{1}{d} \sum_{k=1}^{d} \text{Var}(\mathbf{r}_l[k])$.

is data dependent). Therefore, the MSE of LSP in a window of size w can be calculated as $\text{MSE}_{\text{LSP}} = V(\epsilon, N) + \frac{1}{w} \sum_{k=1}^{w-1} (\mathbf{c}_t - \mathbf{c}_{t+k})^2$. An implicit assumption motives this method is that $\mathbf{c}_t$ (or D_t) in the stream does not fluctuate too much. Therefore, for streams with few changes, LSP may work better by saving up privacy budgets; otherwise, the estimation error on those skipped timestamps may become excessively large.

Considering the general non-deterministic sparsity in data streams, both LBU and LSP can not achieve better utility in general cases.

5.3 Adaptive Budget Division Methods

In this subsection, we propose two adaptive methods by constructing a unified distortion analysis under LDP.

BD/BA in the centralized setting [27] inspire us that higher utility can be achieved by adaptively allocating privacy budget in data streams. As summarized in Section 3.2, BD/BA compares the dissimilarity *dis* in aggregate statistics $\mathbf{c}_t$ with the potential publication error *err* at each time to adaptively choose between publication and approximation. However, in the local setting, since the central server cannot observe individuals' reports or directly obtain the true $\mathbf{c}_t$, the design of such LDP solutions is challenging. In particular, it is infeasible to accomplish the private dissimilarity calculation or data publication by adding noise over the true statistics, but we need to use FO protocols to do so. However, with FO protocols, it remains unclear how to model the *dissimilarity dis* and *publication error err* under LDP for empirically optimal strategy determination.

*5.3.1 **Private dissimilarity estimation**.* To address the above challenges, we first redefine the dissimilarity measure dis^* as the square error between the true statistics $\mathbf{c}_t$ of current timestamp and the previous release $\mathbf{r}_l$, i.e.,

$$dis^* = \frac{1}{d} \sum_{k=1}^{d} (\mathbf{c}_t[k] - \mathbf{r}_l[k])^2 \tag{3}$$

Then, in $\mathcal{M}_{t,1}$, we aim to obtain the dissimilarity dis^* privately, i.e., from users' LDP perturbed data using the dissimilarity budget $\epsilon_{t,1}$.

THEOREM 5.2. *Let $\bar{\mathbf{c}}_{t,1}$ denote the unbiased estimate of $\mathbf{c}_t$ from an ϵ-LDP frequency count over the perturbed data in $\mathcal{M}_{t,1}$. Then, the following dissimilarity measure*

$$dis = \frac{1}{d} \sum_{k=1}^{d} (\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2 - \frac{1}{d} \sum_{k=1}^{d} Var(\bar{\mathbf{c}}_{t,1}[k]). \tag{4}$$

is ϵ-LDP and an unbiased estimation of dis^* in Eq. (3).

PROOF. See Appendix A.2. □

Therefore, the dissimilarity can be calculated from $\bar{\mathbf{c}}_{t,1}$ while satisfying $\epsilon_{t,1}$-LDP. In the left term $\frac{1}{d} \sum_{k=1}^{d} (\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2$ of Eq (4), $\bar{\mathbf{c}}_{t,1}$ is obtained from FO while $\mathbf{r}_l$ is publicly known. The right term $\frac{1}{d} \sum_{k=1}^{d} Var(\bar{\mathbf{c}}_{t,1}[k])$ denoted as $V(\epsilon_{t,1}, N)$, can be calculated based on the population N and the dissimilarity budget $\epsilon_{t,1}$.

*5.3.2 **Private strategy determination**.* The key to choose the strategy of approximation or publication is to compare the dissimilarity (i.e., the potential *approximation error*) with the potential *publication error*. Considering that LDP protocols (e.g., GRR) are

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

different from CDP mechanism, the LDP-based publication error should also be re-formulated.

Here, in the LDP setting, considering that dis in Eq (3) is an L_2 distance measure, we propose to use Mean Square Error (MSE) to measure the publication error, denoted as err. Suppose $\bar{c}_{t,2}$ as the histogram estimated via FO protocol (e.g., GRR), the estimation error can be measured as

$$err = \frac{1}{d} \sum_{k=1}^{d} (\bar{c}_{t,2}[k] - c_t[k])^2 \qquad (5)$$

Since $\bar{c}_{t,2}$ is an unbiased estimation of c_t, i.e., $\mathbb{E}(\bar{c}_{t,2}) = c_t$,

$$err = \frac{1}{d} \sum_{k=1}^{d} \mathrm{Var}(\bar{c}_{t,2}[k]) \qquad (6)$$

which denotes as $V(\epsilon_{t,2}, N)$ and can be calculated from the user population N and the *publication budget* $\epsilon_{t,2}$. And, taking GRR as the LDP FO, it can be written as

$$err = \frac{1}{d} \sum_{k=1}^{d} \mathrm{Var}(\bar{c}_{t,2}[k]) = \frac{d - 2 + e^{\epsilon_{t,2}}}{N(e^{\epsilon_{t,2}} - 1)^2} + \frac{d - 2}{N(e^{\epsilon_{t,2}} - 1)}.$$

It is worth noting that, err is independent of f_v in Eq. (2).

Based on above formulations, an empirically optimal strategy at current timestamp t can be determined as follows.

- If $dis < err$, the approximation strategy is chosen. For example, the server can directly publish the last released value without consuming the publication budget $\epsilon_{t,2}$.
- Otherwise, the perturbation strategy is chosen. Each user reports value via a LDP FO using the publication budget $\epsilon_{t,2}$ to the server, who releases a freshly estimated statistics.

5.3.3 Privacy budget allocation. From the high level, we evenly divide the entire budget in a time window, ϵ, for two components: private dissimilarity estimation and private strategy determination. That is to say, the entire dissimilarity budget and publication budget, in a time window is $\sum_{i=t-w+1}^{t} \epsilon_{i,1} = \sum_{i=t-w+1}^{t} \epsilon_{i,2} = \epsilon/2$. In the private dissimilarity estimation, the dissimilarity budget is divided evenly to each timestamp in the time window, i.e., $\epsilon_{i,1} = 2\epsilon/w$. However, we aim to invest the publication budget economically to the timestamps, which leads to two different methods, **LDP budget distribution** (LBD) and **LDP budget absorption** (LBA).

In **LBD**, the publication budget is distributed in an exponentially decreasing way to the timestamps where a publication to occur. Algorithm 1 gives the details. For each time window, the entire budget ϵ is evenly divided into $\epsilon/2$ as dissimilarity budget and $\epsilon/2$ as publication budget, respectively. In sub mechanism $\mathcal{M}_{t,1}$, $\epsilon/2$ dissimilarity budget is uniformly distributed to each timestamp (Line 3). Then, all users apply an FO with the budget to report their data, which can be used to estimate a dissimilarity error dis (Lines 4-6). In sub mechanism $\mathcal{M}_{t,2}$, the remaining publication budget at current time is calculated first (Line 7). Then, half of it is pre-assigned as the potential publication budget, which is used to estimate the potential publication error err (Lines 8-9). By comparing dis and err, a strategy will be chosen between publication and approximation. If publication is chosen, the potential publication budget is truly used to get a fresh publication $\bar{c}_{t,2}$ (Lines 11-13). Otherwise, the last

release is published as an approximation. At this time, the potential publication budget is not truly used and thus reset as 0 (Line 15).

Algorithm 1: LDP Budget Distribution (LBD)

Input: Total privacy budget ϵ, window size w
Output: Released statistics $R_t = (r_1, r_2, \ldots, r_t, \ldots)$

1   Initialize $r_0 = \langle 0, \ldots, 0 \rangle^d$;
2   **for** *each timestamp t* **do**
    // Sub Mechanism $\mathcal{M}_{t,1}$:
3   Set dissimilarity budget $\epsilon_{t,1} = \epsilon/(2w)$;
4      $\overline{D}_{t,1} \leftarrow$ All Users report via an FO with privacy budget $\epsilon_{t,1}$;
5   Estimate $\bar{c}_{t,1} \leftarrow \text{FO}(\overline{D}_{t,1}, \epsilon_{t,1})$;
6   Calculate $dis = \frac{1}{d} \sum_{k=1}^{d} (\bar{c}_{t,1}[k] - r_{t-1}[k])^2 - \frac{1}{d} \sum_{k=1}^{d} \mathrm{Var}(\bar{c}_{t,1}[k])$;
    // Sub Mechanism $\mathcal{M}_{t,2}$:
7   Calculate remaining publication budget $\epsilon_{rm} = \epsilon/2 - \sum_{i=t-w+1}^{t-1} \epsilon_{i,2}$;
8   Set potential publication budget $\epsilon_{t,2} = \epsilon_{rm}/2$;
9   Calculate potential publication error err by Eq. (2);
10   **if** $dis > err$ **then**
        // Publication Strategy
11         $\overline{D}_{t,2} \leftarrow$ All Users report via an FO with budget $\epsilon_{t,2}$;
12   Estimate $\bar{c}_{t,2} \leftarrow \text{FO}(\overline{D}_{t,2}, \epsilon_{t,2})$;
13   **return** $r_t = \bar{c}_{t,2}$;
14   **else**
        // Approximation Strategy
15   **return** $r_t = r_{t-1}$; set $\epsilon_{t,2} = 0$.
16   **end**
17   **end**

In **LBA**, the publication budget is uniformly allocated budget at all timestamps then the unused budget is absorbed at the timestamps where publication is chosen. Algorithm 2 gives the details. Similarly, in each time window of size w, the entire budget ϵ is evenly divided into $\epsilon/2$ as dissimilarity budget and $\epsilon/2$ as publication budget, respectively. In sub mechanism $\mathcal{M}_{t,1}$, the process is identical to that of LBD. In sub mechanism $\mathcal{M}_{t,2}$, the number of timestamps to be nullified t_N is calculated based on the used publication budget at the publication timestamp l, and then skipped with approximation (Lines 4-6). Then, the number of timestamps to be absorbed, and potential publication budget can be calculated to derive the potential publication error err (Lines 8-10). In the next, by comparing err and dis, an empirically optimal strategy is chosen between publication and approximation (Lines 11-16).

