Optimal Locally Private Data Stream Analytics

Shaowei Wang[ab], Yun Peng[a*], Kongyang Chen[ab*], Wei Yang[c]
[a] *Institute of Artificial Intelligence, Guangzhou University*, Guangzhou, China
[b] *Guangdong Provincial Key Laboratory of Blockchain Security*, Guangzhou, China
[c] *Hefei National Laboratory, University of Science and Technology of China*, Hefei, China
{wangsw,yunpeng,kychen}@gzhu.edu.cn, qubit@ustc.edu.cn

***Abstract*—Online data analytics with local privacy protection is widely adopted in real-world applications. Despite numerous endeavors in this field, significant gaps in utility and functionality remain when compared to its offline counterpart. This work demonstrates that private data analytics can be conducted online without excess utility loss, even at a constant factor. We present an optimal, streamable mechanism for local differentially private sparse vector estimation. The mechanism enables a range of online analytics on streaming binary vectors, including multi-dimensional binary, categorical, or set-valued data. By leveraging the negative correlation of occurrence events in the sparse vector, we attain an optimal error rate under local privacy constraints, only requiring streamable computations during the input's data-dependent phase. Through experiments with both synthetic and real-world datasets, our proposals have been shown to reduce error rates by 40% to 60% compared to SOTA approaches.**

***Index Terms*—local differential privacy; streaming data; data aggregation; minimax optimality**

I. Introduction

Streaming user data analytics, which tracks user behaviors and status over time, plays a pivotal role in online decision-making and service quality improvement (e.g., for healthcare [1], location-based services [2], and other Internet services [3], [4]). However, it faces significant privacy challenges. Temporal user data, including demographic (e.g., ages, locations) and activity data (e.g., sensor readings, visited pages/Apps), can expose sensitive personal information. Concurrently, privacy-related regulations, such as GDPR in the European Union [5], CCPA in California [6], and the Personal Information Protection Law in China [7], are being globally enforced.

Local differential privacy (LDP [8], [9]) is frequently utilized for streaming user data analytics in industry (e.g., by Google Chrome [3], and operating systems from Microsoft [4] and Apple [10], [11]) where servers are untrustable. Unlike the well-studied sporadic/offline data collection (see [12], [13] for reviews), the online variant offers several distinct benefits but also introduces additional challenges:

(1) **Cumulative privacy loss.** The total privacy leakage increases with the number of reports. Adversaries can leverage multiple reports for more accurate inference [14].
(2) **Real-time response.** The strength of online analytics lies in real-time feedback, which necessitates that privatization mechanisms are streamable. In contrast, offline mechanisms observe the complete input.

A straightforward approach involves dividing the local privacy budget ϵ into T parts and independently sanitizing data at each timestamp $t \in [T]$. However, this results in a mean squared error scaling at $O(T^3)$, hiding the $1/\epsilon^2$ factor. Recent research notes that user data often exhibit limited changes over time [15], prompting proposals to leverage change sparsity. This strategy treats streaming data as *sparse ternary vectors* reflecting temporal changes, which are then processed by streamable, locally private mechanisms that independently sanitize multiple dimensions or non-zero entries. This approach reduces the error factor to $O(T^2 \log^2 T)$ (e.g., in [16], [17]) or $O(Ts^2 \log^2 T)$ (e.g., in [15], [18], [19]), where s denotes the number of changes in a stream (i.e. a sparsity parameter).

While online locally private data analytics has seen significant advancements, it still lags its offline counterpart in terms of utility performance and functionality. **I)** Regarding the key sub-problem of locally private ternary vector estimation in offline settings, recent work [23] proposes mapping all non-zero entries with random weights to one bucket, then adding noise to the sum of the weights, achieving mean squared error bound of $O(Ts \log n/\epsilon^2)$. Another recent work [22] achieves minimax lower error bound of $O(Ts/\epsilon^2)$. These offline approaches surpass current online streamable mechanisms by a factor of T/s or s. **II)** Existing online private mechanisms typically consider mean estimation on binary/categorical values, while offline studies accommodate diverse analytic tasks (e.g., range queries [24], [25]) on multi-dimensional vectors. Real-world user data often takes multi-dimensional forms, as demographic and activity data are jointly collected to study occurrences. The server/statistician may perform range queries, offering time-based summary statistics (e.g., App usage frequency within a week) and facilitating advanced temporal analytics.

A. Our Contributions

We propose an optimal protocol for online locally private data analytics, capable of handling multi-dimensional data and various analytic tasks, including mean/frequency and range queries. Contrary to the pessimistic results of earlier studies, our findings suggest that online private analytics does not incur additional utility losses than offline approaches.

To handle general multi-dimensional vectors of user data, we consider each user holds a streaming binary vector $\{0,1\}^{d\times T}$, with each datum $\{0,1\}^d$ at timestamp $t \in [T]$

*Corresponding authors. This work is supported by National Key Project of China (No. 2020YFB1005700), NSFC No.62372120, 62102108, 61802383, 62172385, 62172117, U23A20307, U1936218, U21A20463, NSF of Guangdong Province No.2022A1515010061, 2023A1515030273, and Guangzhou Research Foundation (No.202201010194, No.622191-098).

TABLE I

COMPARISON OF ϵ-LDP MECHANISMS FOR SPARSE TERNARY VECTOR ESTIMATION, FEATURING d' DIMENSIONS AND s NON-ZERO ENTRIES. THE *comput.* COLUMN REPRESENTS COMPUTATIONAL COSTS, THE *comm.* COLUMN INDICATES COMMUNICATION COSTS, AND THE **MSE** COLUMN PRESENTS MEAN SQUARED ERROR BOUNDS. THE **ONLINE** COLUMN SPECIFIES WHETHER THE MECHANISM CAN BE EXPLICITLY MADE ONLINE.

approaches	**user comput.**	**user-server comm.**	**server comput.**	**MSE**$\times(n\epsilon^2)$	**optimal?**	**online?**
PrivKV [17]	$O(\log d')$	$O(\log d')$	$O(n)$	$O(d'^2)$	✗	✓
KVUE [20]	$O(\log d')$	$O(\log d')$	$O(n)$	$O(d'^2)$	✗	✓
PCKV-GRR [18]	$O(\log d')$	$O(\log d')$	$O(n)$	$O(d'^2)$	✗	✓
PCKV-UE [18]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
EFMRTT [15]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
ToPL [21]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
DDRM [19]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
Collision [22]	$O(s)$	$O(\log s)$	$O(nd')$	$O(d's)$	✓	✗
SUCCINCT [23]	$O(s)$	$O(\log s)$	$O(nd')$	$O(d's\log n)$	✓	✗
this work	$O(d')$	$O(\min(d', d'/s\log d')$	$O(nd')$	$O(d's)$	✓	✓

encompassing binary/categorical/set-valued data. We represent the streaming data as hierarchical ternary vectors $\{-1, 0, 1\}^{d'}$. These vectors indicate temporal changes/residues at varying granularities. We sanitize one of the ternary vectors in the hierarchical tree online with the newly proposed ExSub mechanism. The ExSub mechanism utilizes the information of the whole vector (instead of one or few dimension(s) as in [15]–[19]) and exploit the negative correlation of two options $\{-1, 1\}$ in each entry, which never coexist in the input, and outputs an **ex**clusive **sub**set of entries. This minimizes estimation error and aligns with the minimax error lower bound. Moreover, the ExSub mechanism solely uses streamable primitives, and the exclusive subset can be sampled via a recursive implementation of *sequential random sampling* [26]. The server then reconstructs the statistics of interest from ternary vector estimators. By considering the importance of each hierarchical level for a particular task (i.e. mean value/frequency or range query), we provide calibrated hierarchy selection weights to further reduce estimation error. We present the key properties of our proposal in Theorem 1, and compare it with existing approaches for the core subproblem of ternary vector aggregation in Table I.

Theorem 1: Consider n users, each with $\mathbf{x}_{i,t} \in \{0, 1\}^d$ at timestamp $t \in [T]$. Assuming there are at most s changes across T timestamps for each user, then there exists a streamable local ϵ-differential private mechanism that provides an unbiased estimator of $\{\frac{1}{n}\sum_{i\in[n]} \mathbf{x}_{i,t}\}_{t\in[T]}$. The mechanism has mean squared error of $O\big(dTs\log^2 T/(n\epsilon^2)\big)$, incurs $O(d)$ memory consumption, $O(dT)$ computational costs, and $O\big(dT/s\big)$ communication costs on the user side.

In summary, this work offers the following contributions:

- We introduce a generic protocol for online locally private data analytics, accommodating a broad range of data types (e.g., binary, categorical, set-valued data, and multi-dimensional vector) and analytic tasks (including mean estimation, frequency estimation, and range queries).
- We propose the ExSub mechanism, an optimal locally private solution for sparse ternary vector aggregation. It works as a core building block of our analytics protocol. This mechanism optimizes data utility while relying solely on streamable computations. In comparison with existing SOTA offline mechanisms, it reduces the error by 10%-30%.
- We devise optimized hierarchy selection strategies to simultaneously minimize user-side computational/communication costs and maximize server-side estimation utility.
- Through extensive experiments on both synthetic and real-world datasets, we demonstrate that our proposals reduce the error by 40%-60% compared with existing online protocols.

B. Organization

The remaining paper is organized as follows. Section II reviews related works. Section III provides background information and problem formulation. Section IV presents the protocol for online locally private data analytics. Section V details the design of the ExSub mechanism and provides both offline and online implementations. Section VI reports experimental results. Finally, Section VII concludes the paper.

II. RELATED WORKS

A. Online Locally Private Protocols

RAPPOR [3] utilizes a Bloom-filter and randomized response [27] for user data encapsulation and sanitization. It uses a memoization strategy, reusing private views until data changes. For ongoing data streams, Ding *et al.* [4] suggest discretizing local values to reduce temporal changes, and employ randomized response for sanitizing. Alternatively, Joseph *et al.* [28] segment users into groups each with a similar report pattern, independently sanitizing each report. Though these methods [3], [4], [28] use memoization to lessen privacy consumption in continuous collection, they ensure only a weaker local differential privacy form specific to identical change pattern domains. Yet, these change patterns, as sensitive user information, necessitate stringent protection.

