TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 25/38 pp1722–1736
DOI: 10.26599/TST.2025.9010063
Volume 31, Number 3, June 2026

Research Article

Efficient Privacy-Preserving Streaming Histogram Publication via Sampling-Based Shuffled Method

Xiao Zheng, Yimin Jin, Xiujun Wang*, and Zixia Liu

Abstract: The rapid growth of web technologies and the industrial Internet has resulted in massive data streams that are essential for real time decision-making applications. A critical challenge lies in dynamically publishing histograms for the most recent w elements within sliding windows while ensuring individual privacy protection. Although Shuffle Differential Privacy (SDP) provides strong privacy guarantees, existing methods for SDP histograms face huge time and space overhead, as they require caching all sliding window elements. This limitation hinders their practicality in large-scale, real time settings. To overcome these limitations, this paper proposes Sampling-based Shuffle Differential Privacy (SSDP), an efficient algorithm for the publication of privacy-preserving histogram. Central to SSDP is the Online Sampling Sketch Structure (OSSS), which dynamically captures essential data patterns from streaming windows through adaptive sampling, enabling approximate histogram generation with dramatically reduced time and space complexities. To counter advanced privacy threats, SSDP integrates targeted noise injection and a hash-optimized Generalized Randomized Response (GRR) mechanism, providing robust defenses against shuffling and publisher collusion attacks. Moreover, a correlated unbiased estimation mechanism enhances both privacy preservation and data utility. Extensive experiments on real-world datasets demonstrate the superiority of SSDP, achieving up to 40% reduction in processing time compared to state-of-the-art methods, with minimal loss in utility.

Key words: Shuffled Differential Privacy (SDP); data streams; sampling; histogram publishing

1 Introduction

The rapid growth of network technologies and the widespread adoption of the industrial Internet have significantly increased the volume of data generated across various sectors. This surge in data generation presents both opportunities and challenges, particularly in the context of real time analytics. One crucial aspect of data analysis involves the publication of statistical summaries, such as histograms, which capture the distribution of data over time and enable applications, such as anomaly detection, trend analysis, and predictive modeling[1–4]. However, publishing histograms, with their possible privacy risks, has become a major issue. The continuous stream of sensitive information allows adversaries to exploit temporal correlations between consecutive data points, potentially leading to severe privacy breaches such as identity theft or unauthorized profiling[5, 6]. As a result, histograms must now be secure in terms of personal privacy in addition to providing meaningful and accurate statistical summaries of streaming data.

• Xiao Zheng, Yimin Jin, Xiujun Wang, and Zixia Liu are with School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, China, and also with Anhui Engineering Research Center for Intelligent Applications and Security of Industrial Internet, Ma'anshan 243032, China. E-mail: xzheng@ahut.edu.cn; 13074031365@163.com; xjwang@ahut.edu.cn; zxliu@ahut.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2024-11-30; revised: 2025-03-05; accepted: 2025-04-16

To address these concerns, Differential Privacy (DP) has become a widely accepted approach for ensuring individual privacy in data analysis. By providing a mathematical guarantee that the inclusion or exclusion of any individual's data will not significantly alter the outcome of a computation, DP enables data analysis without compromising privacy[7]. Local Differential Privacy (LDP), in particular, has gained traction in applications, such as Google's RAPPOR[8] and Apple's differential privacy techniques, as it allows for the collection of valuable statistical information without requiring access to raw data. While these approaches are effective in preserving privacy, they encounter significant challenges when applied to streaming data due to the dynamic nature of data and the large volumes involved.

Specifically, traditional DP methods, such as the Laplace mechanism[9] and NoiseFirst[10], are often inefficient for streaming data. These techniques typically require substantial memory to store large portions of the data, making them impractical for real time applications, where data streams are continuous and unbounded. Furthermore, many dynamic data stream methods, such as RAPPOR and AUE[3], trade off processing efficiency in exchange for maintaining privacy and data utility. For instance, regression techniques used in RAPPOR or One-Hot encoding employed by AUE introduce a linear relationship between communication costs and the data value domain, further exacerbating the system's inefficiency. Moreover, the traditional differential privacy model faces the drawbacks of the model itself, e.g., DP requires a trusted third party, LDP is limited by the number of users. In short, for dynamic data scenarios, existing histogram publishing methods based on traditional differential privacy models still lack a complete solution to address the above challenges.

A promising alternative to these challenges is the Shuffle Differential Privacy (SDP) framework, introduced by Bittau et al.[7], SDP strengthens privacy protection by locally perturbing the data and shuffling them before processing, which not only mitigates the risk of individual re-identification but also reduces the impact of user count on accuracy, without the need for a trusted third party. While this method offers stronger privacy guarantees, its application to real time, large-scale data streams remains under explored.

Considering that sampling can be approximated as incomplete shuffling and that the sampling process itself can be efficient. Consequently, a Sampling-based Shuffle Differential Privacy (SSDP) algorithm is proposed in this paper to overcome both model and time-space constraints. Through the combination of DP advantages and efficient sampling techniques, the time and space complexities traditionally associated with SDP in streaming data applications are effectively reduced. Specifically, random sampling techniques are utilized to extract statistical summaries from data streams while strong privacy guarantees are maintained. Computational overhead is minimized through the implementation of an Online Sampling Sketch Structure (OSSS), where dynamic adjustment of the sampling rate is achieved based on real time feedback, thereby establishing an optimized balance between accuracy and efficiency.

Our approach significantly improves upon existing methods by ensuring that the privacy-preserving histograms generated under the sliding window model can be efficiently published in real time, without clearly sacrificing data utility or privacy. Theoretical analysis of the SSDP algorithm demonstrates that it provides strong differential privacy guarantees, while experimental results confirm that it achieves substantial improvements in terms of both time and space efficiency compared to traditional methods. These findings position SSDP as an effective solution for privacy-preserving streaming data analysis, offering a practical method for histogram publication in large-scale, real time environments.

In summary, the main contributions of this paper are as follows:

- The SSDP algorithm is proposed, which integrates DP based on shuffling with efficient sampling techniques to achieve publication of a histogram in real time that preserves privacy from streaming data.
- An OSSS is introduced, which is designed to reduce time and space complexities.
- Detailed theoretical analysis and experimental evaluation are presented, demonstrating the effectiveness of SSDP in terms of privacy protection and computational efficiency.

The remainder of this paper is organized as follows: Section 2 reviews related work in privacy-preserving streaming data analysis, highlighting the limitations of existing methods. Section 3 presents the system model and theoretical framework. Section 4 describes the SSDP algorithm and its privacy analysis. Section 5 presents experimental results that validate the

proposed method. Finally, Section 6 concludes the paper.

2 Related Work

This section provides an overview of related work on privacy-preserving data stream publishing, particularly in the context of DP and LDP. Given that static data stream methods cannot directly handle continuous counting queries, existing research can be broadly classified into two categories: methods for static data and methods for dynamic data streams.

2.1 Methods for static data

Static data stream publishing methods are well-established, with a primary focus on publishing data with high accuracy while preserving privacy and maintaining data integrity. One of the key research areas is grouping, specifically, determining the number and structure of groups. For instance, Qardaji et al.[4] proposed a layered approach that transforms histograms into a binary tree, ensure that the number of nodes used in any range query does not exceed the number of nodes minus one multiplied by the tree height. This significantly reduces the Mean Squared Error (MSE) of range queries. Xu et al.[10] introduced the NoiseFirst and StructureFirst algorithms, both of which use dynamic programming to derive the optimal histogram. However, these methods focus solely on the reconstruction error introduced by grouping and do not account for the noise error resulting from the Laplace mechanism.

Another significant issue in static methods is the merging of data. Acs et al.[2] proposed the P-HPartition algorithm, which balances reconstruction and noise errors using a greedy binary strategy. However, this approach only merges adjacent buckets with similar counts and fails to consider global bucket count relationships. Similarly, Kellaris et al.[11] introduced the GS algorithm, which suggests that grouping based on ordered histograms can reduce reconstruction errors, but its simple equal-width interval segmentation results in considerable errors. In their work, Cheu and Zhilyaev[12] proposed an approximate histogram protocol that uses randomization to achieve a simplified privacy protection structure.

2.2 Methods for dynamic data

In contrast to static data, dynamic data streams require continuous updates as new data arrive. These updates must maintain the balance between availability, privacy, and the efficient allocation of privacy budgets. However, the real time requirements of dynamic data publishing, combined with the high memory consumption of traditional methods, make static differential privacy mechanisms unsuitable for dynamic stream settings. As such, the definition of differential privacy must be extended for streaming contexts.