5.4 Analysis

5.4.1 Privacy Analysis. Both LBD and LBA satisfy w-event LDP.

THEOREM 5.3. *LBD and LBA satisfy w-event LDP for each user.*

PROOF. See Appendix A.3. □

5.4.2 Utility Analysis. For simplicity, in both LBD and LBA, we assume there are $m < w$ publications occur at the timestamps $p_1, p_2, \ldots, p_m$ in the window of size w. Besides, no budget is recycled from past timestamps outside the window, and each publication approximates the same number of skipped/nullified publications. Similar to the analysis of LSP, at any timestamp t, if publication occurs, then the MSE of the release r_t is $\text{MSE}_{\text{pub}} = \mathrm{Var}[r_t]$; if approximation is chosen, its MSE equals to the sum of the variance of last release at timestamp l (i.e., $\mathrm{Var}[r_l]$), and the variance of the true statisics at the current timestamp t from that at timestamp l (i.e., $(c_t - c_l)^2$), i.e., $\text{MSE}_{\text{apr}} = \mathrm{Var}[r_l] + (c_t - c_l)^2$. Then we express

---

Algorithm 2: LDP Budget Absorption (LBA)

Input: Total privacy budget ϵ, window size w
Output: Released statistics $R_t = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_t, \ldots)$

---

1   Initialize $\mathbf{r}_0 = \langle 0, \ldots, 0 \rangle^d$, last publication timestamp $l = 0$, and $\epsilon_{l,2} = 0$;
2   **for** *each timestamp t* **do**
      // Sub Mechanism $\mathcal{M}_{t,1}$:
3   Same as Lines 3-6 in **Algorithm 1**

      // Sub Mechanism $\mathcal{M}_{t,2}$:
4   Calculate timestamps to be nullified $t_N = \frac{\epsilon_{l,2}}{\epsilon/(2w)} - 1$;
5   **if** $t - l \le t_N$ **then**
6 | **return** $\mathbf{r}_t = \mathbf{r}_{t-1}$;
7   **else**
8   Calculate timestamps can be absorbed $t_A = t - (l + t_N)$;
9   Set potential publication budget $\epsilon_{t,2} = \epsilon/(2w) \cdot \min(t_A, w)$;
10   Calculate potential publication error err by Eq. (2);
11   **if** $dis > err$ **then**
            // Perturbation Strategy
12 | $\overline{D}_{t,1} \leftarrow$ All Users report via an FO with budget $\epsilon_{t,2}$;
13 | Estimate $\overline{\mathbf{c}}_{t,2} \leftarrow$ FO$(\overline{D}_{t,2}, \epsilon_{t,2})$;
14 | **return** $\mathbf{r}_t = \overline{\mathbf{c}}_{t,2}$, set $l = t$;
15   **else**
            // Approximation Strategy
16 | **return** $\mathbf{r}_t = \mathbf{r}_{t-1}$; set $\epsilon_{t,2} = 0$.
17   **end**
18   **end**
19   **end**

MSE in a whole time window as follows

$$\text{MSE}_{LBD/LBA} = \frac{1}{w} \left[\frac{w}{m} \sum_{i=1}^{m} \text{Var}[\mathbf{r}_{p_i}] + \sum_{i=1}^{m} \sum_{t=p_i}^{p_{i+1}-1} (\mathbf{c}_t - \mathbf{c}_{p_i})^2 \right] \quad (7)$$

where the second term in the bracket solely dependes on the underlying data and shows the data-dependent characteristics of LBD and LBA. In the following, we analyze the left term in the bracket.

In LBD, since the budget is distributed to the m publications in an exponentially decreasing way, the budget sequence of $\epsilon_{t,2}$ is then $\epsilon/4, \epsilon/8, \ldots, \epsilon/2^{m+1}$. There is

$$\sum_{i=1}^{m} \text{Var}_{\text{LBD}}[\mathbf{r}_{p_i}] = \sum_{i=1}^{m} V(\epsilon/2^{i+1}), N) < m \cdot V(\epsilon/2^{m+1}, N) \quad (8)$$

where $V(\epsilon, n)$ denotes the estimation variance of an FO from n users' LDP data using budget ϵ. As we can see, with the increase of m, the error of LBD would increase dramatically.

In LBA, due to m publications in the assumption, there are $w - m$ approximations. Since each publication approximates the same number of skipped/nullified publications, there are $\frac{w-m}{2 \cdot m}$ skipped (whose budgets are absorbed) and $\frac{w-m}{2 \cdot m}$ nullified publications in average. Then, each publication receives those skipped budget $\frac{(\frac{w-m}{2 \cdot m}+1) \cdot \epsilon}{2 \cdot w} = \frac{(w+m) \cdot \epsilon}{4 \cdot w \cdot m}$ and incurs MSE of $V(\frac{w+m}{4 \cdot w \cdot m} \cdot \epsilon, N)$.

$$\sum_{i=1}^{m} \text{Var}_{\text{LBA}}[\mathbf{r}_{p_i}] = m \cdot V(\frac{w+m}{4 \cdot w \cdot m} \cdot \epsilon, N) \quad (9)$$

Compared to LBD, LBA's error increases with m more mildly.

5.4.3 Communication Analysis. Given an FO, the LDP perturbed data in different methods has the same packet size in each communication. So, the communication cost can be simply measured by the average communication times of each user per timestamp, or *communication frequency per user* (CFPU).
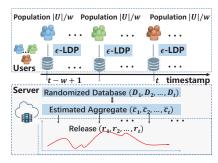

Figure 3: Illustration of population division framework

In both LBD/LBA, each user have to communication twice with the server at publication timestamps and only once at approximation timestamps. Therefore, when there are m publications in a window of size w, the average CFPU is $(2m + w - m)/w = 1 + m/w$.

6 POPULATION DIVISION-BASED METHODS

In this section, we first present a novel idea of population division for LDP over data streams. Then, we propose to transform the above budget division-based methods to the population division-based methods for better utility and communication efficiency.

6.1 Basic idea

The budget division framework provides a feasible solution to infinite streaming data collection with LDP. However, we notice that, data utility in LDP scenarios is much more sensitive to privacy budget than that in CDP. Recall that $V(\epsilon, n)$ denotes the FO estimation variance from n users with privacy budget ϵ. According to Eq (2), with fixed n, $V(\epsilon, n)$ is $O((e^\epsilon - 1)^{-1})$ in terms of average budget ϵ. It increases sharply as the budget assigned to each timestamp becomes small. Recently, several previous studies [38, 41] indicate that it can achieve much smaller overall error by partitioning users into groups and using the entire privacy budget in each group. With the fixed ϵ, $V(\epsilon, n)$ is $O(n^{-1})$ in terms of average user population n, which increases much mildly as n becomes small. Therefore, we adopt this idea in streaming data collection with LDP. Intuitively, a baseline methods can be derived to achieve w-event LDP.

LDP Population Uniform Method (LPU). At the beginning, the central server uniformly assign the uses into w disjoint groups, each with roughly N/w users [4]. At each timestamp, it requests a group of users that have never been requested before to report their value. In a window with w timestamps, each group of users will only report once with the entire budget ϵ. And after $w - 1$ timestamps, each group users will be requested and report again for the new sliding window. In this case, any user does not report in each sliding window more than once, thus spending no more than ϵ-LDP budget. Hence, w-event LDP is guaranteed for each user. Fig. 3 illustrates the population division methodology.

THEOREM 6.1. *Given the same FO protocol GRR or OUE, the MSE of LPU is smaller than that of LBU, i.e., $MSE_{LPU} < MSE_{LBU}$.*

PROOF. See Appendix A.4.   □

---

[4]Precisely, if $N \mod w \ne 0$, it may be $\lfloor N/w \rfloor$ for some groups or $\lfloor N/w \rfloor + 1$ for the rests. For simplicity, we assume N/w for each.

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

Note that, since only a portion of users participate in reporting, the population division methodology can also greatly reduce the communication cost. In LPU, the number of users upload perturbed data at each timestamp is only $1/w$ of the whole population in average. Therefore, the communication cost is $1/w$ of that in LBU.

LSP in Section 5.2.2 can be also seen as a population division method. Particularly, all users are regarded as to be divided into w groups, in which, one group has the whole population and the rests have zero users. Then, that one group of users are assigned to report at a single timestamp within the window using the entire budget ϵ while no users for the next $w - 1$ timestamps.

However, both LPU and LSP cannot be adaptive to streams with unknown fluctuations, which still limits their utility.

6.2 Adaptive Population Division Methods

LBD/LBA provides a reference framework that improves the utility of baseline methods via adaptively assigning privacy budget according to the non-deterministic sparsity in data streams. In the following, we present two adaptive population division methods LPU/LPA, which migrates this idea to the population division framework to significantly enhance the utility.

6.2.1 Overview. For better analogy to LBD/LBA, we still introduce the population division based methods with two sub mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$. We first evenly partition the whole population U of size N into *dissimilarity users* U_1 of size $|U_1|$ for $\mathcal{M}_1$ and *publication users* U_2 of size $|U_2|$ for $\mathcal{M}_2$, each with $\lfloor N/2 \rfloor$ users. Similarly, $\mathcal{M}_1$ mainly achieves private dissimilarity calculation. Differently, under the population division framework, $\mathcal{M}_2$ accomplishes private strategies determination and participant users allocation.

Private dissimilarity calculation: Section 5.3.1 defines the dissimilarity measure *dis* in the LDP setting. In $\mathcal{M}_1$, at each timestamp t, we still aim to estimate the dissimilarity $dis^* = \frac{1}{d}\sum_{k=1}^{d}(\mathbf{c}_t[k] - \mathbf{r}_l[k])^2$ based on Eq. (4). Similarly, we have to first obtain an unbiased estimation $\bar{\mathbf{c}}_t$ through an FO at timestamp t. Differently, under the population division methodology, it can only be obtained from the LDP protected data (with privacy budget ϵ) of dissimilarity users $U_{t,1}$ at timestamp t. We here partition the $\lfloor N/2 \rfloor$ dissimilarity users over the w timestamps evenly. That is to say, at each timestamp t, $|U_{t,1}| = \lfloor N/(2w) \rfloor$ dissimilarity users report their value via an FO using the entire budget ϵ.

Private strategy determination: In $\mathcal{M}_2$, the estimated dissimilarity *dis* output by $\mathcal{M}_1$ (i.e., approximation error) and the potential publication error *err* are compared to empirically choose a better strategy (i.e., with smaller error) from approximation and publication. The publication error *err* equals to the estimation variance $V(\epsilon, |U_{t,2}|)$, can be calculated based on the available privacy budget $\epsilon_{t,2}$ as well as the number of potential publication users $|U_{t,2}|$, e.g., according to Eq. (2) in GRR. Under the population division framework, the budget is fixed as a constant ϵ and the publication error is determined by $|U_{t,2}|$, which is dynamically assigned in a sliding window. The more publication users, the less the publication error *err* is. However, since any user only participates once in a window, the availability of publication users $U_{t,2}$ in each timestamp t in the sliding window is limited and should be carefully assigned.