For rigorous LDP protection in streaming online data analytics, a direct method involves splitting the privacy budget into T parts and sanitizing each value at every timestamp. Erlingsson *et al.* [15] present a hierarchical tree structure of height $O(\log T)$ to record local data temporal changes, sanitizing one $\{-1, 1\}$ change with the full budget ϵ, which improves the estimation error bound from a factor of $O(T^3)$ to $O(Ts^2\log^2 T)$. Wang *et al.* [21] study online numerical data collection and propose to sanitize each clipped value independently, leading to a mean squared error that scales with $O(Ts^2\log^2 T)$.

Later, the DDRM [19] selects k changes from all s changes and sanitizes each selected change with budget $\frac{\epsilon}{k}$, the error also scales with $O(Ts^2 \log^2 T)$. A theoretical proposal by Ohrimenko *et al.* [29] suggests randomizing every value with binary randomized response while truncating sensitive outputs to save budget to $O(\sqrt{s})$. However, their proposal, while theoretically interesting, exhibits poor empirical performance (see Section VI). In contrast, our proposal achieves an optimal error rate of $O(Ts \log^2 T)$ and accommodates universal binary vector streams (or slowly changing vector streams), including set-valued streams and multi-dimensional categorical streams.

B. Offline Locally Private Protocols

Following the sparsity-aware hierarchical difference transformation [15] (see Section IV), streaming data analytics reduce to sparse vector aggregation. The work [17] sanitizes one randomly chosen dimension using the full privacy budget ϵ. A later study [20] uses an adjusted transition matrix to sanitize the ternary value $\{-1, 0, 1\}$. Yet, the error dependencies rise to $O(T^2)$ as the user count on each dimension drops from n to n/T. PCKV [18] selects and sanitizes one non-zero entry via locally private mechanisms (e.g., general randomized response [30], unary encoding [3]). Its errors scale at $O(Ts^2)$ for PCKV-UE or $O(T^2)$ for PCKV-GRR. DDRM [19] expands PCKV-UE, selecting k non-zero entries, but the error bound remains $O(Ts^2)$. Since they adopt simple dimension/entry sampling, these sub-optimal methods [17], [20] can be adapted to online settings via streamable uniform sampling.

The Collision mechanism [22] maps non-zero entries to a Bloom-filter, outputting a Bloom-filter index. It attains optimal error dependence $O(Ts)$, but the output probability remains uncertain until all entries are ready. The latest work [23] maps entries to one bucket with pseudo-random weight -1 or $+1$, then clips and adds noise. It has a mean squared error of $O(Ts \log n)$, but is unstreamable. In summary, mechanisms yielding optimal utility (i.e., [22], [23]) need full non-zero entries before sanitization, unfit for streamable implementation. This might imply a trade-off between online responsing and data utility in LDP. Fortunately, our proposal refutes this.

III. Background and Problem Statement

We use $[n]$ to denote $\{1, ..., n\}$ and $[n_1 : n_2]$ to denote $\{n_1, n_1+1, ..., n_2\}$. The symbol $[\![statement]\!]$ represents the Iverson bracket, which takes the value 1 if the $statement$ is true and 0 otherwise.

A. Local Differential Privacy

Let $\mathbf{x}_{j,t} \in \{0,1\}^d$ denote the binary vector of user j at timestamp $t \in [T]$, and let $\mathbf{x}_{j,R}$ represent the data $\{\mathbf{x}_{j,t}\}_{t \in R}$ given a subset of timestamps $R \subseteq [T]$. Our goal is to protect the privacy of the complete streaming data $\mathbf{x}_j = \mathbf{x}_{j,[T]} \in \{0,1\}^{d \times T}$ for each user j. We assume that the total number of changes $\sum_{t=1}^{T} \|\mathbf{x}_{j,t} - \mathbf{x}_{j,t-1}\|_1$ is bounded by s, which is usually much less than $d \cdot T$.

Local differential privacy aims to protect data privacy at the user level, it imposes indistinguishability constraints on the data domain for a single user (see Definition 1).

***Definition** 1 (Local ϵ-DP [9]):* Let $\mathcal{D}_K$ denote the output domain, a randomized mechanism K satisfies local ϵ-differential privacy iff for any data pair $\mathbf{x}, \mathbf{x}' \in \{0,1\}^{d \times T}$, and any outputs $\mathbf{z} \in \mathcal{D}_K$,

$$\mathbb{P}[K(\mathbf{x}) = \mathbf{z}] \leq e^{\epsilon} \cdot \mathbb{P}[K(\mathbf{x}') = \mathbf{z}].$$

Exponential Mechanism. A fundamental tool for designing differentially private mechanisms is the exponential mechanism [31]. This mechanism outputs $\mathbf{z}$ for a given input $\mathbf{x}$ with a probability proportional to $\exp\left(\frac{\mathsf{utility}(\mathbf{x},\mathbf{z}) \cdot \epsilon}{2\Delta}\right)$. Here, utility can be any real-valued function, and sensitivity $\Delta = \max_{z,\mathbf{x},\mathbf{x}'} |\mathsf{utility}(\mathbf{x},\mathbf{z}) - \mathsf{utility}(\mathbf{x}',\mathbf{z})|$. The constant 2 in the denominator accounts for varying normalization factors. If the normalization factor is the same for all inputs (e.g., the `ExSub` mechanism in this work), it can be safely removed.

B. Online Mean and Range Queries

The goal of online user data analytics is to provide real-time statistics of user data. One fundamental statistic is the mean value, where the server estimates the mean value over the population: $\overline{\mathbf{x}}_{*,t} = \sum_{j=1}^{n} \mathbf{x}_{j,t}/n$. The server may also process range queries, such as the total value over a specific period: $\overline{\mathbf{x}}_{*,[t_1:t_2]} = \sum_{j=1}^{n} \sum_{t=t_1}^{t_2} \mathbf{x}_{j,t}/n$.

From the users' perspective, they intermittently respond with some information at several (or all) timestamps. We denote the message user i responds with at timestamp t as $\mathbf{z}_{j,t}$. If they do not respond with any information, we consider it as $\perp$. When enhanced with differential privacy, all messages $\{\mathbf{z}_{j,1}, \ldots, \mathbf{z}_{j,T}\}$ across timestamps $[T]$ are ensured to be locally ϵ-DP. That is, the output $\mathbf{z}$ in Definition 1 represents $\{\mathbf{z}_{j,1}, ..., \mathbf{z}_{j,T}\}$. In our protocol, every message $\mathbf{z}_{j,t}$ belongs to $\{-1, +1, \perp\}^d$.

IV. The Protocol of Online Locally Private Data Analytics

Our protocol contains four components: 1) *ternary residue representation* that sparsifies the local streaming data via a hierarchical tree; 2) *hierarchy level selection* that selects one hierarchy level for each user; 3) *online private ternary aggregation* that yields real-time private ternary statistics; 4) *real-time query answering* that reconstructs mean/frequency/range statistics from the online ternary statistics.

(1) Ternary Representation. Following the best-practices for utilizing sparsity in change [15], we concentrate on temporal residues $\mathbf{R}_{j,t} = \{\mathbf{x}_{j,t} - \mathbf{x}_{j,t-1}\}_{t \in [T]}$ instead of the raw binary vector stream, and assume the initial status $\mathbf{x}_{j,0}$ as $\{0\}^d$. Each residue belongs to the ternary vector domain $\{-1, 0, 1\}^d$.

a) Hierarchical reconstruction: Reversely, the original binary vector can be reconstructed as $\mathbf{x}_{j,t} = \sum_{t'=1}^{t} \mathbf{R}_{j,t'}$. When local privacy is imposed, the residues can be replaced by their estimators, but the reconstruction error will grow linearly with t. Therefore, it is necessary to introduce a hierarchy for recording residues. At the hierarchical level $h \in [0 : \lceil \log_r T \rceil]$, we let the t'-th element $\mathbf{R}_{j,t',h}$ records $\mathbf{x}_{j,r^h \cdot t'} - \mathbf{x}_{j,r^h \cdot (t'-1)}$ with temporal granularity r^h (see Algorithm 1). Consequently, the $\mathbf{x}_{j,t}$ can now be reversely reconstructed from a weighted

Algorithm 1: Residue computation at the level h.

```
Input: Online streaming data $\{\mathbf{x}_{j,t}\}_{t\in[T]}$, hierarchy level
       $h \in [0 : H]$.
Output: The ternary residue at the level $h$.
$t' \leftarrow 0$
$previous \leftarrow \{0\}^d$
for $t \in [T]$ do
    if $t \mod r^h = 0$ then
        $t' \leftarrow t' + 1$
        $\mathbf{R}_{j,t',h} \leftarrow \mathbf{x}_{j,t} - previous$
        $previous \leftarrow \mathbf{x}_{j,t}$
        yield $\mathbf{R}_{j,t',h}$
```

Algorithm 2: Binary vector reconstruction.

```
Input: Online ternary residue estimator $\hat{\mathbf{R}}_{j,t',h}\}$ for
       $h \in [0 : H]$ and $t' \in [T_h]$, a timestamp $t$.
Output: The recovered estimator $\hat{\mathbf{x}}_{j,t}$ at timestamp $t$.
$v \leftarrow \{0\}^d,\ rest \leftarrow t$
for $h \leftarrow H$ to $0$ do
    $t' \leftarrow \lfloor \frac{rest}{r^h} \rfloor + \lfloor \frac{t}{r^{h+1}} \rfloor \cdot r$
    $rest \leftarrow rest \mod r^h$
    if $t' > \lfloor \frac{t}{r^{h+1}} \rfloor \cdot r$ then
        $v \leftarrow v + \sum_{t''=\lfloor \frac{t}{r^{h+1}} \rfloor \cdot r+1}^{t'} \hat{\mathbf{R}}_{j,t'',h}$
return $v$
```

summation of at most $\lceil \log_r T \rceil \cdot (r-1)$ residue estimators, according to the base r notation of t (see Algorithm 2). We let H denote the number of levels $\lceil \log_r T \rceil + 1$, and let T_h denote the total number $\lceil \frac{T}{r^h} \rceil$ of residues at the level h.