Recent advancements in dynamic data stream publishing have classified DP techniques into three categories based on privacy granularity: event-level, user-level, and w-event-level privacy. Early research focused on event-level and user-level privacy. Event-level DP protects individual events in the data stream, while user-level DP hides all events associated with a particular user. For example, Dwork et al.[13] proposed a binary tree-based DP algorithm for finite streams. Chan et al.[14] applied the same approach to generate partial sums of binary counts in both finite and infinite streams. Additionally, Dwork[15] introduced a cascaded buffer counter with event-level DP, which adapts to varying stream densities. Bolot et al.[16] introduced the concept of attenuated privacy, gradually reducing the privacy cost of past data.

On the other hand, user-level DP, as introduced by Chen et al.[17], extends DP to multidimensional query settings and formalizes user privacy protection for connected data tuples. Xu et al.[18] also introduced user level LDP to handle multidimensional analytical queries, in order to formalize and protect user privacy when publishing data tuples connected by users. However, event-level privacy alone is insufficient, as it fails to address the privacy risks posed by users participating in multiple events.

To tackle these challenges, more recent work has shifted towards w-event-level DP, which provides privacy guarantees for any time window consisting of two consecutive time instances. Perrier et al.[19] used contribution limitation techniques to truncate the data stream in order to improve utility and provide event level privacy. Ren et al.[20] proposed a new concept of average w-event privacy and an infinite stream privacy protection scheme based on Lyapunov optimization, ensuring data privacy and high practicality. Kellaris et al.[21] proposed a solution to handle infinite data streams while preserving w-event privacy. This approach has gained traction due to its ability to protect a meaningful range of data without overburdening the

system. For example, Ye et al.[22] introduced a method based on LDP that combines w-event DP with practical data stream protection. Similarly, Wang et al.[23] proposed RescueDP, a multi-dimensional data stream mechanism that applies w-event DP to group similar dimensions. Wang et al.[24] proposed a pattern aware data stream collection mechanism based on the w-event LDP metric, and so on.

Despite these advancements, many of the existing dynamic data stream methods still rely on DP or LDP. Both approaches either rely on trusted aggregators or suffer from significant efficiency losses due to LDP's higher privacy costs. Furthermore, to the best of our knowledge, few studies have explored the use of SDP in dynamic data stream processing.

In summary, existing methods for histogram publication over data streams encounter significant challenges. Approaches based on DP or LDP often fail to strike an efficient balance between privacy and accuracy, and they struggle to scale to high-speed streams where data arrive rapidly and distributions evolve continuously. Furthermore, methods that rely on caching all elements within a sliding window suffer from prohibitive space and time complexities, rendering them impractical for real time, large-scale applications. These limitations underscore the need for the SSDP algorithm, which provides a highly efficient, scalable solution for dynamic, privacy-preserving histogram publication in real time environments.

3 System Model and Problem Definition

This section begins by introducing the system model for histogram publication in data streams within the SDP framework. It then presents the essential definitions and concepts required to formally formulate the problem. Finally, a formal definition of the studied histogram publication problem is provided.

3.1 System model

The system model for privacy-preserving histogram publication in data streams within the SDP framework consists of three main components[7]: the user side, the shuffler side, and the publisher side. These components interact to generate and publish histograms from streaming data while preserving privacy. The roles of each component are as follows (refer to Fig. 1 for a visual representation):

- The user side perturbs the raw data before sending it to the shuffler. This perturbation introduces noise to protect individual privacy by obscuring sensitive information.
- The shuffler side receives the perturbed data and performs a shuffling operation to decouple the data from its original source, ensuring that the relationship between data and source is obfuscated.
- The publisher side aggregates the shuffled data and publishes the resulting histograms, which represent aggregate information about the data stream while ensuring privacy preservation.

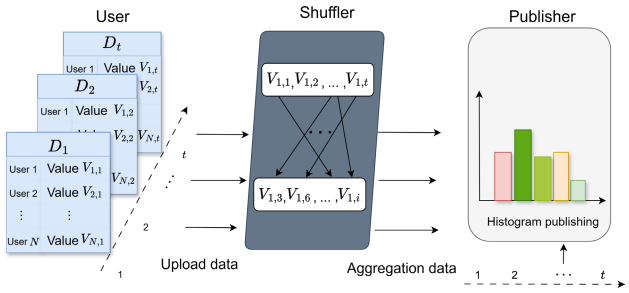

Fig. 1 Overview of system model.

In the model[25], both the publisher and the shuffler are assumed to be honest but curious, meaning they adhere to the protocol but may attempt to infer sensitive information from the data. However, the privacy guarantees of the SDP framework limit their ability to do so.

3.2 Privacy definitions

This subsection introduces the privacy definitions that underpin the histogram publication problem under SDP, providing the theoretical foundation for the privacy guarantees of our approach.

Definition 1 (DP[9]) An algorithm Z satisfies (ε, δ)-DP if, for any pair of neighboring datasets D and D', and any output set $\mathcal{R}$, the following condition holds:

$$\Pr[Z(D) \in \mathcal{R}] \leqslant e^{\varepsilon} \cdot \Pr[Z(D') \in \mathcal{R}] + \delta,$$

where both D and D' are datasets containing user data records, the only difference is that the two datasets differ only at one user record i (i.e., $D = D' - i$), ε is the privacy budget, and δ is the constraint parameter used to relax the inequality equality condition.

Definition 2 (Data stream[13]) A data stream $\mathcal{D} = e_1, e_2, \ldots, e_t, \ldots$, consists of data points $e_t = (t, v_t)$, where t represents the timestamp and v_t is the value of the data point at t.

Definition 3 (Sliding window[21]) A sliding window over a data stream $\mathcal{D}$ at t consists of the most

recent w data points. Formally, the sliding window W_t at t is defined as

$$W_t = (e_{t-w+1}, e_{t-w+2}, \ldots, e_t),$$

where w is the window size and $t \geqslant w$ is the current timestamp.

Definition 4 (w-neighboring[21]) Two data stream prefixes $\mathcal{D}_t$ and $\mathcal{D}'_t$ are w-neighboring if they differ by at most one data point. Specifically, if i_1 and i_2 are the indices where the streams differ, they must satisfy the condition $i_2 - i_1 + 1 \leqslant w$, ensuring the difference is localized within a sliding window of size w.

Definition 5 (w-event privacy[21]) A privacy-preserving algorithm $\mathcal{M}$ satisfies w-event DP if, for any w-neighboring stream prefixes $\mathcal{D}_t$ and $\mathcal{D}'_t$, and any output set $\mathcal{R}'$, the following condition holds:

$$\Pr[\mathcal{M}(\mathcal{D}_t) \in \mathcal{R}'] \leqslant e^{\varepsilon} \cdot \Pr[\mathcal{M}(\mathcal{D}'_t) \in \mathcal{R}'].$$

This ensures that the outputs of $\mathcal{M}$ do not reveal more information than permitted by the privacy budget ε about the individual data points within the stream prefix.

Definition 6 (Generalized Randomized Response (GRR)[26]) The basic mechanism in LDP is called randomized response. It is introduced for the binary case (i.e., $\mathcal{L} = \{0, 1\}$), but can be easily generalized. Here we describe the generalized version of random response. In GRR, each user with private value $v \in \mathcal{L}$ sends GRR(v) to the server, where GRR(v) outputs the true value v with probability p, and a randomly chosen $v' \in \mathcal{L}$ where $v' \neq v$, with probability $1 - p$. Denote the size of the domain as $\zeta = |\mathcal{L}|$, we have

$$\Pr[\mathrm{GRR}(v) = y] = \begin{cases} \dfrac{e^{\varepsilon}}{e^{\varepsilon} + d - 1}, & \text{if } y = v; \\ \dfrac{1}{e^{\varepsilon} + d - 1}, & \text{if } y \neq v \end{cases} \tag{1}$$

Definition 7 ((ε, δ)-SDP[25]) Consider a system with n users, where each user u_i holds local data v_i (in the context of data streams, v_i can represent either a single data point or a substream of data). Let $\boldsymbol{R}$ denote the mechanism used for perturbing the data, $\boldsymbol{S}$ represent the shuffling operation, and $\boldsymbol{A}$ be the analysis function that computes summary statistics. An algorithm $K = (\boldsymbol{R}, \boldsymbol{S}, \boldsymbol{A})$ satisfies (ε, δ)-SDP if, for any output set $\mathcal{R}$, the following condition holds:

$$\Pr[\boldsymbol{S}(K) \in \mathcal{R}] \leqslant e^{\varepsilon} \cdot \Pr[\boldsymbol{S}(K') \in \mathcal{R}] + \delta,$$

where $\boldsymbol{S}(K)$ denotes the shuffled output of the mechanism. The parameters ε and δ control the privacy guarantees, with ε regulating the trade-off between privacy and utility, and δ capturing the potential leakage risk within the privacy mechanism.