Participant users allocation: In $\mathcal{M}_2$, with the above transition from budget division to population division, the adaptive budget

---

Algorithm 3: LDP Population Distribution (LPD)

Input: Total population U of size $N = |U|$, privacy budget ϵ, window size w
Output: Released statistics $R_t = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_t, \ldots)$

1 Initialize available userset $U_A = U$, and $\mathbf{r}_0 = \langle 0, \ldots, 0 \rangle$;
2 **for** *each timestamp t* **do**
      // Sub Mechanism $\mathcal{M}_{t,1}$:
3   Sample users $U_{t,1}$ from U_A with the size of $\lfloor N/(2w) \rfloor$, remove $U_{t,1}$ from U_A, i.e., $U_A = U_A \setminus U_{t,1}$;
4     $\overline{D}_{t,1} \leftarrow$ Users in $U_{t,1}$ report via an FO with privacy budget ϵ;
5   Estimate $\bar{\mathbf{c}}_{t,1} \leftarrow \text{FO}(\overline{D}_{t,1}, \epsilon)$;
6   Calculate $dis = \frac{1}{d}\sum_{k=1}^{d}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_{t-1}[k])^2 - \frac{1}{d}\sum_{k=1}^{d}\text{Var}(\bar{\mathbf{c}}_{t,1}[k])$;
      // Sub Mechanism $\mathcal{M}_{t,2}$:
7   Calculate remaining population size $N_{rm} = N/2 - \sum_{i=t-w+1}^{t-1}|U_{i,2}|$;
8   Set number of potential publication users $N_{pp} = N_{rm}/2$;
9   Calculate potential publication error err by Eq. (2);
10   **if** *dis > err and $N_{pp} \geq u_{min}$* **then**
          // Publication Strategy
11   Sample a userset $U_{t,2}$ from U_A with the size of $|U_{t,2}| = N_{pp}$, $U_A = U_A \setminus U_{t,2}$;
12        $\overline{D}_{t,2} \leftarrow$ Users in $U_{t,2}$ report via an FO with budget ϵ;
13   Estimate $\bar{\mathbf{c}}_{t,2} \leftarrow \text{FO}(\overline{D}_{t,2}, \epsilon)$;
14   **return** $\mathbf{r}_t = \bar{\mathbf{c}}_{t,2}$;
15   **else**
          // Approximation Strategy
16   **return** $\mathbf{r}_t = \mathbf{r}_{t-1}$.
17   **end**
18   **if** $t \geq w$ **then**
          // Recycling Users
19        $U_A = U_A \cup U_{t-w+1,1} \cup U_{t-w+1,2}$.
20   **end**
21 **end**

---

allocation schemes, i.e., budget distribution (in LBD) and budget absorption (LBA) can be transferred for assigning the number of publication users $|U_{t,2}|$ under the population division framework. This also leads to two adaptive population division methods: *population distribution LPD* and *population absorption LPA*.

In the following, we present the details of LPD and LPA.

6.2.2 LDP Population Distribution Method (LPD). Algorithm 3 presents the details of LPD. Firstly, for calculation of dissimilarity *dis* in $\mathcal{M}_1$ (Lines 3-6), the dissimilarity population $|U_1|$ is uniformly divided into disjoint groups of dissimilarity users $U_{t,1}$ at each timestamp, i.e., $|U_{t,1}| = \lfloor N/(2w) \rfloor$. Next in $\mathcal{M}_2$, the remaining number of publication users N_{rm} is calculated by removing the already used publication users in the last $w - 1$ timestamps from the total number of publication users $N/2$ (Line 7). Then, the number of potential publication users is set as $N_{pp} = N_{rm}/2$ to calculate a potential publication error *err* (Lines 8-9). By comparing *err* with *dis*, the publication or approximation strategy is decided then (Lines 10-17). In case of too many publications and N_{pp} decays too quickly to have no available user, a threshold u_{min} (e.g., $u_{min} = 1$) is set (Line 10). Once publication is chosen, N_{pp} new users will be sampled as actual publication users $U_{t,2}$ from U_A to accomplish publication (Lines 11-14). Otherwise, $\mathbf{r}_t$ is approximated by $\mathbf{r}_{t-1}$, without using N_{pp} users (Line 16). Finally, both the used dissimilarity users and publication users (may be *null*) at timestamp $t - w + 1$, which is falling outside of the next active window, are recycled as available users U_A (Line 19). The recycling process ensures each user can contribute again after w timestamps while guaranteeing no users participate more than once. Detail description of LPD can be referred to Appendix B.1.

6.2.3 **LDP Population Absorption Method (LPA).** Algorithm 4 presents the details of LPA. The private dissimilarity calculation process of $\mathcal{M}_{t,1}$ in LPA is the same as that in LPD. In $\mathcal{M}_{t,2}$, the basic idea is to uniformly allocated users at all timestamps then the unused publication users is absorbed at the timestamps where publication is chosen. Once a publication occurs at time l, the same number of users must be skipped from the succeeding timestamps to ensure available users within the active sliding window. So, the number of timestamps to be nullified t_N is first calculated based on the number of publication users at timestamp l, and thus skipped with approximation (Lines 4-6). After that, based on the timestamps can be absorbed, the number of potential publication users N_{pp} is calculated at each time t, which can further derive the potential publication error *err* (Lines 8-10). By comparing *err* with *dis*, $\mathcal{M}_{t,2}$ decides whether to freshly publish with the potential publication users (Lines 11-15) or continues to approximate with the last release (Lines 16-18). Similarly, both the used dissimilarity users and publication users at timestamp $t - w + 1$ are finally recycled as available users U_A (Lines 20-22). Detail description of LPA can be referred to Appendix B.2.

Algorithm 4: LDP Population Absorption (LPA)

Input: Total population U of size $N = |U|$, privacy budget ϵ, window size w
Output: Released statistics $R_t = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_t, \ldots)$

1 Initialize available userset $U_A = U$, and $\mathbf{r}_0 = \langle 0, \ldots, 0 \rangle$, last publication timestamp $l = 0$, and $\epsilon_{l,2} = 0$;
2 **for** *each time t* **do**
  // Sub Mechanism $\mathcal{M}_{t,1}$:
3   Same as Lines 3-6 in **Algorithm 3**
  // Sub Mechanism $\mathcal{M}_{t,2}$:
4   Calculate timestamps to be nullified $t_N = \frac{|U_{l,2}|}{\lfloor N/(2w) \rfloor} - 1$;
5   **if** $t - l \le t_N$ **then**
6   **return** $\mathbf{r}_t = \mathbf{r}_{t-1}$;
7   **else**
8   Calculate timestamps can be absorbed $t_A = t - (l + t_N)$;
9   Set number of potential publication users
     $N_{pp} = \lfloor N/(2w) \rfloor \cdot \min(t_A, w)$;
10   Calculate potential publication error *err* by Eq. (2);
11   **if** *dis* > *err* **then**
      // Publication Strategy
12   Sample a userset $U_{t,2}$ from U_A with the size of $|U_{t,2}| = N_{pp}$,
      $U_A = U_A \setminus U_{t,2}$;
13       $\overline{D}_{t,1} \leftarrow$ Users in $U_{t,2}$ report via an FO with budget ϵ;
14   Estimate $\overline{c}_{t,2} \leftarrow$ FO$(\overline{D}_{t,2}, \epsilon)$;
15   **return** $\mathbf{r}_t = \overline{c}_{t,2}$, set $l = t$;
16   **else**
      // Approximation Strategy
17   **return** $\mathbf{r}_t = \mathbf{r}_{t-1}$.
18   **end**
19   **end**
20   **if** $t \ge w$ **then**
    // Recycling Users
21     $U_A = U_A \cup U_{t-w+1,1} \cup U_{t-w+1,2}$;
22   **end**
23 **end**

6.3 Analysis

6.3.1 Privacy Analysis. LPD and LPA satisfy w-event LDP because each user reports to the server at most once in a time window of size w and each report goes through an FO with ϵ-LDP.

Theorem 6.2. *LPD and LPA satisfies w-event LDP for each user.*
Proof. See Appendix A.5. $\square$

6.3.2 Utility Analysis. With the same assumptions, similar MSE expression can be obtained as Eq. (7) in Section 5.4.2. Then, under the population division framework, in LPD, since the population is distributed to the m publications in an exponentially decreasing way, the population alloction sequence of $N_{t,2}$ is then $N/4, N/8, \ldots, N/2^{m+1}$. There is

$$\sum_{i=1}^{m} \text{Var}_{\text{LPD}}[\mathbf{r}_{p_i}] = \sum_{i=1}^{m} V(\epsilon, N/2^{i+1}) \tag{10}$$

Therefore, the error of LPD would still increase with m. However, according to Lemma 6.1, we can conclude that $V(\epsilon, N/2^{m+1}) < V(\epsilon/2^{m+1}, N)$. That is to say, LPD can achieve less error than LBD. Similarly, in LPA,

$$\sum_{i=1}^{m} \text{Var}_{\text{UA}}[\mathbf{r}_{p_i}] = m \cdot V(\epsilon, \frac{w+m}{4 \cdot w \cdot m} \cdot N) \tag{11}$$

which is smaller than $m \cdot V(\frac{w+m}{4 \cdot w \cdot m} \cdot \epsilon, N)$ in Eq. (9) of the budget absorption method LBA, given the same assumptions.

6.3.3 Communication Analysis. In LPD, all m publications in a window need $\sum_{i=1}^{m} (N/2^{i+1} + N/(2w)) = (\frac{1-(1/2)^m}{2} + \frac{m}{2w}) \cdot N$ users to communicate and the rest $w - m$ approximations need $\frac{w-m}{2w} \cdot N$ users. Therefore, the average CFPU is $\frac{1}{w \cdot N}[(\frac{1-(1/2)^m}{2} + \frac{m}{2w}) \cdot N + \frac{w-m}{2w} \cdot N] = \frac{1}{w} - \frac{1}{w \cdot 2^{m+1}}$.