On the user side, computing every residue $\mathbf{R}_{j,t',h}$ consumes $O(d)$ time and $O(d)$ memory. On the server side, reconstructing estimator $\widehat{\mathbf{x}}_{j,t}$ takes $O(r \log_r T)$ time.

(2) Hierarchy Level Selection. The hierarchical ternary residues $\mathbf{R}_{j,t',h}$ form a ternary vector with approximate $\frac{dT}{1-1/r}$ dimensions and at most s non-zero entries. Since the MSE grows at least linearly with the number of non-zero entries (see Section II) and every element at level h can be used at most r^h times for reconstruction, every user in our protocol selects only one hierarchy level to response. A side consequence is that the memory/computation/communication costs on the user side is reduced. Let $\mathbf{W}_h$ denote the portion of users who selected the hierarchical level h, we have $\sum_{h=1}^{H} \mathbf{W}_h \equiv 1$. The concrete choice of $\mathbf{W}$ should calibrate the underlying statistical query and settings, and is deferred to the fourth component.

(3) Online Private Ternary Aggregation. After the ternary residue representation and hierarchy level selection, each user now works with streaming ternary vectors $\{\mathbf{R}_{j,t',h}\}_{t'\in[T_h]}$ at level h, which have at most s non-zero entries. The user sanitizes the streaming ternary vectors with an online private mechanism, and sends the output messages when $t \mod r^h = 0$ holds. The server receives these sanitized messages from each user, and derives an (unbiased) estimator $\widehat{\mathbf{R}}_{i,t',h}$. All estimators belonging to the level h are then averaged:

$$\widehat{\mathbf{R}}_{*,t',h} = \Big(\sum_{i=1}^{n} [\![h_i = h]\!] \cdot \widehat{\mathbf{R}}_{i,t',h}\Big) / \#\big\{h_i = h \mid i \in [n]\big\}.$$

(4) Real-time Query Responding. Based on private ternary statistics, the server reconstructs answers for queries. Replacing residues $\widehat{\mathbf{R}}_{i,t',h}$ in Algorithm 2 with their average estimators $\tilde{\mathbf{R}}_{,t',h}$, we obtain an unbiased estimation $\widehat{\mathbf{X}}_{,t}$. Similarly, for a range query $\mathbf{X}_{*,[t_1:t_2]}$, the summation $\sum_{t\in[t_1:t_2]} \widehat{\mathbf{X}}_{*,t}$ yields an unbiased estimator.

Given the query task and the error characterization of the ternary privatization mechanism, we can now specify the hierarchy selection portions $\mathbf{W} \in \Delta_H$. For a fixed $\mathbf{W}$, the number of users selecting the level h is $n \cdot \mathbf{W}_h$. According to Theorem 2 on the proposed ExSub, the mean squared error of each residue estimator $\widehat{\mathbf{R}}_{i,t',h}$ is bounded by $O(\frac{s}{\epsilon^2})$ (see Theorem 2), thus the mean squared error of $\widehat{\mathbf{R}}_{*,t',h} t \in [T_h]$ is bounded by $O(\frac{dTs}{2^h \cdot n \cdot \mathbf{W}_h \cdot \epsilon^2})$ (ignoring the hierarchy sampling error that is negligible when $\epsilon = O(1)$).

b) Mean queries: Since every $\mathbf{R}_{*,t',h}$ contributes 0 or 1 time in one mean query and appears in at most $(r-1)r^h$ queries [32], according to the variance bound on summed variables: $\mathsf{Var}[A+B] \leq 2\mathsf{Var}[A] + 2\mathsf{Var}[B]$, the mean squared error of all mean queries $\{\mathbf{x}_{*,t}\}_{t\in[T]}$ is bounded by (ignored negligible error $O(\frac{s}{n\mathbf{W}_h})$ due to sampling):

$$O\Big(\sum_{h=1}^{H} \frac{dsT(r-1)r^h}{n\mathbf{W}_h \epsilon^2 r^h}\Big) = O\Big(\sum_{h=1}^{H} \frac{dsT(r-1)}{n\mathbf{W}_h \epsilon^2}\Big).$$

By solving the minimization problem given $\mathbf{W} \in \Delta_H$, it is derived that setting $\mathbf{W}_h = 1/H$ approximately minimizes the error and the corresponding error bound is $O(\frac{dsT(r-1)\log_r^2 T}{n \cdot \epsilon^2})$.

c) Range queries: Every $\mathbf{R}_{*,t',h}$ contributes at most $(r-1)r^h$ times in one range query and can appear in at most $T^2/4$ queries, thus the factor on each residue variance is $O\big((r-1)^2 r^{2h} \cdot T^2\big)$. The error of all range queries $\{\mathbf{x}_{*,[t_1:t_2]}\}_{t_1\in[T], t_2\in[t_1:T]}$ is bounded by

$$O\Big(\sum_{h=1}^{H} \frac{dsT \cdot (r-1)^2 r^{2h} \cdot T^2}{n\mathbf{W}_h \epsilon^2 r^h}\Big) = O\Big(\sum_{h=1}^{H} \frac{dsT^3 (r-1)^2 r^h}{n \cdot \mathbf{W}_h \cdot \epsilon^2}\Big).$$

Consequently, setting sampling portions as $\mathbf{W}_h = \frac{r^h}{\sum_{h'=1}^{H} r^{h'}}$ approximately minimizes the error, and the error can be $O(dsT^4(r-1)^2 \log_r T/(n \cdot \epsilon^2))$.

V. The ExSub Mechanism

This section introduces the Exclusive Subset mechanism (ExSub), a locally private mechanism for the streaming ternary vectors $\{\mathbf{R}_{j,t',h}\}_{t'\in[T_h]}$ at level h. Before exploring its online implementation, we first describe its design and properties for the offline version, which presumes complete visibility of the streaming ternary vector.

Data Preparation. We treat the residue vectors $\{\mathbf{R}_{j,t',h}\}_{t'\in[T_h]}$ at the level h as a ternary vector $\mathbf{R}$ of $d \cdot T_h$ dimensions with at most s non-zero entries. To simplify our analysis, we augment this vector with s stub entries, letting d' denote the augmented length $d \cdot T_h + s$. We then interpret the sparse ternary vector as a set. By using symbols i_- and

i_+ to indicate that the i-th element of $\mathbf{R}$ equals -1 and 1 respectively, we express the ternary vector as a subset:

$$\mathbf{S_R} = \{i_-|i \in [d'] \, \text{and} \, \mathbf{R}_i = -1\} \bigcup \{i_+|i \in [d'] \, \text{and} \, \mathbf{R}_i = 1\}.$$

If the vector contains fewer than s non-zero entries, we add stub symbols to $\mathbf{S_R}$ to ensure exactly s elements:

$$\left\{i_+ \mid i \in \left[dT_h : \; dT_h + s - \|\{\mathbf{R}_{j,t',h}\}_{t' \in [T_h]}\|_1\right]\right\}.$$

A. Mechanism Design

Current streamable mechanisms rely on uniform random selection on dimensions [17], [20] or on non-zero entries [15], [18], [19], [21], and perform independent sanitation on the selected partial data. This design simplifies the mechanism but leads to an extra $O(d'/s)$ or $O(s)$ error factor compared to optimal mechanisms [22], [23]. Contrarily, in optimal mechanisms, each non-zero or zero entry is not treated equally (after specifying the local hash), and the privatized signal for each entry cannot be determined until all entries are ready. Our `ExSub` mechanism, however, is permutation invariant for both input and output entries. Thus, each entry can establish the signaling parameter prior to the remaining entries' readiness.

Let $\mathcal{Z}$ represent the symbol domain $\{1_-, 1_+, ..., d'-, d'+\}$. The input $\mathbf{S_R}$ can be seemed as a subset of $\mathcal{Z}$ with a fixed cardinality of s. The `ExSub` mechanism will operate on the domain of $\mathcal{Z}$ and probabilistically output a subset $\mathbf{Z} \subseteq \mathcal{Z}$ with a fixed size of m. Since $i-$ and i_+ cannot simultaneously appear in the input, they should also be mutually exclusive in the output. We define the output domain as:

$$\mathcal{Z}^{m\pm} = \left\{\mathbf{Z} \in \mathcal{Z}^m \, \text{and} \, |\{i_-, i_+\} \cup \mathbf{Z}| \leq 1 \, \forall i \in [d']\right\}.$$

Inspired by the extremal property [33], [34] of LDP, if the output $\mathbf{Z}$ is close to the input $\mathbf{S_R}$, its output probability is proportional to 1. Otherwise, the output probability is proportional to $\exp(-\epsilon)$. The closeness criterion between $\mathbf{Z}$ and $\mathbf{S_R}$ is their commonality of elements. We therefore define a binary utility function within the framework of exponential mechanism as follows:

$$\mathsf{utility}(\mathbf{S_R}, \mathbf{Z}) = \begin{cases} 0, & \text{if } \mathbf{Z} \cap \mathbf{S_R} \neq \Phi; \\ -1. & \quad \textit{otherwise.} \end{cases} \tag{1}$$

We describe the design of the (d', s, ϵ, m)-`ExSub` mechanism in Definition 2. The specific choice of output cardinality m will be discussed in Section V-D.

***Definition** 2 ((d', s, ϵ, m)-ExSub mechanism):* Taking an s-sparse d'-dimenisonal ternary vector $\mathbf{R} \in \{-1, 0, 1\}^{d'}$ as input, the `ExSub` mechanism randomly outputs $\mathbf{Z} \in \mathcal{Z}^{m\pm}$ according to the following probability design:

$$\mathbb{P}[\mathbf{Z}|\mathbf{R}] = \exp(\mathsf{utility}(\mathbf{S_R}, \mathbf{Z}) \cdot \epsilon)/\Omega,$$

where $\Omega = 2^m \binom{d'}{m} + (e^{-\epsilon} - 1) \sum_{m'=0}^{m} 2^{m-m'} \binom{s}{m'} \binom{d'-s}{m-m'}$ is the normalization factor.