3.3 Problem definition

Building on the system model and privacy definitions introduced above, This paper defines the histogram publication problem under SDP as follows:

Definition 8 (Problem definition) Given a data stream $\mathcal{D} = e_1, e_2, \ldots, e_t, \ldots$, and a predefined sliding window size w, the objective is to design an algorithm that computes a histogram for each sliding window W_t such that the published histogram satisfies (ε, δ)-SDP. Specifically, for each sliding window W_t, the goal is to compute an SDP-histogram $H(W_t)$ such that

$$\Pr[K(\mathcal{D}_t) \in \mathcal{R}] \leqslant e^{\varepsilon} \cdot \Pr[K(\mathcal{D}'_t) \in \mathcal{R}] + \delta,$$

where D_t and D'_t are neighboring data streams differing by at most one data point within the window, and $\mathcal{R}$ represents the set of histogram outputs. The challenge is to design an efficient mechanism for computing these histograms while maintaining the privacy guarantees of SDP.

4 Proposed SSDP Algorithm

This section provides an overview of the proposed SSDP algorithm, designed to address the histogram publication problem defined in Definition 8.

Firstly, this section introduces the core idea and specific steps of the SSDP algorithm. Then, a theoretical analysis is conducted to prove that SSDP can solve the problem efficiently and optimize the time and space complexities. In addition, this section explores the theoretical relationship between the sampling rate and the privacy budget, and establishes the equivalence between the sampling operation and the privacy protection mechanism.

4.1 Overview of SSDP algorithm

SSDP is designed with the objective of protecting the privacy of data streams while improving the efficiency in processing time and space. As shown in Fig. 2, SSDP encompasses three primary components: data sampling and perturbing via the OSSS mechanism, adaptive tuning of the sampling rate, and data aggregation.

Firstly, SSDP employs the OSSS mechanism for the purpose of extracting key information from the data stream. Subsequently, a hash-optimized GRR

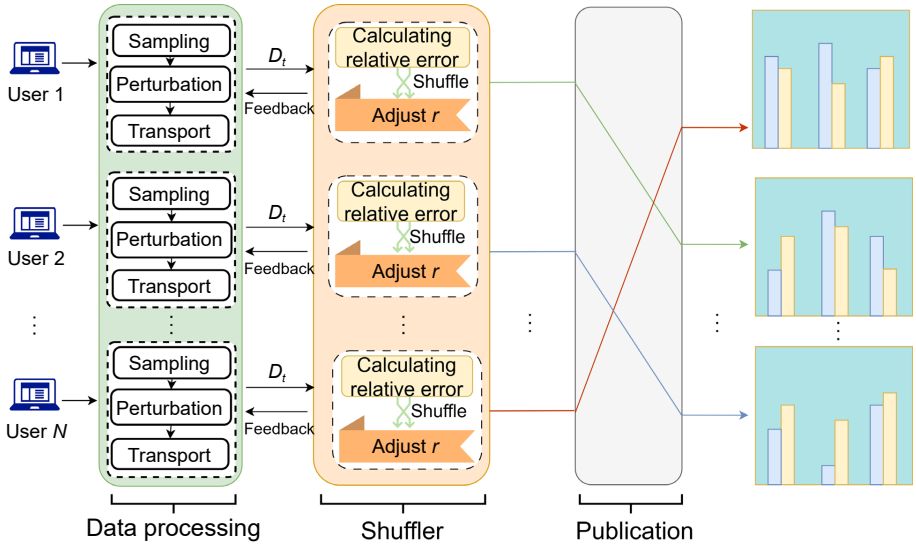

Fig. 2 SSDP processing.

technique is employed for generating perturbed data. Specifically, the large value domain is mapped to a smaller domain through a hash function in SSDP, thereby addressing the challenge where extensive value domains hinder effective privacy preservation. Concurrently, unbiased estimation correction is implemented under this processing framework to ensure accurate publication of approximate histograms.

Secondly, SSDP calculates the relative error and, based on the discrepancy between the current published result and the actual result, makes a dynamic adjustment to the sampling rate for the subsequent timestamp. This adaptive mechanism allows for a balance to be struck between data accuracy and computational efficiency.

Lastly, the data undergoes shuffling through a shuffler, which disrupts the linkage between users and their respective data. This step serves to further augment privacy protection, as tracing individual data contributions back to specific users becomes challenging. Following shuffling, the anonymity data is transferred to the publishing entity for aggregation.

In summary, SSDP integrates data sampling and perturbation, adaptive sampling rate adjustment, and data shuffling to provide a holistic solution for histogram publishing.

4.2 Key information sampling

The OSSS data structure is presented for the purpose of extracting essential information from data streams. This structure employs a sliding window technique to identify and capture the most pertinent and forthcoming w data segments. The objective of this approach is to facilitate efficient processing of these selected data elements via sampling. A detailed breakdown of the process is provided as follows:

Give a data stream $\mathcal{D} = e_1, e_2, \ldots, e_t, \ldots$ $(t \geqslant 1)$, let w denote the size of sliding window. The current window at timestamp t, denoted as W_t, consists of elements $(e_{t-w+1}, e_{t-w+2}, \ldots, e_t)$. Furthermore, let M represent the size of the sampling set, and the auxiliary space U of size $W+b$ holds elements with a triple $(\mathrm{ts}_i, \mathrm{id}_i, \mathrm{size}_i)$, Formally, the elements in U can be represented as a set of triples $\{(\mathrm{ts}_i, \mathrm{id}_i, \mathrm{size}_i)\}_{i=1}^{w+b}$ where ts_i is the timestamp of the element, id_i identifies the position within the auxiliary space, and size_i is the size of the data field used for GRR perturbation calculations.

For each element e_t in the data stream, the following requirements apply:

(1) If there are expired elements in $U[1]$ to $U[w+b]$, replace the oldest expired element with e_t, the specific structure is shown in Fig. 3.

(2) If there are no expired elements in $U[1]$ to $U[w+b]$, ignore e_t and wait for the next element.

(3) If a sampling set of size M needs to be selected from U, randomly choose M unexpired elements from U.

The sampling process is applied iteratively to each sliding window within the data stream, with the inclusion of a new element (denoted e_t) ensuring the currency and relevance of both the current window (denoted W_t) and the sampling set. To maintain accuracy, a comparison of the histogram results published at the current timestamp with the actual

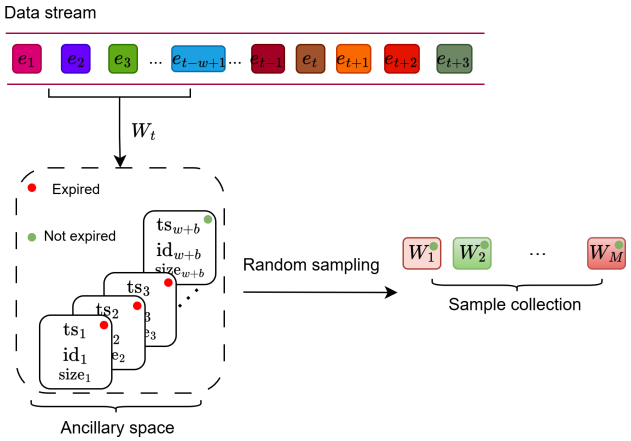

Fig. 3 OSSS processing.

results facilitates the calculation of the relative errors re and re_{prev}. Adjustments to the sampling rate r are then made based on the variation in the magnitudes of re and re_{prev}. Specifically, if the error value is within a predefined threshold $\mathcal{T}$, further analysis is performed. If re_{prev} is less than re, this indicates an increase in error, which may require adjustments to the sampling rate to mitigate the error. Conversely, if re_{prev} exceeds re, it indicates a reduction in error, allowing flexible adjustments to the sampling rate (which may involve increasing, maintaining, or decreasing it) to achieve a balance between error reduction and computational efficiency.