In LPA, all m publications in a window need $m \cdot (\frac{w+m}{4 \cdot w \cdot m} \cdot N + N/(2w))$ users to communicate and the rest timestamps need $(w - m) \cdot N/(2w)$ users. Therefore, the average CFPU is $\frac{1}{w \cdot N}[m \cdot (\frac{w+m}{4 \cdot w \cdot m} \cdot N + N/(2w)) + (w - m) \cdot N/(2w)] = \frac{1}{2w} + \frac{w+m}{4w^2}$.

6.4 Discussion

In this subsection, we briefly discuss the differences of our methods from existing ones, open problems, and future directions.

Remark 1: LDP methods LBD/LBA and LPD/LPA proposed above are inspired from BD/BA in the centralized setting of DP. But they are different in many aspects, including but not limited to, the information that the server can access to and operations it can perform, the perturbation mechanisms, the measurement of error in perturbation and approximation, and the population division framework instead of budget division.

Remark 2: The LPD and LPA methods proposed above can be applied to a large spectrum of IoTs scenarios that massive reliable devices persistently monitor the environment or events, such as smart metering systems, security and video cameras. Note that, in mobile scenarios, the number of joining devices may be time-varying, e.g., new users may join in and churn randomly, which may make this framework complicated.

Remark 3: Besides LPD and LPA, the population division-based LDP framework can be easily applied and extended to other state-of-the-art DP methods for streams (including user-level DP), such as FAST [20], PeGaSus [6] and RescueDP [37], which may achieve better utility but the techniques in these methods often require complicated parameters tuning with extra effort.

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

7 PERFORMANCE EVALUATION

In this section, we conducted extensive experiments to evaluate the performance of our proposed algorithms.

7.1 Experimental Setup

*7.1.1 **Synthetic Datasets**.* We synthesized binary streaming datasets with different sequence models. Given a probability process model $p_t = f(t)$, the length of time T, and user population N, we first generated a probability sequence $(p_1, p_2, \ldots, p_T)$ with T timestamps. Then, at each timestamp t, we randomly chose a portion of p_t users from the total N users to set their true report value v_t^j as 1, and set the rest as 0. The following typical sequence patterns were used.

- LNS is a linear process $p_t = p_{t-1} + \mathcal{N}(0, Q)$, where $p_0 = 0.05$ and $\mathcal{N}(0, Q)$ is Gaussian noise with the standard variance $\sqrt{Q} = 0.0025$.
- Sin is a sequence composed by a sine curve $p_t = A\sin(bt) + h$ with $A = 0.05$, $b = 0.01$ and $h = 0.075$.
- Log is a series with the logistic model $p_t = A/(1 + e^{-bt})$ where $A = 0.25$ and $b = 0.01$.

Without specifying, we used above models and default parameters to generate synthetic binary streams with 800 timestamps of $200,000$ users. To demonstrate the varying fluctuations, we set N fixed but changed the parameters Q in LNS and b in Sin respectively to obtain different datasets. To demonstrate the varying populations, we used the probability sequences generated with the default parameters above, but performed different number of sampling processes to obtain datasets with different population N.

*7.1.2 **Real-world Datasets**.* To evaluate the practical performance of algorithms, the following three real-world datasets with non-binary values were also used.

- Taxi[5] contains the real-time trajectories of 10,357 taxis during the period of Feb. 2 to Feb. 8, 2008 within Beijing. We obtained $N = 10,357$ data streams for each taxi by extracting $T = 886$ timestamps (each at 10-minute level) and partitioning area into 5 grids, i.e., $d = 5$.
- Foursquare[6] includes 33,278,683 check-ins of Foursquare users from Apr. 2012 to Sep. 2013, where each record includes time, place and user ID. We transformed it into $N = 265,149$ data streams with the length of $T = 447$ timestamps, each records a user's check-in sequence over $d = 77$ countries.
- Taobao[7] contains the AD click logs of 1.14 million customers at Taobao.com. For simplicity, we first grouped the AD commodities into $d = 117$ categorizes. Then, we extracted all the $N = 1,023,154$ customers' click data streams, where each item corresponds to the categorize of the user's last click during each ten minutes in three consecutive days, i.e., $T = 432$ timestamps.

*7.1.3 **Compared Algorithms**.* We compared the following algorithms. All are implemented using Matlab 2020.

---

[5]https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/
[6]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[7]https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

Scheme Dimensions	Non-adaptive allocation		Adaptive allocation	
	Uniform	Sampling	Distribution	Absorption
Budget division	LBU	LSP	LBD (Algorithm 1)	LBA (Algorithm 2)
Population division	LPU		LPD (Algorithm 3)	LPA (Algorithm 4)

LBU (Sec. 5.2.1) and LPU (Sec. 6.1) are baseline methods that uniformly divide budget and population, respectively. LSP (Sec. 5.2.2 and 6.1) invests the entire budget and users at sampling timestamps with fixed interval. LBD/ LBA (Algs. 1, 2 in Sec. 5.3.3) and LPD/LPA (Algs. 3, 4 in Sec. 6.2.2 and 6.2.3) adaptively allocate the budget and population via two different schemes, respectively.

All experiments were conducted on a PC with an Intel Core i5-6300HQ 3.20GHz and 16GB memory.

*7.1.4 **Performance Metrics**.* We evaluated the performance of different algorithms in terms of data utility, event monitoring efficiency, and communication efficiency. The utility was measured as the *mean relative error* (MRE) between the released and true statistics. The event monitoring efficiency was measured as the ratio that, from perturbed reports, the server successfully detects extreme events, i.e., the statistics of which are greater than a given threshold. The communication efficiency is mainly compared by counting the *communication frequency per user* (CFPU).

7.2 Overall Utility

Fig. 4 shows the release accuracy of all compared w-event LDP methods on all synthetic and real-world datasets, with different privacy budget ϵ. These methods are categorized into budget division-based (LBU, LBD, LBA) and population division-based methods (LSP, LPU, LPD, LPA). Overall, the error of all methods decreases with ϵ, which shows the tradeoff between data utility and privacy. Besides, the population division-based methods significantly outperform budget-division ones with much smaller MRE. This is because LDP is more sensitive to the budget division than population division. LBD/LBA generally shows smaller error than the straightforward method of LBU. This is because LBD/LBA can utilize temporal correlations in data streams to reduce the privacy budget consumption rate. The advantage of LPD/LPA is clearer as noise does not increase dramatically when the population is divided. Although LSP achieves even smaller error than LPD/LPA, its performance varies dramatically across datasets.

Fig. 5 shows the release accuracy of all compared w-event LDP methods, with different window size w, on all datasets. In general, the MRE of all methods increases with w, since less privacy budget or users will be allocated to each timestamp. For budget division methods, with the increasing w, LBD distributes budget in an exponentially decaying way and allocates very small budget for the newest timestamp in the window, thus causing quite large estimation error. For example, when w is large, LBD may even have larger MRE than LBU. LBA to avoid this issue and well adapt to data fluctuations. For population division methods, despite the similar trends, LPD manages to achieve smaller MRE than baseline LPU while LPA performs even much better. Besides, as shown, with large w, LPD/LPA gains more prominent advantages.

(a) LNS, $w = 20$

(b) Sin, $w = 20$

(c) Log, $w = 20$

(d) Taxi, $w = 20$

(e) Foursquare, $w = 20$

(f) Taobao, $w = 20$

Figure 4: Data utility with different ϵ

(a) LNS, $\epsilon = 1$

(b) Sin, $\epsilon = 1$

(c) Log, $\epsilon = 1$

(d) Taxi, $\epsilon = 1$

(e) Foursquare, $\epsilon = 1$

(f) Taobao, $\epsilon = 1$
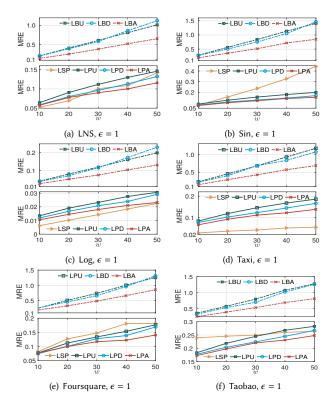
Figure 5: Data utility with different w

7.3 Impact of Dataset Parameters

To demonstrate the impact of dataset (population size and data fluctuations) on utility performance, we changed population N, and noise variance Q and period parameter b in the synthetic datasets LNS and Sin respectively. Note that, while varying the population size N, we kept the frequency fixed.

Figs. 6(a) and 6(b) show the MRE of compared methods on LNS and Sin with respect to different population size N. As shown, the MRE of all methods decreases with N. This is because that, enlarging N while fixing the frequency leads to better estimation accuracy. Figs. 6(c) and 6(d) show the MRE on LNS with different variance Q and Sin with different period parameter b, respectively. On LNS, the MRE of all methods increases with Q, which measures the fluctuation in streams. This result verifies that these methods are data-dependent and perform better on streams with few changes. We can also see that budget division methods including LBU, LBD and LBA have much larger error than the population division methods. Although LSP manages to have the smallest error when the variance is small (i.e., $\sqrt{Q} = 0.001, 0.002$), it grows fast and surpasses LPD and LPA with the increase of Q. LPD and LPA induce much smaller error, and perform slightly worse than LPU when the variance is large. Note that, period parameter b also represents data fluctuations and larger b means larger fluctuations. Similar conclusions can be obtained from Sin that population division based LDP methods manage to achieve much higher utility, and LPD and LPA manage to further improve the utility on steady streams.
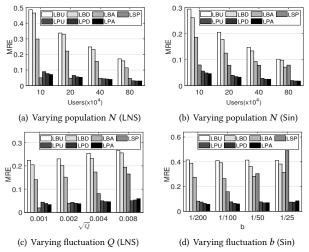

(a) Varying population N (LNS)

(b) Varying population N (Sin)

(c) Varying fluctuation Q (LNS)

(d) Varying fluctuation b (Sin)

Figure 6: Impact of dataset parameters ($\epsilon = 1$, $w = 30$)

7.4 Event Monitoring

The overall distance metric of MRE on the whole stream cannot reflect the estimate accuracy at individual timestamps. Instead, event monitoring is commonly used in data streams to detect whether the estimate at each timestamp is larger than a given threshold δ. Fig. 7 displays the ROC curves for detecting the above-threshold points on all six dataset. On synthetic binary datasets LNS, Sin, Log, δ was directly set as $0.75 \times (\max(\mathbf{c}) - \min(\mathbf{c})) + \min(\mathbf{c})$. On other three non-binary real-world datasets, we monitored the mean-value $\mathbf{c}_{\text{mean}}$ of the histogram $\mathbf{c}_t$ and δ was set as

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

$0.75 \times (\max(\mathbf{c}_{\mathrm{mean}}) - \min(\mathbf{c}_{\mathrm{mean}})) + \min(\mathbf{c}_{\mathrm{mean}})$. Overall, the population division methods generally perform better than the budget division method LBA, as they can achieve higher accuracy in above-threshold value detection. Despite varying on different datasets, LPD and LPA in general outperform the other three methods. Although LSP manages to have much smaller MRE in Figs. 4 and 5, it generally performs the worst on most datasets. This is because too many approximations are adopted in LSP, which hinders its efficiency in detecting real-time changes.
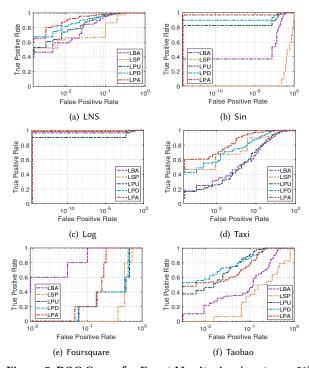

(a) LNS

(b) Sin

(c) Log

(d) Taxi

(e) Foursquare

(f) Taobao

Figure 7: ROC Curve for Event Monitoring ($\epsilon = 1$, $w = 50$)

7.5 Communication Efficiency

Fig. 8 compares the average communication frequency per user (CFPU) of different methods, with different parameters, on LNS.