Algorithm 3: The offline (d', s, ϵ, m)-`ExSub`.

```
Input: A ternary vector $\mathbf{R}$ in set form $\mathbf{S_R} \in \mathcal{Z}^s$.
Output: A private view $\mathbf{Z} \in \mathcal{Z}^{m\pm}$ that satisfies $\epsilon$-LDP.
// Data-independent phase
1  $\Omega = 2^m \binom{d'}{m} + (e^{-\epsilon} - 1) \sum_{m'=0}^{m} 2^{m-m'} \binom{s}{m'} \binom{d-s}{m-m'}$
2  $r \leftarrow \mathsf{Uniform}(0.0, 1.0), \;\; acc \leftarrow 0.0$
3  for $num_t \leftarrow 0$ to $m$ do
4      for $num_r \leftarrow 0$ to $m - num_t$ do
5          $l \leftarrow \binom{s}{num_t} \binom{s-num_t}{num_r} \binom{d'-s}{m-num_t-num_r} \cdot 2^{m-num_t-num_r}$
6          $acc \leftarrow acc + \frac{1}{\Omega \cdot e^{\epsilon \cdot [\![num_t=0]\!]}} \cdot l$
7          if $acc \geq r$ then break
8      if $acc \geq r$ then break
// Data-dependent phase
9  nonzeros $\leftarrow \mathsf{UniformSample}(\mathbf{S_R}, \; num_t + num_r)$
10 trues $\leftarrow \mathsf{UniformSample}(\text{nonzeros}, \; num_t)$
11 reverses $\leftarrow \{i_{-b} \mid i_b \in \text{nonzeros} \backslash \text{trues}\}$
12 allfalses $\leftarrow \{i_b \mid j \in [d'], b \in \{-1, 1\}, i_b \notin \mathbf{S_R} \; \textit{and} \; i_{-b} \notin \mathbf{S_R}\}$
13 falses $\leftarrow \mathsf{UniformSample}(\text{allfalses}, \; m - num_t - num_r)$
14 $\mathbf{Z} \leftarrow$ trues $\cup$ reverses $\cup$ falses
15 return $\mathbf{Z}$
```

An Example. Let us illustrate the `ExSub` mechanism through an example with $s = 1$, $\epsilon = \log 2$, and $m = 2$. Consider a user with local residue $\mathbf{R} = [0, -1] \in \{-1, 0, 1\}^2$. Here, the augmented domain size is $d' = 3$ and $\mathbf{R}$'s set representation is $\mathbf{S_R} = \{2_-\}$. In the $(3, 1, \log 2, 2)$-`ExSub` mechanism, the mechanism would select any of the following outputs with a probability of 1/8 (i.e., when 2_- presents):

$$\{2_-, 1_-\}, \; \{2_-, 1_+\}, \; \{2_-, 3_-\}, \; \{2_-, 3_+\},$$

and it would select each of the remaining outputs with a probability of 1/16:

$$\{2_+, 1_-\}, \{2_+, 1_+\}, \{2_+, 3_-\}, \{2_+, 3_+\},$$
$$\{3_-, 1_-\}, \{3_+, 1_+\}, \{3_-, 1_+\}, \{3_+, 1_+\}.$$

The exclusive output domain of `ExSub`, differing from PrivSet [34] used for item frequency estimation, leads to optimal mean value estimation (refer to Figure 1 and Theorem 2).

B. Offline Implementation

The output universe $\mathcal{Z}^{m\pm}$ grows exponentially with d' and m, making a direct scan for output selection infeasible. Here, we propose an efficient approach in Algorithm 3 based on uniform sampling without replacement. We split the output universe into $\frac{(m+2)(m+1)}{2}$ distinct subgroups, each identified by two parameters: $num_t \in [0, m]$ and $num_r \in [0, m - num_t]$. The num_t denotes the intersection size of $\mathbf{S_R}$ and $\mathbf{Z}$, while num_r indicates the intersection size of $\{i_{-b} \mid i_b \in \mathbf{S_R}\}$ and $\mathbf{Z}$. Elements within the same subgroup possess the same output probability $\frac{e^{-[\![num_t=0]\!] \cdot \epsilon}}{\Omega}$.

Algorithm 3 starts by choosing a subgroup (i.e., num_t, num_r) in lines 1-8 based on the subgroup's size l and the output probability $\frac{e^{-[\![num_t=0]\!] \cdot \epsilon}}{\Omega}$ within that subgroup, where $\mathsf{Uniform}(0.0, 1.0)$ gives a uniform random sample from $[0.0, 1.0]$. The overall subgroup selection probability is $l \cdot \frac{e^{-[\![num_t=0]\!] \cdot \epsilon}}{\Omega}$. Secondly, in line 9, we uniformly sample $num_t + num_r$ symbols from $\mathbf{S_R}$ to form the set *nonzeros*, given the total size $num_t + num_r$ of the intersection between $\mathbf{Z}$ and $\mathbf{S_R} \bigcup \{i_{-b} \mid i_b \in \mathbf{S_R}\}$. Next, we uniformly sample num_t symbols from $nonzeros$ to create the intersection $trues = \mathbf{Z} \bigcap \mathbf{S_R}$ (line 10), and num_r symbols from the reversed elements of $nonzeros$ $trues$ to form $reverses = \mathbf{Z} \bigcap \{i_{-b} \mid i_b \in \mathbf{S_R}\}$ (line 11). Lastly, we sample $m - num_t - num_r$ symbols from $allfalses = \mathcal{Z} \backslash (\mathbf{S_R} \bigcup \{i_{-b} \mid i_b \in \mathbf{S_R}\})$ to form $falses = \mathbf{Z} \bigcap allfalses$, and generate the final private view $\mathbf{Z}$. In essence, the last three steps uniformly select an output $\mathbf{Z}$ from the subgroup (num_t, num_r). Uniform sampling within each subgroup is assured by the definition of num_t and num_r and the uniform randomness of each sampling subroutine.

C. Online Implementation

Next, we discuss an online implementation of the ExSub, similar to Algorithm 3, but with $\mathbf{x}_{i,t}$ serving as the streaming input and each output of $\perp$ or i_b being delivered in real time.

A crucial aspect of ExSub is that the selection of the subgroup (num_t, num_r) is data-independent, and all data-dependent operations are based on uniform sampling with known cardinalities. Hence, before timestamp 1, we perform a pre-computation of the data-independent part: selecting a subgroup (num_t, num_r). From timestamp 1 onwards, as streaming data arrives, we execute an online uniform sampling of $num_t + num_r$ from s non-zero entries. Within this sampling, num_t elements are immediately chosen uniformly from $num_t + num_r$ symbols. Simultaneously, for $d' - s$ zero entries, num_f symbols are uniformly sampled. Notably, once (num_t, num_r) is chosen, the parameters (i.e., the input/output cardinalities) for the three uniform samplings are fixed, enabling the possibility of streamable sampling. Real-time uniform-random sampling of a combination from a fixed population is referred to as *sequential random sampling* in the literature. We utilize *selection sampling* or *Algorithm S* [35], [36] for these sub-procedures, which selects each element with a probability equal to the number of remaining elements to be sampled over the total number of remaining elements. More efficient alternative approaches are described in [26].

The complete online implementation is presented in Algorithm 4, which executes several (recursive) sequential random samplings. The probability of each final output is identical to that of the offline version in Algorithm 3, thus ensuring the same utility guarantees. Additionally, because each symbol i_b in Algorithm 4 is submitted at its corresponding timestamp j, no extra information is leaked, thereby providing the same privacy guarantee as Algorithm 3. The overall computational costs for each user are $O(d \cdot \frac{T}{r^h})$, memory costs are $O(d+5) = O(d)$, and communication costs are $O(m \cdot \log d)$.

D. Estimating Value and Frequency

In this section, we exploit the probability transition behavior of the ExSub mechanism and provide unbiased estimators. When considering a symbol $i_b \in \mathcal{Z}$, there are three potential cases related to the input $\mathbf{S_R}$: I) $i_b \in \mathbf{S_R}$; II) $i_{-b} \in \mathbf{S_R}$; III) $i_b \notin \mathbf{S_R}$ *and* $i_{-b} \notin \mathbf{S_R}$. Based on the design of the ExSub mechanism, the probability of i_b appearing in the output $\mathbf{Z}$ varies depending on the case. For any $j \in [1 : d']$ and $b \in \{+, -\}$, we define the conditional probabilities as the true, false, and reverse positive rates, respectively. The specific definitions are as follows.

Algorithm 4: The online (d', s, ϵ, m)-ExSub.

```
Input: Streaming ternary vector $\mathbf{Y}_{j,t',h}$, the vector's overall
       length $d' = d \cdot T_h + s$.
Output: Streaming output $\mathbf{Z}_{t'} \in \mathcal{Z}^{m\pm}$ with $\epsilon$-LDP.
Get $num_t, num_r$ as in line 1-10 of Algorithm 3
$num_{rp} \leftarrow s, \quad num_{rf} \leftarrow d'$
$num_f \leftarrow m - num_t - num_r$
for $t \leftarrow 1$ to $T$ do
    if $t \mod r^h = 0$ then
        $t' \leftarrow \frac{t}{r^h}, \quad \mathbf{Z}_{t'} \leftarrow \Phi$
        Get residue $\mathbf{Y}_{j,t',h}$ according to Algorithm 1
        for $j \leftarrow 1$ to $d$ do
            Let $b$ denote the $i$-th value of $\mathbf{Y}_{j,t',h}$
            if $b = 0$ then
                $num_{rf} \leftarrow num_{rf} - 1$
                if Uniform$(0.0, 1.0) < \frac{num_f}{num_{rf}}$ then
                    if Uniform$(0.0, 1.0) < 0.5$ then
                    $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_+\}$
                    else $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_-\}$
                    $num_f \leftarrow num_f - 1$
            else
                $num_{rp} \leftarrow num_{rp} - 1$
                if Uniform$(0.0, 1.0) < \frac{num_t + num_r}{num_{rp}}$ then
                    if Uniform$(0.0, 1.0) < \frac{num_t}{num_t + num_r}$
                     then
                        $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_b\}$
                        $num_t \leftarrow num_t - 1$
                    else
                        $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_{-b}\}$
                        $num_r \leftarrow num_r - 1$
        $\mathbf{Z}_{t'} = \{(i + t' \cdot d)_{b'} \mid i_{b'} \in \mathbf{Z}_{t'}\}$
        yield $\mathbf{Z}_{t'}$
```