In summary, the OSSS incorporated in the SSDP algorithm integrates the sampling mechanism with the sliding window, and subsequently regulates the privacy budget to achieve the goal of error optimization by dynamically adjusting the sampling rate based on real time error analysis. This distinguishes the SSDP from existing methods and effectively improves spatio-temporal efficiency while controlling for error. The exact fitting procedures are outlined in Fig. 3.

4.3 Aggregator mathematical formula

On the client side (user side), suppose an attribute j is selected from the user data to be reported. Next, We use the hash transform to map it to the smaller value domain G, where $g=|G|$ denotes the size of the hash function's value domain, so that according to the decomposition idea provided in Ref. [27], for any output $y\in G$, we have (for convenience, note $H_y=(F(h(v))=y)$),

$$\Pr[H_y]=(1-\gamma)I_{h(v)=y}+\gamma\Pr[U(G)=y] \tag{2}$$

where $F(\cdot)$ is the GRR perturbation mechanism after hash optimization, $h(\cdot)$ is hash function, $I(\cdot)$ is an indicator function. The perturbation mechanism operates as follows: when $h(v)=y$, the true value is reported with probability $1-\gamma$, while with probability γ, a uniformly random value from G is reported instead. The parameter $\gamma=\dfrac{g+e^{\varepsilon}-1}{g}$ controls the privacy-utility trade-off, where ε is the privacy budget. The term $U(\cdot)$ stands for uniform distribution over G, with $\Pr[U(G)=y]=1/g$.

Theorem 1 For $j\in[1,d]$, If there are n users, the estimation result $\hat{f}_v$ provides an unbiased estimate of the true frequency f_v, expressed as

$$\hat{f}_v=\frac{1}{n}\cdot\frac{m_1 g-n\gamma(g-1)}{g-2\gamma(g-1)} \tag{3}$$

where m_1 denotes the number of users whose v is perturbed into y.

Proof The probability that v is perturbed into y is

$$\Pr[y]=f_v\left(1-\gamma\frac{g-1}{g}\right)+(1-f_v)\gamma\frac{g-1}{g},$$

and the probability that it is perturbed into y' is

$$\Pr[y']=f_v\gamma\frac{g-1}{g}+(1-f_v)\left(1-\gamma\frac{g-1}{g}\right).$$

The likelihood function for observing m_1 occurrences of y across n independent trials is

$$L(f_v)=\Pr[y]^{m_1}\times\Pr[y']^{n-m_1}.$$

The corresponding log-likelihood function becomes

$$\ln L(f_v)=m_1\ln\Pr[y]+(n-m_1)\ln\Pr[y'].$$

To find the maximum likelihood estimator, we take the partial derivative of $\ln L(f_v)$ with respect to f_v and set it to zero,

$$\frac{\partial\ln(L(f_v))}{\partial f_v}=m_1\frac{\left(1-2\gamma\dfrac{g-1}{g}\right)}{\Pr[y]}+(n-m_1)\frac{2\gamma\dfrac{g-1}{g}-1}{\Pr[y']}=0,$$

then we have

$$m_1\left(1-2\gamma\frac{g-1}{g}\right)\Pr[y']=(n-m_1)\left(1-2\gamma\frac{g-1}{g}\right)\Pr[y'].$$

Assuming $1-2\gamma\dfrac{g-1}{g}\neq 0$, we have

$$m_1\Pr[y']=(n-m_1)\Pr[y].$$

Substitut $\Pr[y]$ and $\Pr[y']$ in the above equation, we obtain

$$m_1\left[f_v\gamma\frac{g-1}{g}+(1-f_v)\left(1-\gamma\frac{g-1}{g}\right)\right]=$$

$$(n-m_1)\left[f_v\left(1-\gamma\frac{g-1}{g}\right)+(1-f_v)\gamma\frac{g-1}{g}\right],$$

which can be simplified to

$$m_1\left[1-\gamma\frac{g-1}{g}+f_v\left(2\gamma\frac{g-1}{g}-1\right)\right]=$$
$$(n-m_1)\left[\gamma\frac{g-1}{g}+f_v\left(1-2\gamma\frac{g-1}{g}\right)\right],$$

then we have

$$f_v\left[m_1\left(2\gamma\frac{g-1}{g}-1\right)+(n-m_1)\left(1-2\gamma\frac{g-1}{g}\right)\right]=$$
$$(n-m_1)\gamma\frac{g-1}{g}-m_1\left(1-\gamma\frac{g-1}{g}\right).$$

Further simply it,

$$f_v\left(1-2\gamma\frac{g-1}{g}\right)(n-2m_1)=n\gamma\frac{g-1}{g}-m_1.$$

Thus, the maximum likelihood estimator is:

$$\hat{f}_v=\frac{1}{n}\cdot\frac{m_1g-n\gamma(g-1)}{g-2\gamma(g-1)}.$$

To show that $\hat{f}_v$ is unbiased, we compute its expectation,

$$E[\hat{f}_v]=E\left[\frac{1}{n}\cdot\frac{m_1g-n\gamma(g-1)}{g-2\gamma(g-1)}\right]=$$
$$\frac{1}{g-2\gamma(g-1)}\left(\frac{g}{n}E[m_1]-\gamma(g-1)\right).$$

Since $E[m_1]=n\Pr[y]$ and $n\Pr[y]=n\left[f_v+\gamma\frac{g-1}{g}(1-2f_v)\right]$, we obtain

$$E[\hat{f}_v]=\frac{g\left[f_v+\gamma\frac{g-1}{g}(1-2f_v)\right]-\gamma(g-1)}{g-2\gamma(g-1)}=$$
$$\frac{f_v[g-2\gamma(g-1)]}{g-2\gamma(g-1)}=f_v.$$

■

Theorem 2 The variance of the unbiased estimate of $\hat{f}_v$ is $\mathrm{Var}[\hat{f}_v]=\dfrac{\gamma(g-1)(g-\gamma(g-1))}{n(g^2-4\gamma(g-1)g+4\gamma^2(g-1)^2)}$.

Proof According to the above variance formula,

$$\mathrm{Var}[\hat{f}_v]=\mathrm{Var}\left[\frac{1}{n}\cdot\frac{m_1g-n\gamma(g-1)}{g-2\gamma(g-1)}\right]=\frac{1}{n^2}\cdot\frac{g^2\mathrm{Var}[m_1]}{(g-2\gamma(g-1))^2}.$$

Since m_1 follows a binomial distribution with parameters n and $\Pr[y]$, we have $\mathrm{Var}[m_1]=n\Pr[y](1-\Pr[y])$. Noting that

$$\Pr[y](1-\Pr[y])=\gamma\frac{g-1}{g}\left(1-\gamma\frac{g-1}{g}\right),$$

so we have

$$\mathrm{Var}[\hat{f}_v]=\frac{1}{n^2}\cdot\frac{g^2\cdot n\cdot\gamma\frac{g-1}{g}\left(1-\gamma\frac{g-1}{g}\right)}{(g-2\gamma(g-1))^2}=$$
$$\frac{\gamma(g-1)(g-\gamma(g-1))}{n(g^2-4\gamma(g-1)g+4\gamma^2(g-1)^2)}.$$

This establishes the variance expression. ■

4.4 Privacy analysis

This subsection presents a formal demonstration that our algorithm satisfies shuffle differential privacy requirements. We establish the relationship between the sampling rate r and the privacy budget in SSDP, and provide an algorithm for adjusting the sampling rate based on this relationship.

To prove SSDP satisfies feasibility under SDP and streaming SDP settings, we verify (1) local perturbations satisfy LDP, (2) perturbed data meet w-event privacy under LDP, and (3) shuffled data guarantee DP at the release end. The proof framework follows Refs. [27, 28], treating sampling results as incomplete shuffles after data reduction, utilizing privacy blanket decomposition[27] and shuffle-perturbation order swapping[28].

Theorem 3 proves DP under the fungible mechanism, Theorem 4 shows w-event privacy in streaming settings, and Theorem 5 demonstrates hash-optimized GRR satisfies LDP. By differential privacy composition, SDP's privacy budget equals the sum of sampling and hash-optimized GRR budgets, confirming SSDP satisfies SDP.

Theorem 3 SSDP provides ε-DP for sampling and processing of collected data at each time t.