Fig. 8(a) depicts the impact of population N on CFPU. For budget division methods, the average CFPU is above 1.0 since each user has to report at least once at each timestamp. For LBU, each user just reports once at each timestamp. For LBD and LBA, some users may report twice for both $\mathcal{M}_1$ and $\mathcal{M}_2$ at each timestamp. Differently, for population division methods, the CFPU is significantly less since only a small portion of users contribute data at each timestamp. In both LSP and LPU, each user only reports once in a window of $w = 20$ timestamps, with the average chance of $1/w = 0.05$ per timestamp. In LPD and LPA, users are adaptively divided and chosen to report according to data density of streams, therefore, the CFPU can be effectively reduced.

Fig. 8(b) shows CFPU with respect to data variance Q. Similarly, Q has no impact on data-independent methods LBU, LSP and LBU. However, with the increase of Q, data-dependent methods LBD/LBA and LPD/LPA have a larger CFPU, although it is not evident for

(a) CFPU wrt. N

(b) CFPU wrt. Q
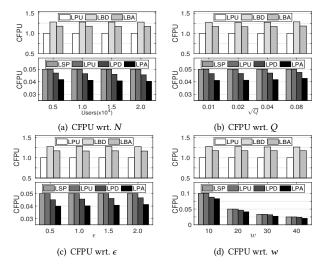
(c) CFPU wrt. ϵ

(d) CFPU wrt. w

Figure 8: CFPU with respect to different parameters (LNS)

LBD/LBA. The former is because that, with more fluctuations, data-dependent methods would increase publication frequency to adapt to stream changes. The latter is because that the larger noise in LBD/LBA overwhelms the impact of data fluctuations.

Fig. 8(c) presents the CFPU with respect to ϵ. For budget division methods, it generally keeps the same since a slight increase of ϵ has rather limited impact. However, the CFPU increases with ϵ in LPD and LPA. The reason is that, with more budget ϵ, the publication error would become smaller and more publications would be chosen in the adaptive methods.

Fig. 8(d) shows the CFPU, with respect to w. Among the budget division methods, LBU is data-independent and keeps unchanged. For LBD and LBA, the CFPU shows a slight decrease with w. This is because the publication noise increases with w and the approximation strategy is more favorable. Among the population division methods, besides data-dependent methods LPD and LPA, the CFPU in LSP and LPU also decreases with w because each user reports at each timestamp with a probability of $1/w$.

Table 2 further summarizes the CFPU of different methods over other datasets. As shown, the population division methods manage to have much less communication overhead. Data-adaptive methods LPD and LPA further reduce the communication cost via exploiting the sparsity in data streams. All above results are consistent with the analysis in Sections 5.4.3 and 6.3.3.

8 CONCLUSION

We propose LDP-IDS, a decentralized privacy-preserving scheme for infinite streaming data collection and analysis. We first formalize the definition of w-event LDP for infinite data streams. Then, based on the budget division methodology, we present several baseline methods that can satisfy w-event LDP for streaming data collection and analysis. Furthermore, we propose a novel framework of population division, which can achieve significant utility improvement and communication reduction for streaming data collection and analysis with LDP. Specifically, considering the non-deterministic sparsity in data streams, two data-adaptive methods are also presented to achieve further utility improvement. Through theoretical

Table 2: CFPU Comparison on All Datasets

$\epsilon = 1, w = 20$		Synthetic Datasets		Real-world Datasets		
		Sin	Log	Taxi	Foursquare	Taobao
B	LBU	1.0000	1.0000	1.0000	1.0000	1.0000
	LBD	1.2719	1.2671	1.2734	1.2733	1.2962
	LBA	1.1709	1.1687	1.1685	1.1775	1.1996
P	LSP	0.0500	0.0500	0.0500	0.0500	0.0500
	LPU	0.0500	0.0500	0.0500	0.0500	0.0500
	LPD	0.0457	0.0457	0.0461	0.0458	0.0467
	LPA	0.0404	0.0403	0.0406	0.0403	0.0418
$\epsilon = 2, w = 20$		Synthetic Datasets		Real-world Datasets		
		Sin	Log	Taxi	Foursquare	Taobao
B	LBU	1.0000	1.0000	1.0000	1.0000	1.0000
	LBD	1.2800	1.2823	1.2762	1.2692	1.3243
	LBA	1.1731	1.1737	1.1682	1.1704	1.2350
P	LSP	0.0500	0.0500	0.0500	0.0500	0.0500
	LPU	0.0500	0.0500	0.0500	0.0500	0.0500
	LPD	0.0466	0.0468	0.0475	0.0468	0.0475
	LPA	0.0414	0.0413	0.0425	0.0412	0.0434
$\epsilon = 2, w = 40$		Synthetic Datasets		Real-world Datasets		
		Sin	Log	Taxi	Foursquare	Taobao
B	LBU	1.0000	1.0000	1.0000	1.0000	1.0000
	LBD	1.2643	1.2575	1.2641	1.2487	1.2771
	LBA	1.1729	1.1676	1.1755	1.1670	1.2046
P	LSP	0.0250	0.0250	0.0250	0.0250	0.0250
	LPU	0.0250	0.0250	0.0250	0.0250	0.0250
	LPD	0.0242	0.0245	0.0244	0.0245	0.0245
	LPA	0.0206	0.0207	0.0210	0.0204	0.0214

*B and P refer to budget division and population division, respectively.

analysis and experiments with real-world datasets, we demonstrate the superiority of our LDP solutions against the budget division-based benchmark methods in terms of estimation accuracy, practical event monitoring efficiency and communication cost.

REFERENCES

[1] H. Arcolezi, J Couchot, B. Bouna, and X. Xiao. 2021. Longitudinal Collection and Analysis of Mobile Phone Data with Local Differential Privacy. *Privacy and Identity Management: 15th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Maribor, Slovenia, September 21–23, 2020, Revised Selected Papers* 619 (2021), 40.

[2] B. Balle, G. Barthe, and M. Gaboardi. 2018. Privacy amplification by subsampling: tight analyses via couplings and divergences. In *Proc. NeurIPS*. 6280–6290.

[3] E. Bao, Y. Yang, X. Xiao, and B. Ding. 2021. CGM: An Enhanced Mechanism for Streaming Data Collection with Local Differential Privacy. *Proc. of VLDB Endowment* 14, 11 (2021), 2258–2270.

[4] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft. 2013. Private decayed predicate sums on streams. In *Proc. ACM ICDT*. 284–295.

[5] T.-H. Hubert Chan, E. Shi, and D. Song. 2011. Private and Continual Release of Statistics. *ACM Trans. Inf. Syst. Secur.* 2010 (2011), 76.

[6] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau. 2017. PeGaSus: Data-Adaptive Differentially Private Stream Processing. *Proc. ACM CCS* (2017), 1375–1388.

[7] G. Cormode, T. Kulkarni, and D. Srivastava. 2018. Marginal release under local differential privacy. In *Proc. ACM SIGMOD*. 131–146.

[8] G. Cormode, T. Kulkarni, and D. Srivastava. 2019. Answering range queries under local differential privacy. *Proc. of VLDB Endowment* 12, 10 (2019), 1126–1138.

[9] differential privacy team at Apple. 2017. Learning with privacy at scale. https://machinelearning.apple.com/research/learning-with-privacy-at-scale

[10] B. Ding, J. Kulkarni, and S. Yekhanin. 2017. Collecting telemetry data privately. In *Proc. NeurIPS*. 3574–3583.

[11] J. Duchi, M. Jordan, and M. Wainwright. 2013. Local privacy and statistical minimax rates. In *Proc. IEEE FOCS*. 429–438.

[12] J. Duchi, M. Jordan, and M. Wainwright. 2014. Privacy aware learning. *Journal of the ACM (JACM)* 61, 6 (2014), 1–57.

[13] C. Dwork. 2006. Differential privacy. In *Proc. ICALP*. 1–12.

[14] C. Dwork. 2010. Differential Privacy in New Settings. In *Proc. ACM-SIAM SODA*. 174–183.

[15] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. 2010. Differential Privacy under Continual Observation. In *Proc. ACM STOC*. 715–724.

[16] C. Dwork and A. Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9 (2014), 211–407.

[17] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. 2019. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proc. ACM-SIAM SODA*. 2468–2479.

[18] Ú. Erlingsson, A. Korolova, and V. Pihur. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proc. ACM CCS*. 1054–1067.

[19] L. Fan and L. Xiong. 2012. Real-time aggregate monitoring with differential privacy. In *Proc. ACM CIKM*. 2169–2173.

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

[21] L. Fan, L. Xiong, and V. S. Sunderam. 2013. Differentially Private Multi-dimensional Time Series Release for Traffic Monitoring. In *DBSec*. 33–48.

[22] M. Hassan, M. Rehmani, and J. Chen. 2019. Differential privacy techniques for cyber physical systems: a survey. *IEEE Commun. Surveys Tuts.* 22, 1 (2019), 746–789.

[23] N. Johnson, J. Near, and D. Song. 2018. Towards practical differential privacy for SQL queries. *Proc. VLDB Endow.* 11, 5 (2018), 526–539.