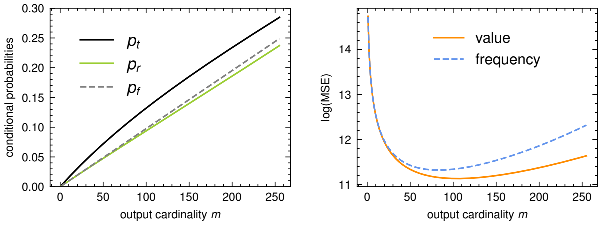


Fig. 1. The true/reverse/false positive rates and mean squared error on value/frequency of $(d'\!=\!512, s\!=\!4, \epsilon\!=\!0.5, m)$-ExSub with $n = 1$.

$$
\begin{aligned}
p_t &= \mathbb{P}[i_b \in \mathbf{Z} \ \mid \ i_b \in \mathbf{S_R}] \ = 2^{m-1}\tbinom{d'-1}{m-1}/\Omega; \\
p_f &= \mathbb{P}[i_b \in \mathbf{Z} \ \mid \ i_b \notin \mathbf{S_R} \ \mathit{and} \ i_{-b} \notin \mathbf{S_R}] \\
&= \frac{2^{m-1}\tbinom{d'-1}{m-1} - (1-e^{-\epsilon})\sum_{m'=0}^{m-1} 2^{m'}\tbinom{s}{m-1-m'}\tbinom{d'-s-1}{m'}}{\Omega}; \\
p_r &= \mathbb{P}[i_b \in \mathbf{Z} \ \mid \ i_{-b} \in \mathbf{S_R}] \\
&= \frac{2^{m-1}\tbinom{d'-1}{m-1} - (1-e^{-\epsilon})\sum_{m'=0}^{m-1} 2^{m'}\tbinom{s-1}{m-1-m'}\tbinom{d'-s}{m'}}{\Omega}.
\end{aligned}
$$

The discrepancies in transition probability allow the server to partially discern between the three cases regarding the input, enabling the derivation of unbiased estimators for value and frequency as shown in Proposition 1.

***Proposition** 1 (Unbiased estimators):* Given the private view $\mathbf{Z}$ about ternary vector $\mathbf{R}$ from ExSub, an unbiased estimator of ternary value $\overline{\mathbf{R}}_j = [\![i_+ \in \mathbf{S_R}]\!] - [\![i_- \in \mathbf{S_R}]\!]$ for $j \in [1, d']$ is $\widehat{\overline{\mathbf{R}}}_j = \frac{[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]}{p_t - p_r}$; an unbiased estimator of frequency $\underline{\mathbf{R}}_j = [\![i_+ \in \mathbf{S_R}]\!] + [\![i_- \in \mathbf{S_R}]\!]$ is $\widehat{\underline{\mathbf{R}}}_j = \frac{[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!] - 2p_f}{p_t + p_r - 2p_f}$.

Next, we analyze the utility guarantee. Based on transition probabilities, we derive the error formulation of the ExSub with a fixed m, we then select an appropriate value for m, which is $\Theta(d'/s)$, as outlined in Theorem 2. Consequently, the mean squared error is roughly minimized to $O(\frac{d's}{\epsilon^2})$. As the choice of m has significant impact on performance (as illustrated in Figure 1), we empirically set m to $\lceil \frac{d'}{e^\epsilon s + s + 2} \rceil$.

***Theorem** 2 (Mean Squared Error Bounds):* When $\epsilon = O(1)$, takes as an input $\mathbf{R}$, the (d', s, ϵ, m)-ExSub mechanism with $m = \lceil d'/(e^\epsilon s + s + 2) \rceil$ satisfies $\mathbb{E}[\sum_{i=1}^{d'} \|\widehat{\overline{\mathbf{R}}}_i - \overline{\mathbf{R}}_i\|_2^2] \leq O(\frac{d's}{\epsilon^2})$; the (d', s, ϵ, m)-ExSub with $m = \lceil d'/(e^\epsilon s + 2s + 1) \rceil$ satisfies $\mathbb{E}[\sum_{j=1}^{d'} \|\widehat{\underline{\mathbf{R}}}_j - \underline{\mathbf{R}}_j\|_2^2] \leq O(\frac{d's}{\epsilon^2})$.

Proof: According to the transition probabilities, we have $\mathbb{E}[\sum_{i=1}^{d'} \|\widehat{\overline{\mathbf{R}}}_i - \overline{\mathbf{R}}_i\|_2^2] = \frac{s((p_t + p_r) - (p_t - p_r)^2) + (d' - s)(2p_f)}{(p_t - p_r)^2}$ and $\mathbb{E}[\sum_{j=1}^{d'} \|\widehat{\underline{\mathbf{R}}}_j - \underline{\mathbf{R}}_j\|_2^2] = \frac{s(p_t + p_r)(1 - p_t - p_r) + (d' - s)2p_f(1 - 2p_f)}{(p_t + p_r - 2p_f)^2}$. Given $\epsilon = O(1)$, observe that when $m = \Theta(d/s)$, both $\frac{p_f}{(p_t - p_r)^2} = O(s/\epsilon^2)$ and $\frac{p_f(1 - 2p_f)}{(p_t + p_r - 2p_f)^2} = O(s/\epsilon^2)$ hold, thus we have the conclusion. ■

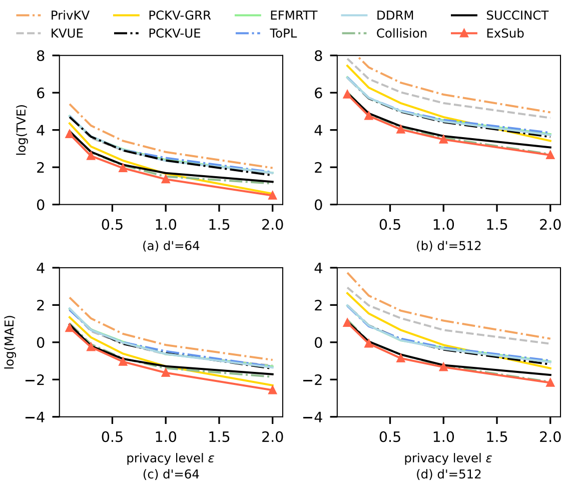
It should be noted that both the data preparation and the element-wise estimation of value/frequency can be implemented in a streaming fashion, meaning that the entire ExSub mechanism can be executed online.

VI. Experimental Evaluation

We assess the utility performance of locally private protocols, beginning with offline scenarios for comprehensive comparison of the ExSub with existing methods, and subsequently considering online mean and range queries. For offline assessments, we compare against PrivKV [17], KVUE [20], PCKV-GRR, PCKV-UE [18], Collision [22], and SUCCINCT [23]. For online comparisons, we compare against the Erlingsson *et al.*'s hierarchical-tree-based method [15] (EFMRTT), ToPL [21], DDRM [19], PCKV-GRR, and PCKV-UE. Several online protocols [3], [4], [28] with weaker data privacy protection are omitted from comparison. We note that though FutureRand [29] is theoretically optimal (ignoring logarithm factors), its empirical errors overwhelm other methods (See Table II). Our code is available at https://github.com/wangsw/OnlineLDP.

A. Datasets & Metrics

We use two real-world datasets, Stock [1] and Trajectory [2], and several additional synthetic datasets. The Stock dataset includes daily prices for 7136 U.S. stocks from 2014 to 2017. We segment it into 16,0000 records, each with 32 consecutive



(a) d'=64

(b) d'=512

(c) d'=64

(d) d'=512

Fig. 2. Error results without post-processing with $n = 10000$, $m = 8$ and dimension parameter d' varies from 64 to 512.

trading days of closing prices, processed into ternary values of $\{-1, 0, 1\}$ based on whether the accumulated change since the last record exceeds 2%. The Trajectory dataset contains 442 taxi trajectories in Porto, Portugal. We partition a specific rectangular area into 3×4 cells, and segment the data into 1,044,693 trajectories, each consisting of 32 celled locations, following [19]. For the synthetic datasets, each user's ternary residue vector is independently generated, with s non-zero entries randomly selected from $d \cdot T$ entries.

We assess utility using total variation error (TVE) and maximum absolute error (MAE). For mean queries, TVE is given by TVE $= \sum_{t \in [T]} \|\widehat{\overline{\mathbf{x}}}_{*,t} - \overline{\mathbf{x}}_{*,t}\|_1$, where $\overline{\mathbf{x}}*, t$ is the true mean value at timestamp t, defined as $\sum_{j=1}^{n} \mathbf{x}_{j,t}/n$; MAE is defined as MAE $= \max_{t \in [T]} \|\widehat{\overline{\mathbf{x}}}_{*,t} - \overline{\mathbf{x}}_{*,t}\|_{+\infty}$. These metrics are also applicable to range queries and intermediate residue statistics. Each reported result is the average of 100 independent simulations. For offline queries, we process the intermediate residue estimators $\hat{\overline{\mathbf{R}}}$ and $\hat{\underline{\mathbf{R}}}$ from all protocols by projecting them onto a capped $\Delta_{d'}$ simplex [37]. An exception is the SUCCINCT protocol [23], which only estimates the mean residue value $\hat{\overline{\mathbf{R}}}_*$; here, we truncate each mean residue value to $[-1, 1]$. Experimental results are presented both with and without this post-processing in standard scenarios.

B. Offline Queries

We begin with offline experiments under extensive synthesized settings to study the impact of various parameters.