Proof Assuming the neighboring databases A and B have N_A and N_B records, respectively, where $N_A=N_B+1$. Randomly extract M records ($M\leqslant N_B$) from databases A and B with equal probability. The number of M-combinations from database A is

$$C_{N_A}^{M}=C_{N_B+1}^{M} \tag{4}$$

For database B, we have $C_{N_B}^{M}$. Identical sampling results occur only when the additional record in A is not selected, equivalent to selecting M records from the remaining N_B records ($C_{N_B}^{M}$). The probability ratio is

$$P=\frac{C_{N_B}^{M}}{C_{N_B+1}^{M}} \tag{5}$$

Using combinatorial identities, we have

$$C_{N_B+1}^{M}=\frac{(N_B+1)!}{M!(N_B+1-M)!} \tag{6}$$

$$C_{N_B}^{M} = \frac{N_B!}{M!(N_B - M)!} \tag{7}$$

Therefore, for $\varepsilon \geqslant 0$,

$$P = \frac{\dfrac{N_B!}{M!(N_B-M)!}}{\dfrac{(N_B+1)!}{M!(N_B+1-M)!}} = \frac{N_B! \times M!(N_B+1-M)!}{M!(N_B-M)! \times (N_B+1)!} = \frac{N_B+1-M}{N_B+1} \leqslant \mathrm{e}^{\varepsilon}.$$

So, SSDP satisfies ε-DP. ■

Theorem 4 SSDP provides w-event privacy.

Proof Due to the use of independent randomness in all mechanisms, for the time series Q and mechanism outputs $(r[1], r[2], \ldots, r[t]) \in \mathcal{R}$, our mechanism is abbreviated as $\mathcal{S}$ with independent perturbations $\Pr[P_k(q[k]) = r[k]]$ $(k \in [1, t])$,

$$\Pr[\mathcal{S}(Q) = (r[1], r[2], \ldots, r[t])] = \prod_{k=1}^{t} \Pr[P_k(q[k]) = r[k]].$$

Similarly, for any w-neighboring stream prefixes Q',

$$\Pr[\mathcal{S}(Q') = (r[1], r[2], \ldots, r[t])] = \prod_{k=1}^{t} \Pr[P_k(q'[k]) = r[k]].$$

According to Definition 3, there exits $i \in [1, t]$, such that $q[k] = q'[k]$ for $i \leqslant k \leqslant i - w$ and $i + 1 \leqslant k \leqslant t$. Thus,

$$\frac{\Pr[\mathcal{S}(Q) = (r[1], r[2], \ldots, r[t])]}{\Pr[\mathcal{S}(Q') = (r[1], r[2], \cdots, r[t])]} = \prod_{k=1}^{t} \frac{\Pr[P_k(q[k]) = r[k]]}{\Pr[P_k(q'[k]) = r[k]]}.$$

Form Definition 4, $q[k]$ and $q'[k]$ are neighboring with $i - w + 1 \leqslant k \leqslant i$. According to Theorem 3, our mechanism satisfies ε-DP, therefore, it can be deduced that

$$\frac{\Pr[\mathcal{S}(Q) = (r[1], r[2], \ldots, r[t])]}{\Pr[\mathcal{S}(Q') = (r[1], r[2], \ldots, r[t])]} \leqslant \prod_{k=i-w+1}^{t} \mathrm{e}^{\varepsilon_k} = \mathrm{e}^{\sum_{i-w+1}^{i} \varepsilon_k}.$$

So, for any $\mathcal{R}$ we have $\dfrac{\Pr[\mathcal{S}(Q) \in \mathcal{R}]}{\Pr[\mathcal{S}(Q') \in \mathcal{R}]} \leqslant \mathrm{e}^{\sum_{k=i-w+1}^{i} \varepsilon_k} \leqslant \mathrm{e}^{\varepsilon_1}$.

Thus SSDP satisfies w-event privacy. ■

Theorem 5 SSDP provides (ε, δ)-SDP.

Proof To prove that SSDP satisfies (ε, δ)-SDP, we need to show that it satisfies (ε, δ)-DP, which implies equivalent privacy under shuffling via post-processing.

By Theorem 4, sampling satisfies w-event privacy. We now demonstrate hash-optimized GRR satisfies differential privacy.

Following the framework in Ref. [27], let X denote the GRR perturbation mechanism after hash mapping, with output y. For any neighboring databases D and D', we prove

$$\Pr_{y \in X(D)}\left[\frac{\Pr[X(D) = y]}{\Pr[X(D') = y]} \geqslant \mathrm{e}^{\varepsilon_2}\right] \leqslant \delta.$$

Without loss of generality, we assume that D and D' differ only at user β, i.e., $v_\beta \neq v'_\beta$. After hash mapping, we can assume that $h(v_\beta)$ and $h(v'_\beta)$ are mapped to positions 1 and 2 in the domain, respectively.

Now, we consider the perturbation mechanism next. With probability $\gamma(g-1)/g$, users report a random value uniformly from G. In this case, the output probability distributions for D and D' are identical, so differential privacy is trivially satisfied.

With probability $1 - \gamma(g - 1/g)$, users report their true value. In this case, we must account for the shuffling process and the use of L hash functions. For a fixed permutation σ, the probability $\Pr[X(D) = y]$ can be expressed as

$$\sum_{\phi} \frac{1}{n!L}\left(\prod_{i \in [n-1]} I_{h_{\phi(i)}(v_i) = y_i} \times \prod_{i \in [n-1]} \frac{1}{g} I_{h_{\phi(n)}(v_n) = y_{\phi(n)}}\right).$$

and the probability $\Pr[X(D') = y]$ equals

$$\sum_{\phi} \frac{1}{n!L}\left(\prod_{i \in [n-1]} I_{h_{\phi(i)}(v_i) = y_i} \times \prod_{i \in [n-1]} \frac{1}{g} I_{h_{\phi(n)}(v'_n) = y_{\phi(n)}}\right),$$

thus

$$\frac{\Pr[X(D) = y]}{\Pr[X(D') = y]} = \frac{N_1}{N_2},$$

where $N_1 \sim \mathrm{Bin}(n-1, \gamma/g) + 1$ and $N_2 \sim \mathrm{Bin}(n-1, \gamma/g)$.

Therefore,

$$\Pr_{y \in X(D)}\left[\frac{\Pr[X(D) = y]}{\Pr[X(D') = y]} \geqslant \mathrm{e}^{\varepsilon_2}\right] = \Pr\left[\frac{N_1}{N_2} \geqslant \mathrm{e}^{\varepsilon_2}\right].$$

Let $c = E[N_2] = \dfrac{\gamma(n-1)}{k}$. Note that the event $\dfrac{N_1}{N_2} \geqslant \mathrm{e}^{\varepsilon_2}$ implies that at least one of the following events occurs: $N_1 \geqslant c\mathrm{e}^{\varepsilon_2/2}$ or $N_2 \geqslant c\mathrm{e}^{-\varepsilon_2/2}$. This is because if both $N_1 < c\mathrm{e}^{\varepsilon_2/2}$ and $N_2 < c\mathrm{e}^{\varepsilon_2/2}$ hold, then we would have $\dfrac{N_1}{N_2} < \mathrm{e}^{\varepsilon_2}$.

Hence, we have

$$\Pr\left[\frac{N_1}{N_2} \geqslant \mathrm{e}^{\varepsilon_2}\right] \leqslant \Pr[N_1 \geqslant c\mathrm{e}^{\varepsilon_2/2}] + \Pr[N_2 \geqslant c\mathrm{e}^{-\varepsilon_2/2}] = \Pr[N_2 - E[N_1] \geqslant c(\mathrm{e}^{\varepsilon_2/2} - 1 - 1/c)] + \Pr[N_2 - E[N_2] \leqslant c(\mathrm{e}^{-\varepsilon_2/2} - 1)].$$

Applying multiplicative Chernoff bounds, we have

$$\Pr\left[\frac{N_1}{N_2} \geqslant e^{\varepsilon_2}\right] \leqslant e^{-\frac{c}{3}(e^{\varepsilon_2/2}-1-\frac{1}{c})^2} + e^{-\frac{c}{2}(1-e^{-\varepsilon_2/2})^2}.$$

For $\varepsilon_2 \leqslant 1$, each term satisfies the bound when

$$c = \frac{\gamma(n-1)}{g} \geqslant \frac{14\ln(2/\delta)}{\varepsilon_2^2}.$$

Thus, hash-optimized GRR satisfies differential privacy with

$$\varepsilon' = \sqrt{\frac{14\ln(2/\delta)(e^{\varepsilon_2}+g-1)}{n-1}}.$$

By Theorem 4, SSDP satisfies real-time DP in streaming settings, and local perturbation satisfies ε'-DP. By composition, SSDP's privacy budget is $\varepsilon = \varepsilon_1 + \varepsilon'$, satisfying SDP. ■

Theorem 6 The time complexity of SSDP is $O(M)$.