[24] M. Joseph, A. Roth, J. Ullman, and B. Waggoner. 2018. Local Differential Privacy for Evolving Data. *Proc. NeurIPS* 31 (2018), 2375–2384.

[25] P. Kairouz, S. Oh, and P. Viswanath. 2016. Extremal mechanisms for local differential privacy. *The Journal of Machine Learning Research* 17, 1 (2016), 492–542.

[26] S. Kasiviswanathan, H. Lee, N, S. Raskhodnikova, and A. Smith. 2011. What can we learn privately? *SIAM J. Computing* 40, 3 (2011), 793–826.

[27] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias. 2014. Differentially private event sequences over infinite streams. *Proc. VLDB Endow.* 7 (2014), 1155–1166.

[28] F. McSherry. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proc. ACM SIGMOD*. 19–30.

[29] T. Murakami and Y. Kawamoto. 2019. Utility-optimized local differential privacy mechanisms for distribution estimation. In *Proc. USENIX Security*. 1877–1894.

[30] T. Nguyên, X. Xiao, Y. Yang, S. Hui, H. Shin, and J. Shin. 2016. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053* (2016).

[31] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren. 2016. Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy. *Proc. ACM CCS* (2016), 192–203.

[32] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren. 2017. Generating synthetic decentralized social graphs with local differential privacy. In *Proc. ACM CCS*. 425–438.

[33] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. McCann, and S. Philip. 2018. LoPub: High-Dimensional Crowdsourced Data Publication with Local Differential Privacy. *IEEE Trans. Inf. Forensics Security* 13, 9 (2018), 2151–2166.

[34] J. Tang, A. Korolova, X. Bai, X. Wang, and X. Wang. 2017. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753* (2017).

[35] N. Wang, X. Xiao, Y. Yang, T.-D. Hoang, H. Shin, J. Shin, and G. Yu. 2018. PrivTrie: Effective frequent term discovery under local differential privacy. In *Proc. IEEE ICDE*. 821–832.

[36] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. Hui, H. Shin, J. Shin, and G. Yu. 2019. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In *Proc. IEEE ICDE*. 638–649.

[37] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren. 2016. RescueDP: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. *Proc. IEEE INFOCOM* (2016), 1–9.

[38] T. Wang, J. Blocki, N. Li, and S. Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *Proc. USENIX Security*. 729–745.

[39] T. Wang, J.-Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, and S. Jha. 2020. Continuous Release of Data Streams under both Centralized and Local Differential Privacy. *arXiv preprint arXiv:2005.11753* (2020).

[40] T. Wang, B. Ding, J. Zhou, C. Hong, Z. Huang, N. Li, and S. Jha. 2019. Answering multi-dimensional analytical queries under local differential privacy. In *Proc. ACM SIGMOD*. 159–176.

[41] T. Wang, N. Li, and S. Jha. 2019. Locally differentially private heavy hitter identification. *IEEE Trans. Dependable Secure Comput.* (2019).

[42] Z. Wang, W. Liu, X. Pang, J. Ren, Z. Liu, and Y. Chen. 2020. Towards Pattern-aware Privacy-preserving Real-time Data Collection. In *Proc. IEEE INFOCOM*. 109–118.

[43] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi. 2018. Privacy-preserving crowd-sourced statistical data publishing with an untrusted server. *IEEE Trans. Mobile Computing* 18, 6 (2018), 1356–1367.

[44] Q. Ye, H. Hu, X. Meng, and H. Zheng. 2019. PrivKV: Key-value data collection with local differential privacy. In *Proc. IEEE SP*. 317–331.

[45] Z. Zhang, T. Wang, N. Li, S. He, and J. Chen. 2018. CALM: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy. In *Proc. ACM CCS*. 212–229.

A PROOFS TO THEOREMS

A.1 Proof to Theorem 5.1

PROOF. Considering the independent randomness of each mechanism $\mathcal{M}_i$, for stream prefix V_t and a given output transcript $\mathbf{O}^* =$

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

$(\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t) \in O$, there is

$$\Pr[\mathcal{M}(V_t) \in \mathbf{O}^*] = \prod_{i=1}^{t} \Pr[\mathcal{M}_i(v_i) = \mathbf{o}_i]. \tag{12}$$

Similarly, for any w-neighboring stream prefix V'_t and the same $\mathbf{O}^* = (\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t)$, there is

$$\Pr[\mathcal{M}(V'_t) \in \mathbf{O}^*] = \prod_{i=1}^{t} \Pr[\mathcal{M}_i(v'_i) = \mathbf{o}_i]. \tag{13}$$

According to the definition of w-neighboring, there exists $\tau \in [t]$, such that $v_i = v'_i$ for $1 \le k \le \tau - w$ and $\tau + 1 \le i \le t$. Then, there is

$$\frac{\Pr[\mathcal{M}(V_t) \in \mathbf{O}^*]}{\Pr[\mathcal{M}(V'_t) \in \mathbf{O}^*]} = \prod_{i=\tau-w+1}^{\tau} \frac{\Pr[\mathcal{M}_i(v_i) = \mathbf{o}_i]}{\Pr[\mathcal{M}_i(v'_i) = \mathbf{o}_i]} \tag{14}$$

Note that database pairs v_i and v'_i are neighboring for $\tau - w + 1 \le i \le i$, and $\mathcal{M}_i$ satisfies ϵ_i-DP. So, there is $\frac{\Pr[\mathcal{M}_i(v_i)=\mathbf{o}_i]}{\Pr[\mathcal{M}_i(v'_i)=\mathbf{o}_i]} \le e^{\epsilon_i}$. Then, we can have

$$\log\left(\frac{\Pr[\mathcal{M}(V_t) \in \mathbf{O}^*]}{\Pr[\mathcal{M}(V'_t) \in \mathbf{O}^*]}\right) \le \log\left(\prod_{i=\tau-w+1}^{\tau} e^{\epsilon_i}\right) \tag{15}$$

$$= \sum_{i=\tau-w+1}^{\tau} \epsilon_i$$

Therefore, for any $\mathbf{O} \in O$, we have

$$\log\left(\frac{\Pr[\mathcal{M}(V_t) \in \mathbf{O}]}{\Pr[\mathcal{M}(V'_t) \in \mathbf{O}]}\right) \le \sum_{k=\tau-w+1}^{\tau} \epsilon_k. \tag{16}$$

And, if formula $\sum_{k=\tau-w+1}^{\tau} \epsilon_k = \epsilon$ holds, then we have $\log\left(\frac{\Pr[\mathcal{M}(V_t) \in \mathbf{O}]}{\Pr[\mathcal{M}(V'_t) \in \mathbf{O}]}\right) \le \epsilon$, which concludes the proof. □

A.2 Proof to Theorem 5.2

PROOF. The LDP guarantee can be directly obtained according to the post-processing theorem [16]. In the following, we mainly prove dis is an unbiased estimation of dis^*.

For any k that $1 \le k \le d$, since $\bar{\mathbf{c}}_{t,1}[k]$ is an unbiased estimate of $\mathbf{c}_t[k]$, we can have

$$\mathbb{E}[\bar{\mathbf{c}}_{t,1}[k]] = \mathbf{c}_t[k] \tag{17}$$

$$\text{Var}(\bar{\mathbf{c}}_{t,1}[k]) = \mathbb{E}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{c}_t[k])^2 \tag{18}$$

We rewrite the variance formula as

$$\text{Var}(\bar{\mathbf{c}}_{t,1}[k]) = \mathbb{E}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{c}_t[k])^2$$

$$= \mathbb{E}((\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k]) - (\mathbf{c}_t - \mathbf{r}_l[k]))^2$$

$$= \mathbb{E}[(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2 + (\mathbf{c}_t[k] - \mathbf{r}_l[k])^2]$$

$$- 2\mathbb{E}[(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k]) \cdot (\mathbf{c}_t[k] - \mathbf{r}_l[k])] \tag{19}$$

Both $\mathbf{c}_t[k]$ and $\mathbf{r}_l[k]$ are constant value, then the above equation can be further written as

$$\text{Var}(\bar{\mathbf{c}}_{t,1}[k]) = \mathbb{E}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{c}_t[k])^2$$

$$= \mathbb{E}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2 + (\mathbf{c}_t[k] - \mathbf{r}_l[k])^2 - 2(\mathbf{c}_t[k] - \mathbf{r}_l[k])^2$$

$$= \mathbb{E}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2 - (\mathbf{c}_t[k] - \mathbf{r}_l[k])^2. \tag{20}$$

Hence, there is

$$\mathbb{E}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2 = (\mathbf{c}_t[k] - \mathbf{r}_l[k])^2 + \text{Var}(\bar{\mathbf{c}}_{t,1}[k]) \tag{21}$$

Therefore, the expectation of dis in Eq (4) satisfies

$$\mathbb{E}(dis)$$

$$= \mathbb{E}\left(\frac{1}{d}\sum_{k=1}^{d}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2 - \frac{1}{d}\sum_{k=1}^{d}\text{Var}(\bar{\mathbf{c}}_{t,1}[k])\right)$$

$$= \frac{1}{d}\sum_{k=1}^{d}\mathbb{E}(\bar{\mathbf{c}}_{t,1}[k] - \mathbf{r}_l[k])^2 - \frac{1}{d}\sum_{k=1}^{d}\text{Var}(\bar{\mathbf{c}}_{t,1}[k])$$

$$= \frac{1}{d}\sum_{k=1}^{d}\left((\mathbf{c}_t[k] - \mathbf{r}_l[k])^2 + \text{Var}(\bar{\mathbf{c}}_{t,1}[k])\right) - \frac{1}{d}\sum_{k=1}^{d}\text{Var}(\bar{\mathbf{c}}_{t,1}[k])$$

$$= \frac{1}{d}\sum_{k=1}^{d}(\mathbf{c}_t[k] - \mathbf{r}_l[k])^2 = dis^*$$

□

A.3 Proof to Theorem 5.3

PROOF. We prove the privacy guarantee of LBD and LBA as followings.

(1) LBD satisfies w-event LDP:

In sub mechanism $\mathcal{M}_1$, the dissimilarity budget $\epsilon_{t,1}$ at each timestamp t is $\epsilon/(2w)$. Then, for every t, there is

$$\sum_{k=t-w+1}^{t} \epsilon_{k,1} = \epsilon/2. \tag{22}$$

In sub mechanism $\mathcal{M}_2$, at each timestamp t, at most half of the remaining publication budget is allocated if publication occurs. That is to say, $\epsilon_{t,2} = (\epsilon/2 - \sum_{k=t-w+1}^{t-1} \epsilon_{k,2})/2$.