Effects of dimensions $d \cdot T$**.** We conduct simulations with $n = 10000$, $s = 8$, and overall dimension $d' = d \cdot T + s$ ranging from 64 to 512. Figure 2 presents the error results for the mean residue value. The ExSub consistently outperforms existing approaches in every setting. When the domain size is relatively small ($d' = 64$), the performance of PCKV-GRR is close to that of the optimal mechanisms (i.e., Collision/SUCCINT/ExSub), but the gap increases as the domain size grows. A similar trend is observed for PrivKV/KVUE, corroborating our theoretical analyses sug-

[1] https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs

[2] https://www.kaggle.com/datasets/crailtap/taxi-trajectory

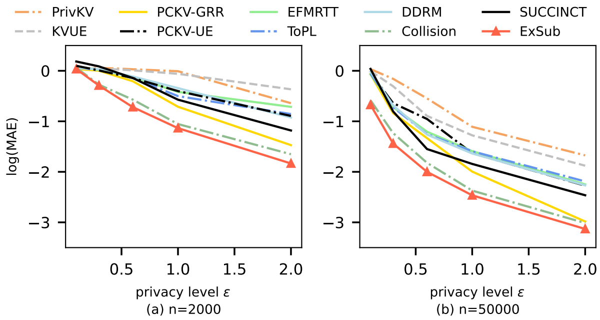


Fig. 3. MAE results with post-processing with $d' = 128$, $m = 8$ and number of users n varies from 2000 to 50000.

TABLE II
ERROR RESULTS WITHOUT POST-PROCESSING UNDER EXTREMELY LOW/HIGH PRIVACY BUDGETS ($n = 10000$, $d' = 128$, $s = 8$).

	TVE results				
	$\epsilon = 0.001$	$\epsilon = 0.01$	$\epsilon = 1.0$	$\epsilon = 3.0$	$\epsilon = 5.0$
FutureRand [29]	3.4e+4	3493.8	32.3	10.1	6.78
Collision [22]	4.6e+3	463.7	4.13	1.15	0.54
SUCCINCT [23]	4.7e+3	479.6	5.10	2.66	2.48
`ExSub`	**4.0e+3**	**393.9**	**3.64**	**0.84**	**0.33**
	MAE results				
	$\epsilon = 0.001$	$\epsilon = 0.01$	$\epsilon = 1.0$	$\epsilon = 3.0$	$\epsilon = 5.0$
FutureRand [29]	1025.8	97.0	0.90	0.31	0.24
Collision [22]	128.1	12.6	0.11	0.034	0.021
SUCCINCT [23]	131.7	13.7	0.14	0.077	0.077
`ExSub`	**113.7**	**10.5**	**0.094**	**0.026**	**0.014**

gesting a sub-optimal dependence on domain size for PCKV-GRR/PrivKV/KVUE.

Effects of users n. Simulations conducted with $d' = 128$ and $s = 8$ vary the number of users from 2000 to 50000. Figure 3 presents the results for the mean residue value, with `ExSub` consistently outperforming existing approaches. As the user population grows (i.e., the effect of post-processing decreases), the performance gap becomes more significant. While the SUCCINT protocol demonstrates excellent performance without post-processing by truncating extreme values, its performance after post-processing is unsatisfactory.

Effects of sparsity s. With $n = 10000$ and $d' = 256$, we vary the sparsity parameter s from 1 to 64. Figure 4 presents the results for the mean residue. As s increases, the performance gap between PCKV-GRR/PrivKV/KVUE and `ExSub` decreases, while the gap between PCKV-UE/EFMRTT/ToPL/DDRM and `ExSub` increases. This confirms our theoretical analyses suggesting PCKV-UE/EFMRTT/ToPL/DDRM suffer sub-optimal dependence on sparsity.

Results under extreme privacy budgets. We conduct simulations with $n = 10000$, $d' = 128$, and $s = 8$ and present the error results of optimal mechanisms under extremely small or large privacy budgets in Table II. These results are insightful for evaluating performance in asymptotic settings and in the emerging shuffle model [38]–[40] of differential privacy. `ExSub` consistently exhibits a minimum constant factor on error bounds in every setting and surpasses existing optimal mechanisms by 10%-30%.

Effects of fan out parameter r. With $n = 10000$, $d = 1$, $T = 256$ and $s = 8$, we vary the fan-out parameter r in the hierarchical tree from 2 to 16. Table III presents the results of `ExSub` and DDRM mechanism for mean queries. The fan-out parameter $r = 2$ shows slightly better performance than other values, thus we use $r = 2$ as default.

TABLE III
MAE RESULTS ON MEAN QUERIES WITHOUT POST-PROCESSING UNDER VARY FAN-OUT PARAMETER ($n = 1000000$, $d = 1$, $T = 128$, $s = 8$).

	$\epsilon = \mathbf{0.1}$			
	$r = 2$	$\mathbf{r = 4}$	$r = 8$	$r = 16$
DDRM [19]	**2.94**	3.02	3.14	3.01
`ExSub`	**1.13**	1.19	1.22	1.31
	$\epsilon = \mathbf{1.0}$			
	$r = 2$	$r = 4$	$r = 8$	$r = 16$
DDRM [19]	**0.277**	0.284	0.281	0.314
`ExSub`	**0.0982**	0.0983	0.103	0.107

C. Online Mean Queries

On Stock dataset. We limit the number of significant changes in stock prices to 6 (i.e., $s = 6$). As the hierarchical residue tree is not applicable to the integer-valued domain in this case, we directly employ ternary privatization mechanisms on the $\{-1, 0, 1\}$ rise/drop record with 32 timestamps. Figure 5 showcases the results, with `ExSub` outperforming existing methods by about 30%.

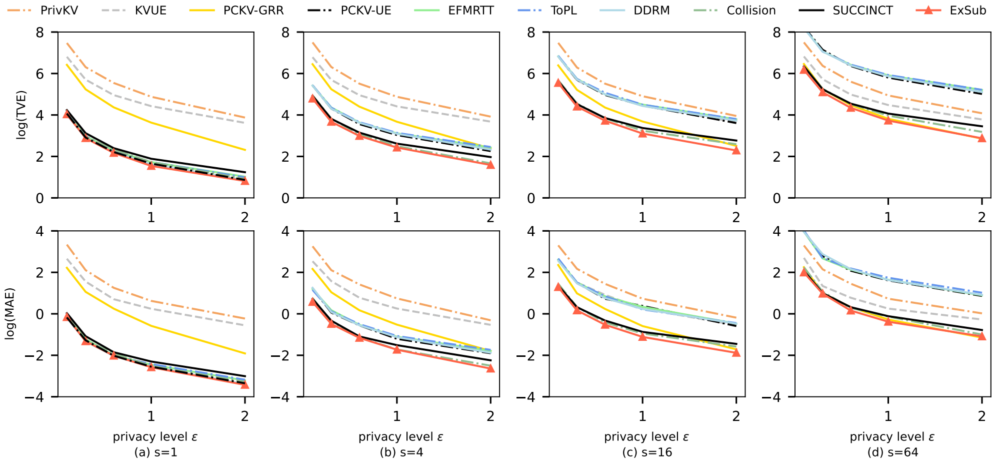
On Trajectory dataset. We limit the number of changes in celled locations to 4 (i.e., $s = 2 \cdot 4$), and utilize a fan-out parameter $r = 2$ for continual location distribution estimation. Figure 6 presents the error results, where `ExSub` shows a reduction of about 50% compared to other methods.

D. Online Range Queries

Here, we assess the performance for range queries and highlight the need for calibrated portion strategies from Section IV. As the hierarchical structure is not suitable for the Stock dataset, we focus on range queries over the Trajectory dataset, such as compiling location heatmaps over a day.

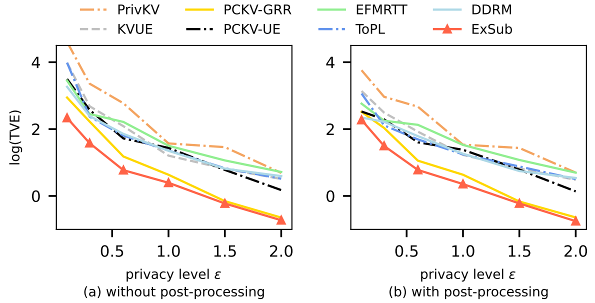
We present the results using uniform portions $\mathbf{W}_h = 1/H$ in Figure 7, where the `ExSub` mechanism surpasses other methods by approximately 50%. We also experiment with calibrated portions, calculated by a more nuanced analysis of the weight of each level h, to mitigate overestimation of weights for levels with large h. Specifically, for T prefix range queries, the multiplicative factor on the variance of residues $\mathbf{R}_{*,t',h}$ is $(\sum_{t'=0}^{(r-1)r^h} t'^2)+\sum_{t'=(r-1)r^h}^{T-r^h}(r-1)^2 r^{2h} = (T - r^{h+1} + \frac{(r-1)r^h((r-1)r^h+1)(2(r-1)r^h+1)}{6(r-1)^2 r^{2h}})r^{2h}$. Therefore, setting $\mathbf{W}_h = (T - r^{h+1} + \frac{(r-1)r^h((r-1)r^h+1)(2(r-1)r^h+1)}{6(r-1)^2 r^{2h}})r^h$ approximately minimizes the error.

The results with this portion strategy are marked as `ExSub`-*calibrated* in Figure 8 for TRAJECTORY dataset and in Figure 9 for a synthetic dataset with relatively long sequences (i.e., $T = 128$). Compared to the uniform portion strategy (denoted as `ExSub`-$uniform$), the calibrated strategy reduces the error by about 20% for prefix/all range queries. This demonstrates that the portions should be calibrated to the specific task.



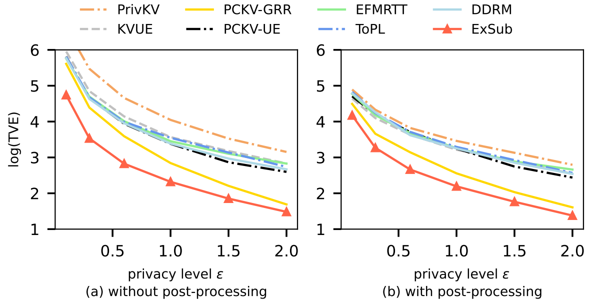
(a) s=1
(b) s=4
(c) s=16
(d) s=64

Fig. 4. Error results without post-processing with $n = 10000$, $d' = 256$ and sparsity parameter s varies from 1 to 64.