Proof SSDP involves two main operation:

- For each new element, traverse auxiliary storage U to remove expired entries (linear in $|U|$);
- For each sampling, uniformly select S unexpired elements from U (depends on $|U|$ and random access efficiency).

Initial verification traverses ϑ lements with complexity $O(v)$. Sampling maintains complexity $O(M)$ when filtering unexpired elements first. Since $M = r\vartheta$ and $M < \vartheta$, total complexity $O(\vartheta + M) = O(M)$. ■

Theorem 7 The space complexity of SSDP is $O(w+\varrho)$.

Proof SSDP's space complexity primarily comes from OSSS auxiliary storage U of size $w+\varrho$, storing tuples $(\mathrm{ts}_i, \mathrm{id}_i, \mathrm{size}_i)$. Each tuple requires constant space λ, so total space is $\lambda \times (w+\varrho)$. Thus space complexity is $O(w+\varrho)$. ■

Theorem 8 Let $n \in \mathrm{N}$, $\delta \in (0, 1]$, $\varepsilon_0 \leqslant \log\left(\frac{n}{16\log(2/\delta)}\right)$. Consider $C \sim \mathrm{Bin}(n-1, e^{-\varepsilon_0})$, $J \sim \mathrm{Bin}(C, 1/2)$, and $\Delta \sim \mathrm{Bern}(r)$. Let $T_1 = (J+\Delta, C-J+1-\Delta)$ then T_1 and X_1 are (ε, δ)-indistinguishable for

$$\varepsilon \approx \log\left(1 + \frac{8re^{\varepsilon_0}}{n}\left(\frac{\sqrt{n\log(4/\delta)}}{\sqrt{e^{\varepsilon_0}}} + 1\right)\right) \tag{8}$$

Proof When $p \in [0, 1]$ and $n \in \mathbf{N}$ satisfy $p \geqslant \frac{16\ln(2/\delta)}{n}$, we consider $C \sim \mathrm{Bin}(n-1, p)$ and $J \sim \mathrm{Bin}(C, 1/2)$. Let $T_1 = (J+1, C-J)$ and $X_1 = (J, C-J+1)$.

By Chernoff bound with probability $\geqslant 1-\delta/2$, we have

$$|C - pn| \leqslant \sqrt{3pn\log(4/\delta)} \tag{9}$$

By Hoeffding's inequality with probability $\geqslant 1-\delta/2$, we have

$$|J - C/2| \leqslant \sqrt{\frac{C}{2}\log(4/\delta)} \tag{10}$$

Both bounds hold simultaneously with probability $\geqslant 1-\delta$.

Given a specific output (a, b), let $K' = \Pr(C = a+b-1)$, we have

$$\Pr(T_1 = (a, b)) = K' \cdot \Pr(J = a-1 | C = a+b-1) = K' \cdot \frac{a}{b} \cdot \Pr(J = a | C = a+b-1) = \frac{a}{b}\Pr(X_1 = (a, b)).$$

If J and C satisfy Formulas (9) and (10),

$$\frac{a}{b} = \frac{J+1}{C-J} \leqslant \frac{C/2 + \sqrt{C/2\log(4/\delta)} + 1}{C/2 - \sqrt{C/2\log(4/\delta)}} \leqslant 1 + \frac{\sqrt{32\log(4/\delta)}}{\sqrt{C}} + \frac{4}{C} \leqslant 1 + \frac{\sqrt{32\log(4/\delta)}}{\sqrt{pn - \sqrt{3pn\log(4/\delta)}}} + \frac{8}{pn} \leqslant 1 + \frac{8}{pn}(\sqrt{pn\log(4/\delta)} + 1).$$

Similarly,

$$\frac{b}{a} = \frac{C-J}{J+1} \leqslant \frac{C/2 + \sqrt{C/2\log(4/\delta)}}{C/2 - \sqrt{C/2\log(4/\delta)} + 1} \leqslant \frac{C/2 + \sqrt{C/2\log(4/\delta)} + 1}{C/2 - \sqrt{C/2\log(4/\delta)}} \leqslant 1 + \frac{8}{pn}(\sqrt{pn\log(4/\delta)} + 1).$$

So, if we define $\varepsilon = \log(1 + \frac{8}{pn}(\sqrt{pn\log(4/\delta)} + 1))$, then T_1 and X_1 are (ε, δ)-indistinguishable.

Let $T_0 = (J+1, C-J)$ and $X_0 = (J, C-J+1)$, with ρ_0 and ρ_1 denoting their respective probability distributions. When $p = e^{-\varepsilon_0}$, the following privacy guarantees hold for the following two distributions z_1 and z_2:

$$\forall z_1 \sim \rho_0 : \Pr\left(-\varepsilon \leqslant \ln\frac{\Pr(T_0 = z_1)}{\Pr(X_0 = z_2)} \leqslant \varepsilon\right) \geqslant 1-\delta,$$

$$\forall z_2 \sim \rho_1 : \Pr\left(-\varepsilon \leqslant \ln\frac{\Pr(T_0 = z_1)}{\Pr(X_0 = z_2)} \leqslant \varepsilon\right) \geqslant 1-\delta.$$

Let $\varphi = \frac{1}{2}(T_0 + X_0)$, with a sub-sampling rate r, we can obtain

$$T_1 = (1-r)\varphi + rT_0,$$

$$X_1 = (1-r)\varphi + rT_0.$$

Using hockey-stick divergence, we have

$$D_{e^{\varepsilon'}}(\Phi\|\Psi) = E_{x\sim\Psi}\left[\max\left(\frac{\Phi(x)}{\Psi(x)} - e^{\varepsilon'},\ 0\right)\right].$$

For intermediate distribution φ, we have

$$\max\left(D_{e^{\varepsilon'}}(\varphi\|T_0),\ D_{e^{\varepsilon'}}(X_0\|\varphi)\right) \leqslant D_{e^{\varepsilon'}}(T_0\|X_0) \tag{11}$$

By Theorem 4, T_1 and X_1 are $(\varepsilon,\ p\delta)$-indistinguishable with

$$\varepsilon \leqslant \log\left(1 + \frac{8}{pn}\left(\sqrt{pn\log(4/\delta)} + 1\right)\right).$$

Substituting $p = e^{-\varepsilon_0}$ and including Bernoulli parameter r, we have

$$\varepsilon \approx \log\left(1 + \frac{8re^{\varepsilon_0}}{n}\left(\frac{\sqrt{n\log(4/\delta)}}{\sqrt{e^{\varepsilon_0}}} + 1\right)\right).$$

■

Since privacy budget correlates with published histogram accuracy (smaller ε indicates better protection), sampling rate-privacy budget relationship (Theorem 8) enables indirect histogram error optimization via sampling rate adjustment. Algorithm 1 details this process, with key symbols explained in Table 1.

Algorithm 1 SSDP

Input: sampling rate r, window size w, privacy budget ε, error threshold T, $\mathrm{re}_{\mathrm{prev}}$, optimization parameter ϖ

Output: Released statistics $H_t = (r_1, r_2, \ldots, r_t, \ldots)$

Initialize $r_0 = \langle 0, 0, \ldots, 0\rangle^d$;
for each timestamp t **do**
  $D_t \leftarrow$ User's data are collected in a sliding window with size w;
  $D_s \leftarrow D_t$;
  D_s is perturbed by the hash-optimized GRR to obtain $\overline{D_t}$, which uses a privacy budget obeying Formula (8);
  Shuffle processing;
  Estimate r_t by Eq. (3);
  Calculate $\mathrm{re} = \dfrac{|r_t - r_{\mathrm{real}}|}{r_{\mathrm{real}}}$;
  if $\mathrm{re} \leqslant T$ and $\mathrm{re} \leqslant \mathrm{re}_{\mathrm{prev}}$ **then**
    $r = r - \varpi$ or maintain sampling rate r;
  else if $\mathrm{re} \leqslant T$ and $\mathrm{re} > \mathrm{re}_{\mathrm{prev}}$ **then**
    $r = r + \varpi$;
  else
    $r = r + \varpi$;
  end if
end for

Table 1 Important symbols and meaning.