Firstly, for any $1 \le t \le w$, LBD distributes the budget in a sequence of $\epsilon/4, \epsilon/8, \ldots$, there will be at most w publications as a time window consisting of w timestamps. So,

$$\sum_{i=1}^{t} \epsilon_{k,2} \le (\epsilon/2) \cdot (1 - \frac{1}{2^w}) \le \epsilon/2. \tag{23}$$

Suppose that $\sum_{k=t-w+1}^{t} \epsilon_{k,2} \le \epsilon/2$ holds for $t = w + m$, i.e., $\sum_{k=m+1}^{w+m} \epsilon_{k,2} \le \epsilon/2$. Then, at timestamp $t = w + m + 1$, there is

$$\sum_{k=m+2}^{w+m+1} \epsilon_{k,2} = \sum_{k=m+2}^{w+m} \epsilon_{k,2} + \epsilon_{w+m+1,2} \tag{24}$$

Since $\epsilon_{w+m+1,2}$ is half of the remaining publication budget at time $w + m + 1$, there is

$$\epsilon_{w+m+1,2} \le (\epsilon/2 - \sum_{m+2}^{w+m} \epsilon_{k,2})/2. \tag{25}$$

By substituting Eq. 25 in to Eq. 26, there is

$$\sum_{k=m+2}^{w+m+1} \epsilon_{k,2} = \sum_{k=m+2}^{w+m} \epsilon_{k,2} + (\epsilon/2 - \sum_{m+2}^{w+m} \epsilon_{k,2})/2 \tag{26}$$

$$= \epsilon/4 + (\sum_{k=m+2}^{w+m} \epsilon_{k,2})/2 \le \epsilon/4 + \epsilon/4 = \epsilon/2.$$

This implies that, if $\sum_{k=t-w+1}^{t} \epsilon_{k,2} \leq \epsilon/2$ holds for $t = w + m$, then it also holds for $t = w + m + 1$. Besides, $\sum_{k=t-w+1}^{t} \epsilon_{k,2} \leq \epsilon/2$ always holds for $1 \leq t \leq w$. Therefore, we can prove that, for every timestamp $t \geq 1$, there is

$$\sum_{k=t-w+1}^{t} \epsilon_{k,2} \leq \epsilon/2. \tag{27}$$

Because LBD executes $\mathcal{M}_{t,1}$ and $\mathcal{M}_{t,2}$ sequentially at each timestamp t, the total privacy budget in a window of size w should be

$$\sum_{k=t-w+1}^{t} \epsilon_k = \sum_{k=t-w+1}^{t} \epsilon_{k,1} + \sum_{k=t-w+1}^{t} \epsilon_{k,2} \leq \epsilon, \tag{28}$$

which proves that LBD satisfies w-event ϵ-LDP.

(2) LBA satisfies w-event LDP:

The sub mechanism $\mathcal{M}_{t,1}$ in LBA is identical to that in LBD. That is, for every t, there is

$$\sum_{k=t-w+1}^{t} \epsilon_{k,1} = \epsilon/2. \tag{29}$$

The sub mechanism $\mathcal{M}_{t,2}$ in LBA is $\epsilon_{t,2}$-LDP where $\epsilon_{t,2}$ may be nullified, or be absorbed, or absorb unused budgets from previous timestamps.

Without loss of generality, we suppose t be a publication timestamp that absorbed the unused budget from the last α timestamps. Then, according to Algorithm 2, the publication budget $\epsilon_{t,2}$ at current timestamp equals to $(1+\alpha) \cdot \frac{\epsilon}{2 \cdot w}$, but the publication budget at both the preceding α timestamps ($i \in [t-\alpha, i-1]$ which are absorbed) and the succeeding α timestamps ($i \in [t+1, t+\alpha]$ which are nullified), will be 0, i.e., $\epsilon_{i,2} = 0$.

Then, any window of size w sliding over timestamp i, must cover at least α timestamps with $\epsilon_{i,2} = 0$. Suppose that, there are n timestamps having $\epsilon_{i,2} = 0$ due to the absorption or nullfication by timestamp t. Then, the sum of budget of publication timestamp t and the n zero-budget timestamps is at most $(1+\alpha) \cdot \frac{\epsilon}{2 \cdot w}$. This is equivalent to the case where each of these $n+1$ *timestamps* is assigned with uniform budget of $(1+\alpha) \cdot \frac{\epsilon}{2 \cdot w \cdot (n+1)} \leq \frac{\epsilon}{2 \cdot w}$. This also holds for other publication timestamp t' that absorbed its previous unused budget in the same window as t. Then, the total publication budget in a time window of size w, summing up the non-zero budget at publication timestamps and zero-budget at nullified and absorbed timestamps, would be $\sum_{k=t-w+1}^{t} \epsilon_{k,2} \leq \frac{\epsilon}{2 \cdot w} \cdot w = \epsilon/2$.

Similarly, as LBA also executes $\mathcal{M}_{t,1}$ and $\mathcal{M}_{t,2}$ sequentially at each timestamp t, the total privacy budget in a window of size w should be

$$\sum_{k=t-w+1}^{t} \epsilon_k = \sum_{k=t-w+1}^{t} \epsilon_{k,1} + \sum_{k=t-w+1}^{t} \epsilon_{k,2} \leq \epsilon, \tag{30}$$

which proves that LBA satisfies w-event ϵ-LDP.

□

A.4 Proof to Lemma 6.1

PROOF. We compare the MSE of LBU (budget division framework) and that of LPU (population division framework) with the same FO, e.g., GRR, which are denoted as $\text{MSE}_{\text{LBU+GRR}}$ and $\text{MSE}_{\text{LPU+GRR}}$ respectively.

With GRR protocol, there is

$$\text{MSE}_{\text{LBU+GRR}} = V_{\text{GRR}}(\epsilon/w, N) = \frac{d - 2 + e^{\epsilon/w}}{N \cdot (e^{\epsilon/w} - 1)^2} \tag{31}$$

and there is

$$\text{MSE}_{\text{LPU+GRR}} = V_{\text{GRR}}(\epsilon, N/w) = w \cdot \frac{d - 2 + e^{\epsilon}}{N \cdot (e^{\epsilon} - 1)^2} \tag{32}$$

Then, there is

$\text{MSE}_{\text{LBU+GRR}} - \text{MSE}_{\text{LPU+GRR}}$

$$= \frac{1}{N} \left[\frac{d - 2 + e^{\epsilon/w}}{(e^{\epsilon/w} - 1)^2} - w \frac{d - 2 + e^{\epsilon}}{(e^{\epsilon} - 1)^2} \right]$$

$$= \frac{d-2}{N} \left[\frac{1}{(e^{\epsilon/w} - 1)^2} - \frac{w}{(e^{\epsilon} - 1)^2} \right] +$$

$$\frac{e^{\epsilon/w}}{N} \left[\frac{1}{(e^{\epsilon/w} - 1)^2} - \frac{we^{-\epsilon/w}}{(e^{\epsilon} - 1)^2} \right]$$

$$= \frac{(d-2)}{N(e^{\epsilon/w} - 1)^2 (e^{\epsilon} - 1)^2} \left[(e^{\epsilon} - 1)^2 - w(e^{\epsilon/w} - 1)^2 \right] +$$

$$\frac{e^{\epsilon/w}}{N(e^{\epsilon/w} - 1)^2 (e^{\epsilon} - 1)^2} \left[(e^{\epsilon} - 1)^2 - we^{-\epsilon/w}(e^{\epsilon/w} - 1)^2 \right]$$

Simply, we denote $e^{\epsilon/w}$ as z. Since $\epsilon > 0$, $z > 1$. Then, we have

$$(e^{\epsilon} - 1)^2 - w((e^{\epsilon/w} - 1)^2)$$

$$= (z^w - 1)^2 - w(z - 1)^2$$

$$= (z - 1)^2 [(1 + z^2 + ... + z^{w-1})^2 - w] > 0$$

and there is

$$(e^{\epsilon} - 1)^2 - we^{-\epsilon/w}(e^{\epsilon/w} - 1)^2$$

$$= (z^w - 1)^2 - wz^{w-1}(z - 1)^2$$

$$= (z - 1)^2 [(1 + z^2 + ... + z^{w-1})^2 - wz^{w-1}] > 0$$

Besides, $d \geq 2$, therefore, $\text{MSE}_{\text{LBU+GRR}} - \text{MSE}_{\text{LPU+GRR}} > 0$. That is to say, $\text{MSE}_{\text{LBU+GRR}} > 0$ is always smaller than $\text{MSE}_{\text{LPU+GRR}}$.

The proof details under the OUE protocol are similar, and therefore omitted here. So, given the same FO protocol GRR or OUE, the MSE of LPU is smaller than that of LBU, i.e., $\text{MSE}_{\text{LPU}} < \text{MSE}_{\text{LBU}}$.

In conclusion, it can achieve much better utility to divide the population, instead of dividing privacy budget, in a time window to achieve w-event LDP.

□

A.5 Proof to Theorem 6.2

PROOF. We prove this theorem by proving the following claims.

(1) In any time window consists of w consecutive timestamps, each user reports to the server at most once.
(2) Each user's reported data satisfies ϵ-LDP.

The second claim holds for both LPD and LPA apparently. This is because, each time LPD and LPA request users to report value to the server, the selected users will report via an FO with ϵ as privacy budget (e.g., Lines 4 and 5 in Algorithm 3, and Line 13 in Algorithm 4).

We prove the first claim by proving that, at each timestamp t, the total number of users allocated in a window of size w is no larger than N, i.e., $\sum_{k=t-w+1}^{t} |U_k| < N$. Then, as long as we sample a fresh

Xuebin Ren[1], Liang Shi[1], Weiren Yu[2], Shusen Yang[1], Cong Zhao[3], Zongben Xu[1]

set of users U_k at each time k and ensure that $U_{k,1} \cap U_{k,2} = \emptyset$, we can guarantee that each user participates only once in a window of size w.