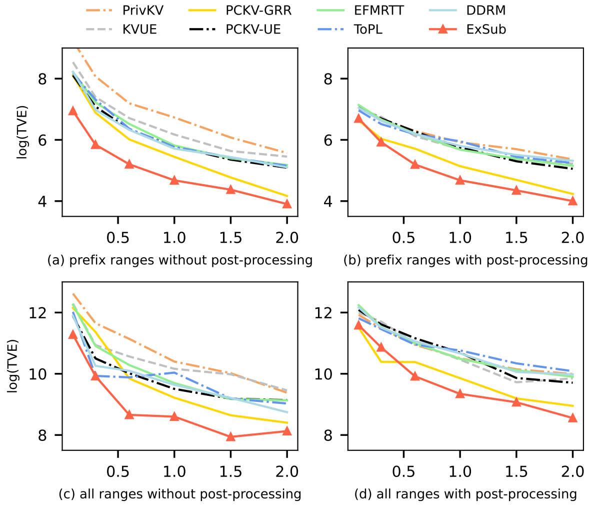
(a) without post-processing
(b) with post-processing

Fig. 5. TVE results for mean queries on the Stock dataset.



(a) without post-processing
(b) with post-processing

Fig. 6. TVE results for mean queries on Trajectory dataset.



(a) prefix ranges without post-processing
(b) prefix ranges with post-processing
(c) all ranges without post-processing
(d) all ranges with post-processing

Fig. 7. TVE results on range queries of Trajectory dataset.

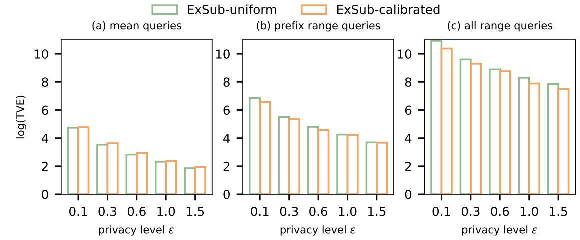


Fig. 8. TVE results comparison on Trajectory dataset.

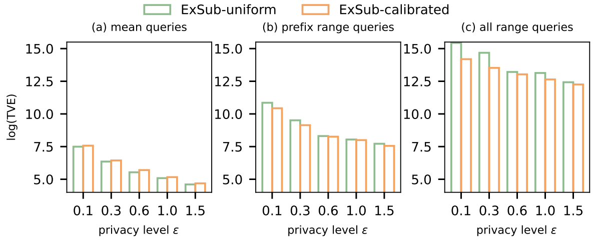


Fig. 9. TVE results comparison on a synthetic dataset with $n = 100000$, $d = 8$, $t = 128$, and $m = 20$.

E. Summary

In summary, ExSub mechanism demonstrates superior performances in both offline and online settings. It reduces about 10%-30% error when compared to existing optimal offline mechanisms (i.e., Collision and SUCCINCT), and reduces 40%-60% error when compared to existing online protocols.

VII. Conclusion

We introduced a general protocol for online locally private data analytics, encompassing an optimal and streamable mechanism for the underlying ternary vector aggregation problem. The protocol is capable of handling a wide range of user data, such as multidimensional binary/set-valued vectors, and supporting a variety of statistical queries, like mean/frequency estimation and range queries. At the heart of our mechanism is the exploitation of the mutually exclusive relationship to match the error lower bounds. Furthermore, our protocol only uses primitives that allow for online computation and response, such as uniform sampling with a fixed population size. Our extensive experiments demonstrate that our proposal outperforms existing protocols by around 50%.

References

[1] Y. Wang, L. Kung, and T. A. Byrd, "Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations," *Technological forecasting and social change*, vol. 126, pp. 3–13, 2018.

[2] C.-Y. Chow and M. F. Mokbel, "Trajectory privacy in location-based services and data publication," *ACM Sigkdd Explorations Newsletter*, vol. 13, no. 1, pp. 19–29, 2011.

[3] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," *CCS*, 2014.

[4] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR)*. Springer, 2017, vol. 18.

[6] E. Goldman, "An introduction to the california consumer privacy act (ccpa)," *Santa Clara Univ. Legal Studies Research Paper*, 2020.

[7] R. Creemers and G. Webster, "Translation: Personal information protection law of the people's republic of china—effective nov. 1, 2021," *DigiChina Project, August*, vol. 20, 2021.

[8] C. Dwork, "Differential privacy: A survey of results," *International Conference on Theory and Applications of Models of Computation*, pp. 1–19, 2008.

[9] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[10] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudinger, V. V. Prakash, A. Legendre, and S. Duplinsky, "Emoji frequency detection and deep link frequency," Jul. 11 2017, uS Patent 9,705,908.

[11] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudiger, V. R. Sridhar, and D. Davidson, "Learning new words," Mar. 14 2017, uS Patent 9,594,741.

[12] X. Xiong, S. Liu, D. Li, Z. Cai, and X. Niu, "A comprehensive survey on local differential privacy," *Security and Communication Networks*, vol. 2020, 2020.

[13] M. Yang, L. Lyu, J. Zhao, T. Zhu, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," *arXiv preprint arXiv:2008.03686*, 2020.

[14] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1298–1309.

[15] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," *SODA*, 2019.

[16] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," *CCS*, 2016.

[17] Q. Ye, H. Hu, X. Meng, and H. Zheng, "Privkv: Key-value data collection with local differential privacy," *IEEE S&P*, 2019.

[18] X. Gu, M. Li, Y. Cheng, L. Xiong, and Y. Cao, "PCKV: Locally differentially private correlated key-value data collection with optimized utility," *USENIX Security*, 2020.

[19] Q. Xue, Q. Ye, H. Hu, Y. Zhu, and J. Wang, "Ddrm: A continual frequency estimation mechanism with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[20] L. Sun, J. Zhao, X. Ye, S. Feng, T. Wang, and T. Bai, "Conditional analysis for key-value data with local differential privacy," *arXiv preprint arXiv:1907.05014*, 2019.

[21] T. Wang, J. Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, and S. Jha, "Continuous release of data streams under both centralized and local differential privacy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1237–1253.

[22] S. Wang, J. Li, Y. Qian, J. Du, W. Lin, and W. Yang, "Hiding numerical vectors in local private and shuffled messages." in *IJCAI*, 2021, pp. 3706–3712.

[23] M. Zhou, T. Wang, T. H. Chan, G. Fanti, and E. Shi, "Locally differentially private sparse vector aggregation," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2022, pp. 1565–1565.

[24] G. Cormode, T. Kulkarni, and D. Srivastava, "Answering range queries under local differential privacy," *VLDB*, 2019.

[25] T. Kulkarni, "Answering range queries under local differential privacy," *SIGMOD*, 2019.

[26] M. Shekelyan and G. Cormode, "Sequential random sampling revisited: Hidden shuffle method," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3628–3636.

[27] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," *FOCS*, 2013.

[28] M. Joseph, A. Roth, J. Ullman, and B. Waggoner, "Local differential privacy for evolving data," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[29] O. Ohrimenko, A. Wirth, and H. Wu, "Randomize the future: Asymptotically optimal locally private frequency estimation protocol for longitudinal data," in *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2022, pp. 237–249.

[30] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[31] F. McSherry and K. Talwar, "Mechanism design via differential privacy." *FOCS*, 2007.

[32] G. Cormode, T. Kulkarni, and D. Srivastava, "Answering range queries under local differential privacy," *Proceedings of the VLDB Endowment*, vol. 12, no. 10, pp. 1126–1138, 2019.

[33] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *Advances in neural information processing systems*, vol. 27, 2014.

[34] S. Wang, L. Huang, Y. Nie, P. Wang, H. Xu, and W. Yang, "Privset: Set-valued data analyses with locale differential privacy," *INFOCOM*, 2018.

[35] C. Fan, M. E. Muller, and I. Rezucha, "Development of sampling plans by using sequential (item by item) selection techniques and digital computers," *Journal of the American Statistical Association*, vol. 57, no. 298, pp. 387–402, 1962.

[36] T. G. Jones, "A note on sampling a tape-file," *Communications of the ACM*, vol. 5, no. 6, p. 343, 1962.

[37] W. Wang and C. Lu, "Projection onto the capped simplex," *arXiv preprint arXiv:1503.01002*, 2015.

[38] B. Balle, J. Bell, A. Gascón, and K. Nissim, "The privacy blanket of the shuffle model," *CRYPTO*, 2019.

[39] V. Feldman, A. McMillan, and K. Talwar, "Stronger privacy amplification by shuffling for rényi and approximate differential privacy," in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2023, pp. 4966–4981.

[40] S. Wang, "Privacy amplification via shuffling: Unified, simplified, and tightened," *arXiv preprint arXiv:2304.05007*, 2023.

[41] J. V. Uspensky, *Introduction to mathematical probability*. McGraw-Hill Book Company, New York, 1937.

APPENDIX A
PROOF OF UNBIASNESS OF MEAN AND FREQUENCY ESTIMATORS

For ternary value estimation, we separately consider three cases of $\overline{\mathbf{R}}_j$ about the input. When $\overline{\mathbf{R}}_j = 0$, we get the equation about the output as: $\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]] = \mathbb{E}[[\![i_+ \in \mathbf{Z}]\!]] - \mathbb{E}[[\![i_- \in \mathbf{Z}]\!]] = p_f - p_f = 0$; when $\overline{\mathbf{R}}_j = 1$, we get $\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]] = p_t - p_r$; when $\overline{\mathbf{R}}_j = -1$, we get $\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]] = p_r - p_t$. Combining three equations, we conclude that $\frac{\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]]}{p_t - p_r} \equiv \overline{\mathbf{R}}_j$.

Similarly, for frequency estimation, we separately consider two cases of $\underline{\mathbf{R}}_j$ about the input. When $\underline{\mathbf{R}}_j = 0$, we can get $\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!]] = 2p_f$; when $\underline{\mathbf{R}}_j = 1$, we get $\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]] = p_t + p_r$. Combining two equations, we finally have $\frac{\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!]] - 2p_f}{p_t + p_r - 2p_f} \equiv \underline{\mathbf{R}}_j$.

APPENDIX B
DETAILED PROOF OF THE MSE BOUNDS ON EXSUB MECHANISM

We firstly fix the parameter m in the ExSub mechanism, and present the mean squared error in Lemma 1 as a formula of true/false/reverse positive rates.