Notation	Description
r_t	Estimation histogram at timestamp t
r_{real}	Histogram without error at timestamp t
d	Histogram dimension
H_t	Histogram publishing task that continues over time
D_t	Raw data
D_s	Raw data after sampling
$\overline{D_t}$	Result obtained by perturbing D_s using Eq. (2)
re	Relative error
$\mathrm{re}_{\mathrm{prev}}$	Relative error when the histogram is last published

5 Experiment

This section evaluates the performance of SSDP on three real-world time series datasets. The focus of the evaluation is on SSDP's processing speed for large-scale data, examined from two perspectives: (1) assessing its capacity to manage extensive data volumes; and (2) evaluating the usability of the results produced by SSDP. It should be noted that the optimization pursued in this study emphasizes time efficiency, with an effort to maintain alignment with the current state-of-the-art in terms of utility.

5.1 Evaluation settings

(1) Datasets setting

The three actual datasets are as follows:

- **Taxi[29]:** Real time trajectory of 10 357 taxis in Beijing from February 2 to February 8, 2008.
- **IPUMS[30]:** The US Census data for the year 1940. We sample 1% of users, and use the city attribute. This results in a total of 602 325 users and 915 cities.
- **Kosarak[31]:** A dataset of 1×10^6 click streams on a Hungarian website that contains around one million users with 42 178 possible values. For each stream, one item is randomly chosen.

(2) Metrics

This paper use the estimated MSE as a metric. For each value v, calculate its estimated frequency $\tilde{f}_v$ and the ground truth value f_v, and calculate their squared difference. Specifically, $\mathrm{MSE} = \dfrac{1}{|\mathcal{L}|}\sum_{v\in\mathcal{L}}(f_v - \tilde{f}_v)^2$.

(3) Comparing methods

The experiment compares SSDP with the following five methods:

- **SOLH[32]:** This hash-based method strikes a balance between computation and communication. While it requires more computation than unary

encoding methods, it reduces overall communication bandwidth.

- **SH[27]:** It is a shuffler-based method for histogram estimation.
- **mixDUMP[33]:** The method sends random data while the user sends the perturbation value, improving the accuracy of posting in multi-message mode.
- **RAP[32]:** RAP's local method mirrors RAPPOR, which uses hash functions in a Bloom filter to process strings.
- **DDRM[34]:** The Dynamic Differential privacy Reporting Mechanism (DDRM) for continuous frequency estimation under LDP, which introduces a difference tree to capture data changes over time, can be a good solution to the problem of potential privacy leakage when data change.

5.2 Frequency estimation

The utility performance of SSDP is initially evaluated through a comparative analysis with alternative methods within the shuffle model, including SH, SOLH, and other relevant approaches.

Figures 4–6 show the fluctuation of MSE values for a series of algorithms, including SH, mixDUMP, DDRM, and RAP on the IPUMS, Taxi, and Kosarak

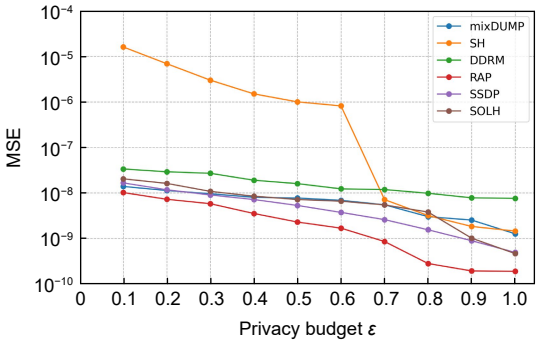

Fig. 4 MSE comparison based on IPMUS datasets.

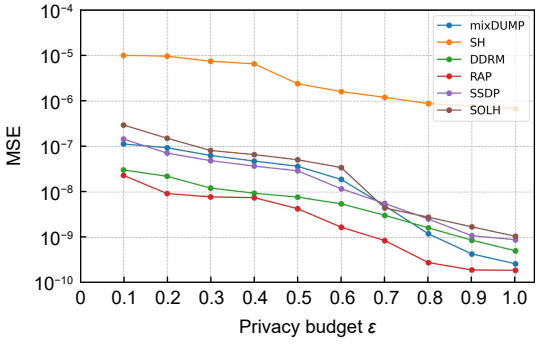

Fig. 5 MSE comparison based on Kosarak datasets.

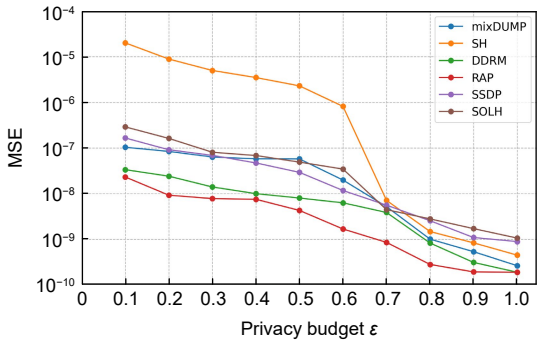

Fig. 6 MSE comparison based on Taxi datasets.

datasets. The experimental results show that when the privacy budget is adjusted from 0.1 to 1, the values of MSE of all six methods show a decreasing trend. This observation can be attributed to the fact that as the privacy budget increases, the noise introduced by the perturbation mechanism decreases, resulting in a decrease in MSE value. However, it is worth noting that as shown in Figs. 4–6, when the privacy budget is below 0.5, SH cannot enhance privacy protection at all. However, as shown in Fig. 5, when the privacy budget exceeds 0.5, the MSE of SH significantly decreases. This phenomenon can be attributed to the relatively large value range of the Kosarak dataset, which hinders the potential benefits of SH privacy amplification, as the local perturbation used by SH is a random response mechanism, which is a method strongly influenced by the value range. Specifically, when the value range is too large, the probability p of the original value being perturbed to other values significantly decreases. Therefore, for publishers, as the privacy budget increases, even after shuffling, the received data are mostly the original data, resulting in improved accuracy (i.e., reduced MSE), although SH is difficult to protect privacy. The RAP algorithm shows the most favorable overall performance on all three datasets, displaying a lower MSE. However, it relies on lasso regression, which benefits MSE but also introduces limitations, such as increased communication costs, and can only be used when compatible with LDP settings removal.

In addition, compared to the state-of-the-art mixDUMP and DDRM methods, the overall performance of our SSDP algorithm is slightly inferior, although this difference is not statistically significant. This slight disadvantage can be attributed to the inherent sampling rate manipulation of the SSDP method, which can result in slight fluctuations in error.

Although this may result in a slight decrease in the accuracy of the final output, it aligns with our primary goal of achieving real-time processing and ensuring user friendliness. As long as the gap with the most advanced methods remains small, this method is sufficient for handling tasks involving large-scale data streams.

5.3 Performance experiment

This experiment evaluates the computational and communication costs associated with SSDP, DDRM, and related protocols, which include various overheads incurred during the histogram publishing process. The results are summarized in Table 2.

Table 2 presents the communication efficiency of various comparative data stream methods in real datasets prepared. Specifically, SSDP demonstrates a 40% improvement in computational efficiency compared to SOLH and SH, and a two-fold increase in speed relative to RAP. This performance superiority can be attributed to the fact that RAP employs a mechanism on the local side similar to Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR), which relies on joint estimation decoding of user data. Consequently, each iteration in RAP is dependent on the previous iteration and the entire datasets, leading to potential excessive consumption of memory and computational resources, especially in scenarios with a large number of users. In contrast, DDRM, which incorporates additional processing through a difference tree structure, and mixDUMP, which increases the number of shuffles in the shuffle amplification process for enhanced privacy protection, exhibit slightly higher communication overheads. However, they also perform slightly worse than SSDPs in terms of time efficiency.

So, based on experimental results, it can be demonstrated that after following the OSSS sampling process, the SSDP algorithm efficiently processes critical information, proving its adaptability to the modern data stream environment.

Table 2 Comparison of communication overhead.

(s)

Dataset	Method					
	SSDP	SOLH	SH	RAP	mixDUMP	DDRM
Taxi	8.52	13.81	12.63	34.84	11.62	10.88
IPUMS	0.55	0.96	0.82	2.26	0.76	0.74
Kosarak	7.14	12.35	1.52	29.03	10.28	9.12

6 Conclusion

This paper introduces SSDP, a novel algorithm for privacy-preserving histogram publication in data streams, which integrates OSSS sampling with shuffling techniques to provide both efficiency and strong privacy guarantees. Theoretical analysis demonstrates that SSDP is capable of quickly generating histograms that satisfy SDP. Experimental results further validate that SSDP achieves high time efficiency while maintaining robust privacy protection, performing on par with state-of-the-art methods. In future work, we will explore adaptive sliding window strategies to further improve the performance of histogram publishing.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (No. 62441207), the University Synergy Innovation Program of Anhui Province (No. GXXT-2023-021), and the Natural Science Research Project of Anhui Educational Committee (No. 2022AH040052).