For LPD, in $\mathcal{M}_{t,1}$, $\lfloor N/(2w) \rfloor$ users are allocated at each timestamp t. So, for every t and $i \in [t]$, there is $\sum_{k=i-w+1}^{i} |U_{k,1}| \leq N/2$. $\mathcal{M}_{t,2}$ each time either publishes with additional users $U_{t,2}$ or approximates the last release without assigning any user. In the latter case, $|U_{t,2}|$ is simply zero. In the former case, $|U_{t,2}| = (N/2 - \sum_{k=i-w+1}^{i-1} |U_{k,2}|)/2$. Particularly, since $\mathcal{M}_{t,2}$ always uses up to half of the available users, there is always. $0 \leq \sum_{k=t-w+1}^{t} |U_{k,2}| \leq N/2$. Therefore, for every t and $i \in [t]$, the total number of publication users in a time window of size w, should be $\sum_{k=i-w+1}^{i} |U_k| = \sum_{k=i-w+1}^{i} |U_{k,1}| + \sum_{k=i-w+1}^{i} |U_{k,2}| \leq N$.

For LPA, similarly, in $\mathcal{M}_{t,1}$, there is $\sum_{k=i-w+1}^{i} |U_{k,1}| \leq N/2$ for every t and $i \in [t]$. In $\mathcal{M}_{t,2}$, suppose i is a timestamp which absorbed budget from m preceding timestamps, where m must be smaller than the window size w, i.e., $0 \leq m \leq w - 1$. Then, at the current timestamp i, the population of publication users should be $|U_{t,2}| = \frac{(m+1)N}{2w}$; at the timestamps $i - m \leq k \leq i - 1$ and $i + 1 \leq k \leq i + m$, the publication users are either skipped or nullified, i.e., $|U_{k,2}| = 0$. Since, according to UA, any w-timestamp-long window that contains timestamp i would also have $l \geq m$ timestamps that were either absorbed or nullified, i.e., $|U_{k,2}| = 0$. Therefore, the sum of population of i along with the these l timestamps is at most $(m + 1)N/(2w)$, which is at most equal to the case where each of these $l + 1$ timestamps receives uniform population $|U_{k,2}| = \frac{(m+1)N/(2w)}{l+1} \leq N/(2w)$. This holds for any timestamp i' that absorbed users from m' previous timestamps and lies in the same window as i. Therefore, $\sum_{k=i-w+1}^{i} |U_{k,2}| \leq \sum_{k=i-w+1}^{i} N/(2w) = N/2$. As $m \geq 0$, $|U_{k,2}| \geq 0$ holds for every k, so, $\sum_{k=i-w+1}^{i} |U_{k,2}| \geq 0$.

So far, we have proved that at each timestamp t, the total number of users allocated in a window of size w is no larger than N, i.e., $\sum_{k=t-w+1}^{t} |U_k| < N$. In both LPD and LPA, the sampled users $U_{t,2}$ at each time always come from the remaining available users U_A excludes $U_{t,1}$. So, there must be $U_{k,1} \cap U_{k,2} = \emptyset$. Therefore, we can always guarantee that each user participate only once in a window of size w. □

B SUPPLEMENTARY METHOD DESCRIPTIONS

B.1 Detailed Description of LPD

B.1.1 Algorithm Description. Algorithm 3 elaborates the implementation of LPD. At each timestamp t, LPD consists of two sub mechanisms, $\mathcal{M}_{t,1}$ for private dissimilarity calculation and $\mathcal{M}_{t,2}$ for private strategies determination and publication users allocation. Initially, the whole population U is taken as available users U_A, and the released frequency histogram is set as $\mathbf{r}_0 = \langle 0, \ldots, 0 \rangle^d$ (Line 1). Then, at each timestamp t, LPD performs $\mathcal{M}_{t,1}$ and $\mathcal{M}_{t,2}$ in sequence.

Sub Mechanism $\mathcal{M}_{t,1}$ (Lines 3-6). $\mathcal{M}_{t,1}$ aims to privately calculate the dissimilarity measure *dis* with the dissimilarity users $U_{t,1}$. Since half the population $\lfloor N/2 \rfloor$ is divided for $\mathcal{M}_1$ in a window of size w, $\mathcal{M}_{t,1}$ randomly samples a subset $U_{t,1}$ with $\lfloor N/(2w) \rfloor$

users from U_A at timestamp t. These sampled users are removed temporarily from U_A, i.e., $U_A \leftarrow U_A \setminus U_{t,1}$ (Line 3) to ensure they only participate once within a window. Then the server requests all users in $U_{t,1}$ to upload locally perturbed data with the entire privacy budget ϵ, which is stored in a database $\overline{D}_{t,1}$ (Line 4). Using the LDP frequency oracle on $\overline{D}_{t,1}$, an unbiased estimation $\overline{\mathbf{c}}_{t,1} \leftarrow \text{FO}(\overline{D}_{t,1}, \epsilon)$ of the true frequency count $\mathbf{c}_{t,1}$ can be obtained from $\overline{D}_{t,1}$ (Line 5). Based on Theorem 5.2, the server computes an LDP but unbiased estimation of dissimilarity as $dis = \frac{1}{d} \sum_{k=1}^{d} (\overline{\mathbf{c}}_{t,1}[k] - \mathbf{r}_{t-1}[k])^2 - \frac{1}{d} \sum_{k=1}^{d} \text{Var}(\overline{\mathbf{c}}_{t,1}[k])$, where $\frac{1}{d} \sum_{k=1}^{d} \text{Var}(\overline{\mathbf{c}}_{t,1}[k])$ can be calculated based on ϵ and $|U_{t,1}|$. We replace $\mathbf{r}_l$ with $\mathbf{r}_{t-1}$ since that the approximation $\mathbf{r}_t = \mathbf{r}_{t-1}$ is always adopted until a publication occurs. The calculated dissimilarity *dis* is then passed to $\mathcal{M}_{t,2}$.

Sub Mechanism $\mathcal{M}_{t,2}$ (Lines 7-19). The sever compares the dissimilarity *dis* returned by $\mathcal{M}_{t,1}$ with the possible publication error *err* to choose publication or approximation. The possible publication error *err* is determined by the privacy budget ϵ and number of potential publication users, i.e., $|U_{t,2}|$. So, to estimate the potential publication error, the server first calculates the number of remaining publication users (indicated as N_{rm}) by subtracting the number of used publication users in the current window $\sum_{k=t-w+1}^{t-1} |U_{k,2}|$ ($U_{k,2} = \emptyset$ for $k \leq 0$) from the number of total publication users $\lfloor N/2 \rfloor$ (Line 7). After that, the number of potential publication users is set as $|U_{t,2}| = N_{rm}/2$ to make it exponentially decreasing within a window (Line 8). Then, the potential publication error *err* at the current timestamp equals to $V(\epsilon, N_{rm}/2)$, can be calculated by Eq. (2). Next, the server simply compares *dis* with *err* and chooses a strategy with less error: (1) if $dis > err$, it implies that the approximation error is larger than potential publication error, the server chooses publication to reduce the error. In particular, it samples a fresh group of publication users $U_{t,2}$ and then requests them to upload perturbed data, from which, the server can obtain an unbiased estimation $\overline{\mathbf{c}}_{t,2}$ as the output. Note that, with exponential decaying, $|U_{t,2}|$ may drop quickly and lead to large sampling error due to insufficient users. Therefore, we set a minimal threshold $u_{\min}$ and $\mathcal{M}_{t,2}$ directly chooses the approximation strategy if $|U_{t,2}|$ is too small. If the publication error *err* is larger than *dis*, the server then decides to approximate with the last release $\mathbf{r}_{t-1}$ with no actual publication, i.e., setting $U_{t,2} = \emptyset$ (Line 16). Finally, at the end of the timestamp t, the server recycles both the used dissimilarity and publication users at $t - w + 1$ as available for the next timestamp, i.e., $U_A = U_A \cup U_{t-w+1,1} \cup U_{t-w+1,2}$. The recycling process ensures each user can contributes again after w timestamps but only contribute once in a window of size w to satisfy w-event LDP.

B.2 Detailed Description of LPA

B.2.1 Algorithm Description. The details of LPA are shown in Algorithm 4. $\mathcal{M}_{t,1}$ for private dissimilarity calculation is identical to that in LPD. We mainly explain the details of $\mathcal{M}_{t,2}$ for publication strategy decision and participant users allocation. At each timestamp t, $\mathcal{M}_{t,2}$ first allocates a fixed number of publication users $|U_{t,2}| = \lfloor N/(2w) \rfloor$. According to the basic idea, there will be two cases at t: (i) if the approximation strategy was selected (i.e., the publications were skipped) in the previous timestamps, the corresponding publication users will be added to $U_{t,2}$; (2) if the publication strategy is selected in the previous timestamp, $U_{t,2}$ may

be nullified and approximation strategy may be selected at current time t. Therefore, $\mathcal{M}_{t,2}$ has to identify which case the current timestamp belongs to. Denote $U_{l,2}$ as the population of publication users at the timestamp l where last publication $\mathbf{r}_l$ occurred. Since $\lfloor N/(2w) \rfloor$ users are uniformly allocated at each timestamp in $\mathcal{M}_{t,2}$, there will be $t_N = |U_{l,2}|/(\lfloor N/(2w) \rfloor) - 1$ timestamps after l where the user allocation must be nullified (Line 4). Therefore, if $t - l \leq t_N$, the user allocation at current time t is nullified and $\mathcal{M}_{t,2}$ outputs with the last release $\mathbf{r}_t = \mathbf{r}_{t-1}$. Otherwise, Case (i) holds, $\mathcal{M}_{t,2}$ absorbs the users of previously skipped publications and decides whether to freshly publish with these absorbed users (Lines 11-15) or continues to approximate with the last release (Lines 16-17). In

particular, it first computes the number of skipped publications between the last publication timestamp l and t, which corresponds to $t_A = t - (l + t_N)$. To satisfy w-event LDP, at most $w - 1$ timestamps can be absorbed. Therefore, the remaining population is $N_{rm} = \lfloor N/(2w) \rfloor \cdot \min(t_A, w)$. Using these remaining users as the publication users $U_{t,2} = N_{rm}$, the potential publication error err can be estimated. If the dissimilarity $dis > err$, the server requests a random sampled userset $U_{t,2}$ with the population $|U_{t,2}|$ to respond with LDP and outputs an unbiased $\bar{\mathbf{c}}_{t,2}$; otherwise, $\mathcal{M}_{t,2}$ chooses the approximation strategy and simply outputs $\mathbf{r}_t = \mathbf{r}_{t-1}$ without publication user $|U_{t,2}|$ allocation. Finally, it also recycles both the dissimilarity and publication users used at timestamp $t - w + 1$.