References

[1] Z. Qin, J. Weng, Y. Cui, and K. Ren, Privacy-preserving image processing in the cloud, *IEEE Cloud Comput.*, vol. 5, no. 2, pp. 48–57, 2018.

[2] G. Acs, C. Castelluccia, and R. Chen, Differentially private histogram publishing through lossy compression, in *Proc. 2012 IEEE 12th Int. Conf. Data Mining*, Brussels, Belgium, 2012, pp. 1−10.

[3] V. Balcer and A. Cheu, Separating local & shuffled differential privacy via histograms, in *Proc. 1st Conf. Information-Theoretic Cryptography*, arXiv prepritn arXiv:1911.06879, 2020.

[4] W. Qardaji, W. Yang, and N. Li, Understanding hierarchical methods for differentially private histograms, *Proc. VLDB Endow.*, vol. 6, no. 14, pp. 1954–1965, 2013.

[5] Q. Zhang, X. Zheng, and X. Wang, An efficient online multiparty interactive medical prediagnosis scheme with privacy protection, *Wireless Commun. Mobile Comput.*, vol. 2021, no. 1, p. 7746286, 2021.

[6] S. Le Blond, C. Zhang, A. Legout, K. Ross, and W. Dabbous, I know where you are and what you are sharing: Exploiting P2P communications to invade users' privacy, in *Proc. 2011 ACM SIGCOMM Conf. Internet Measurement Conf.*, Berlin, Germany, 2011, pp. 45−60.

[7] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld, Prochlo: Strong privacy for analytics in the crowd, in *Proc. 26th Symp. Operating Systems Principles*, Shanghai, China, 2017, pp. 441−459.

[8] Ú. Erlingsson, V. Pihur, and A. Korolova, RAPPOR: Randomized aggregatable privacy-preserving ordinal response, in *Proc. 2014 ACM SIGSAC Conf. Computer and Communications Security*, Scottsdale, AZ, USA, 2014, pp. 1054–1067.

[9] C. Dwork, Differential privacy, in *Proc. 33rd Int. Colloquium on Automata, Languages and Programming*, Venice, Italy, 2006, pp. 1–12.

[10] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, Differentially private histogram publication, *VLDB J.*, vol. 22, no. 6, pp. 797–822, 2013.

[11] G. Kellaris and S. Papadopoulos, Practical differential privacy via grouping and smoothing, *Proc. VLDB Endow.*, vol. 6, no. 5, pp. 301–312, 2013.

[12] A. Cheu and M. Zhilyaev, Differentially private histograms in the shuffle model from fake users, in *Proc. 2022 IEEE Symp. Security and Privacy*, San Francisco, CA, USA, 2022, pp. 440–457.

[13] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, Differential privacy under continual observation, in *Proc. 42nd ACM Symp. Theory of Computing*, Cambridge, MA, USA, 2010, pp. 715–724.

[14] T. H. H. Chan, E. Shi, and D. Song, Private and continual release of statistics, *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 3, p. 26, 2011.

[15] C. Dwork, Differential privacy in new settings, in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms*, Austin, TX, USA, 2010, pp. 174–183.

[16] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft, Private decayed predicate sums on streams, in *Proc. 16th Int. Conf. Database Theory*, Genoa, Italy, 2013, pp. 284–295.

[17] Y. Chen, A. Machanavajjhala, M. Hay, and G. Miklau, PeGaSus: Data-adaptive differentially private stream processing, in *Proc. 2017 ACM SIGSAC Conf. Computer and Communications Security*, Dallas, TX, USA, 2017, pp. 1375–1388.

[18] M. Xu, B. Ding, T. Wang, and J. Zhou, Collecting and analyzing data jointly from multiple services under local differential privacy, *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 2760–2772, 2020.

[19] V. Perrier, H. J. Asghar, and D. Kaafar, Private continual release of real-valued data streams, in *Proc. 26th Annu. Network and Distributed System Security Symp.*, San Diego, CA, USA, 2019, pp. 1–13.

[20] X. Ren, S. Wang, X. Yao, C. M. Yu, W. Yu, and X. Yang, Differentially private event sequences over infinite streams with relaxed privacy guarantee, in *Proc. 14th Int. Conf. Wireless Algorithms, Systems, and Applications*, Honolulu, HI, USA, 2019, pp. 272–284.

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

[22] Q. Ye, H. Hu, N. Li, X. Meng, H. Zheng, and H. Yan, Beyond value perturbation: Local differential privacy in the temporal setting, in *Proc. IEEE Conf. Computer Communications*, Vancouver, Canada, 2021, pp. 1–10.

[23] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, and H. Shin, Collecting and analyzing multidimensional data with local differential privacy, in *Proc. 2019 IEEE 35th Int. Conf. Data Engineering*, Macao, China, 2019, pp. 638–649.

[24] Z. Wang, W. Liu, X. Pang, J. Ren, Z. Liu, and Y. Chen, Towards pattern-aware privacy-preserving real-time data collection, in *Proc. IEEE Conf. Computer Communications*, Toronto, Canada, 2020, pp. 109–118.

[25] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, Distributed differential privacy via shuffling, in *Proc. 38th Annu. Int. Conf. Theory and Applications of Cryptographic Techniques*, Darmstadt, Germany, 2019, pp. 375–403.

[26] T. Wang, J. Blocki, N. Li, and S. Jha, Locally differentially private protocols for frequency estimation, in *Proc. 26th USENIX Conf. Security Symp.*, Vancouver, Canada, 2017, pp. 729–745.

[27] B. Balle, J. Bell, A. Gascón, and K. Nissim, The privacy blanket of the shuffle model, in *Proc. 39th Annu. Int. Cryptology Conf. Advances in Cryptology*, Santa Barbara, CA, USA, 2019, pp. 638–667.

[28] V. Feldman, A. McMillan, and K. Talwar, Stronger privacy amplification by shuffling for Rényi and approximate differential privacy, in *Proc. 2023 Annu. ACM-SIAM Symp. Discrete Algorithms*, Florence, Italy, 2023, pp. 4966–4981.

[29] J. Yuan, Y. Zheng, X. Xie, and G. Sun, Driving with knowledge from the physical world, in *Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Diego, CA, USA, 2011, pp. 316–324.

[30] S. Ruggles, S. Flood, R. Goeken, J. Grover, E. Meyer, J. Pacas, and M. Sobek, Integrated Public Use Microdata Series: Version 9.0 [dataset], http://doi.org/10.18128/D010.V9.0, 2019.

[31] Frequent itemset mining dataset repository, http://fimi.ua.ac.be/data/, 2025.

[32] T. Wang, B. Ding, M. Xu, Z. Huang, C. Hong, J. Zhou, N. Li, and S. Jha, Improving utility and security of the shuffler-based differential privacy, *Proc. VLDB Endow.*, vol. 13, no. 13, pp. 3545–3558, 2020.

[33] X. Li, W. Liu, H. Feng, K. Huang, Y. Hu, J. Liu, K. Ren, and Z. Qin, Privacy enhancement via dummy points in the shuffle model, *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 3, pp. 1001–1016, 2024.

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

Xiao Zheng received the BEng degree from Anhui University, China in 1997, the MEng degree from Zhejiang University of Science and Technology, China in 2003, and the PhD degree in computer science and technology from Southeast University, China in 2014. He is currently a professor at School of Computer Science and Technology, Anhui University of Technology, Anhui, China. He is also a guest professor at Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, China. His research interests include service computing, mobile cloud computing, and privacy protection. He is a senior member of CCF, and a member of IEEE and ACM.

Xiujun Wang received the BEng degree in computer science and technology from Anhui Normal University, China in 2005, and the PhD degree in computer software and theory from University of Science and Technology of China, China in 2011. He is currently a lecturer at School of Computer Science and Technology, Anhui University of Technology, China. His research interests include data stream processing, randomized algorithms, and the Internet of Things (IoTs).

Yiming Jin received the BEng degree from Ma'anshan University, China in 2020. He is currently a master student at Anhui University of Technology, China. His main research interests include differential privacy and data security.

Zixia Liu received the BS and MS degrees in mathematics from Jilin University, China in 2007 and 2009, respectively, and the PhD degree in computer science from University of Central Florida, FL, USA in 2022. He is currently a faculty member at School of Computer Science, Anhui University of Technology, China. His research interests include distributed computing, deep learning, and cybersecurity.