***Lemma** 1:* In the (d', s, ϵ, m)-`ExSub` mechanism taking as input the $\mathbf{R}$, the mean squared errors of estimators are:

$$\sum_{j=1}^{d'} |\widehat{\overline{\mathbf{R}}}_j - \overline{\mathbf{R}}_j|_2^2 = \frac{s((p_t + p_r) - (p_t - p_r)^2) + (d' - s)(2p_f)}{(p_t - p_r)^2};$$

$$\sum_{j=1}^{d'} |\widehat{\underline{\mathbf{R}}}_j - \underline{\mathbf{R}}_j|_2^2 = \frac{s(p_t + p_r)(1 - p_t - p_r) + (d' - s)2p_f(1 - 2p_f)}{(p_t + p_r - 2p_f)^2}.$$

Proof: For the first equation, we separately consider three cases: $[\![i_+ \in \mathcal{S}_\mathbf{R}]\!] = 1$, $[\![i_- \in \mathcal{S}_\mathbf{R}]\!] = 1$, $[\![i_+ \in \mathbf{Y}_\mathbf{x}]\!] = 0 \; and \; [\![i_- \in \mathbf{Y}_\mathbf{x}]\!] = 0$. In the first case, the subtraction $[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]$ from Algorithm 3 is a random variable with probability distribution:

$$[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!] = \begin{cases} 1, & \text{with prob. } p_t; \\ 0, & \text{with prob. } 1 - p_t - p_r; \\ -1, & \text{with prob. } p_r. \end{cases}$$

Thus $Var[[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]] = (p_t + p_r) - (p_t - p_r)^2$ and $Var[\widehat{\overline{\mathbf{R}}}_j] = \frac{(p_t + p_r) - (p_t - p_r)^2}{(p_t - p_r)^2}$. Similarly, in the second case, the subtraction follows distribution:

$$[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!] = \begin{cases} 1, & \text{with prob. } p_r; \\ 0, & \text{with prob. } 1 - p_t - p_r; \\ -1, & \text{with prob. } p_t. \end{cases}$$

Consequently, we have $Var[\widehat{\overline{\mathbf{R}}}_j] = \frac{(p_t + p_r) - (p_t - p_r)^2}{(p_t - p_r)^2}$. In the third case, the subtraction follows distribution:

$$[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!] = \begin{cases} 1, & \text{with prob. } p_r; \\ 0, & \text{with prob. } 1 - 2p_r; \\ -1, & \text{with prob. } p_r. \end{cases}$$

Thus $Var[[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]] = 2p_r$ and $Var[\widehat{\overline{\mathbf{R}}}_j] = \frac{2p_r}{(p_t - p_r)^2}$. For every input $\mathbf{R}$, there are exact s indices $j \in [1 : d']$ satisfying the first or the second case, and exact $d' - s$ indices satisfying the third case. Consequently, the total error is $\frac{s((p_t + p_r) - (p_t - P_r)^2) + (d' - s)(2p_r)}{(p_t - p_r)^2}$.

For the second equation, we separately consider 2 cases: $[i_+ \in \mathcal{S}_\mathbf{R}] = 1 \; or [i_- \in \mathcal{S}_\mathbf{R}] = 1$, $[i_+ \in \mathcal{S}_\mathbf{R}] = 0 \; and \; [i_- \in \mathcal{S}_\mathbf{R}] = 0$. In the first case, the summation $[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!]$ is a Bernoulli variable of success rate $p_t + p_r$, thus $Var[[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!]] = (p_t + p_r)(1 - p_t - p_r)$ and $Var[\widehat{\underline{\mathbf{R}}}_j] = \frac{(p_t + p_r)(1 - p_t - p_r)}{(p_t + p_r - 2p_f)^2}$; In the second case, the summation is a Bernoulli variable of success rate $2p_f$, hence $Var[[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!]] = (2p_r)(1 - 2p_r)$ and $Var[\widehat{\underline{\mathbf{R}}}_j] = \frac{(2p_f)(1 - 2p_f)}{(p_t + p_r - 2p_f)^2}$. In every input $\mathbf{R}$, there are exact s indices $j \in [1 : d']$ satisfying the first case, and $d' - s$ indices satisfying the second case. Therefore, the total error is $\frac{s(p_t + p_r)(1 - p_t - p_r) + (d' - s)(2p_f)(1 - 2p_f)}{(p_t + p_r - 2p_f)^2}$. ■

We now prove the Theorem 2 by specifying an appropriate parameter m based on the previous lemma. For proving the error bound on the mean value, we separately consider two formulas $\frac{(p_t + p_r) - (p_t - p_r)^2}{(p_t - p_r)^2}$ and $\frac{2p_f}{(p_t - p_r)^2}$ in Lemma 1. As both of them involve with $\frac{1}{p_t - p_r}$, we firstly analyses the magnitude of $\frac{1}{p_t - p_r}$. Let C_1 denote the count $2^m \binom{d'}{m}$, C_2 denote count $\sum_{m'=0}^{m} 2^{m'} \binom{s}{m - m'} \binom{d - s}{m'}$, and C_3 denote the count $\sum_{m'=0}^{m-1} 2^{m'} \binom{s - 1}{m - 1 - m'} \binom{d - s}{m'}$, we have $\frac{1}{p_t - p_r} = \frac{C_1 - (1 - e^{\epsilon})C_2}{(1 - e^{-\epsilon})C_3}$. We bound these hyper-geometric counts in Lemmas 2 and 3, and arrive that $2C_2 d' \geq C_1(2d' - ms)$ and $c2 \leq 2sC_3$. Therefore, when $\epsilon = O(1)$, $m \leq \frac{d'}{s}$, and $m = \Theta(\frac{d'}{s})$, we have

$$\begin{aligned} \frac{1}{p_t - p_r} &= \frac{C_1 - (1 - e^{-\epsilon})C_2}{(1 - e^{-\epsilon})C_3} \\ &\leq \frac{C_1/C_3}{1 - e^{-\epsilon}} \\ &\leq \frac{(s + 2(d - s - m + 1)/m)(C_1/C_2)}{1 - e^{-\epsilon}} \\ &\leq \frac{(s + 2(d - s - m + 1)/m) \cdot (2d'/(2d' - ms))}{1 - e^{-\epsilon}} \\ &\leq c_1 \frac{s}{\epsilon} \end{aligned} \tag{2}$$

holds for some constant value $c_1 \in \mathbb{R}^+$.

***Lemma** 2:* Given $d', m, s \in \mathbb{R}^+$, inequality $2^m \binom{d'}{m}(2d' - ms) \leq 2d' \sum_{m'=0}^{m} 2^{m'} \binom{s}{m - m'} \binom{d - s}{m - m'}$ holds.

Proof: To prove the inequality, we only need to show that $(2^m \binom{d'}{m} - \sum_{m'=0}^{m} 2^{m'} \binom{s}{m - m'} \binom{d - s}{m - m'})) \leq s 2^{m-1} \binom{d' - 1}{m - 1}$.

Assuming there are d' different boxes each contains 2 different balls. Among them, there are s special boxes each holds exactly one special ball (i.e., another ball in the special box is non-special). Considering the process of selecting m boxes from d' boxes without replacement, then drawing one ball from each each selected box. There are totally $2^m \binom{d'}{m}$ combinations, the probability that at least 1 out of m ball is special is $\frac{(2^m \binom{d'}{m} - \sum_{m'=0}^{m} 2^{m'} \binom{s}{m - m'} \binom{d - s}{m - m'}))}{2^m \binom{d'}{m}}$. According to the union bound of probability, it is upper bounded by the summation of s probabilities, each of which is the probability that the i-th special ball from the i-th special box is selected (for $i \in [s]$): $\frac{2^{m-1}\binom{d'-1}{m-1}}{2^m\binom{d'}{m}}$. Consequently, we have $2^m\binom{d'}{m} - \sum_{m'=0}^{m} 2^{m'}\binom{s}{m-m'}\binom{d-s}{m-m'} \leq s2^{m-1}\binom{d'-1}{m-1}$. ■

***Lemma** 3:* Given $d', m, s \in \mathbb{R}^+$, inquality $\sum_{m'=0}^{m} 2^{m'}\binom{s}{m-m'}\binom{d-s}{m'} \leq (s + \frac{2(d-s-m+1)}{m})\sum_{m'=0}^{m-1} 2^{m'}\binom{s-1}{m-1-m'}\binom{d-s}{m'}$ holds.

Proof: Consider the first $m-1$ items in $\sum_{m'=0}^{m} 2^{m'}\binom{s}{m-m'}\binom{d-s}{m'}$, for every item that $m' \in [0, m-1]$, we have $2^{m'}\binom{s}{m-m'}\binom{d-s}{m'} = \frac{s}{m-m'}2^{m'}\binom{s-1}{m-1-m'}\binom{d-s}{m-m'} \leq s2^{m'}\binom{s-1}{m-1-m'}\binom{d-s}{m'}$. Now consider the last item, we have $2^m\binom{s}{0}\binom{d-s}{m'} \leq \frac{2(d-s-m+1)}{m}2^{m-1}\binom{s-1}{0}\binom{d-s}{m-1} \leq \frac{2(d-s-m+1)}{m}\sum_{m'=0}^{m-1} 2^{m'}\binom{s-1}{m-1-m'}\binom{d-s}{m'}$. Combining two formulas together, we get the inequality. ■

Now for the first formula, we specify $m \leq \frac{2d'}{se^\epsilon+s+1}$ and $m = \Theta(d'/s)$ that implies $p_r \leq 1/s$ and $p_r = \Theta(1/s)$, to get

$$\begin{aligned}&\frac{(p_t+p_r)-(p_t-p_r)^2}{(p_t-p_r)^2}\\ \leq&\frac{e^\epsilon p_r + p_r}{(p_t-p_r)^2}\\ \leq&\frac{c_1 s^2(e^\epsilon p_r + p_r)}{\epsilon^2}\\ \leq&\frac{(e^\epsilon+1)c_1 s^2 p_r}{\epsilon^2}\\ \leq&c_2\frac{s}{\epsilon^2}\end{aligned}$$

holds for some $c_2 \in \mathbb{R}^+$. Similarly for the second formula, under the same condition as the first one, we have

$$\frac{2p_f}{(p_t-p_r)^2} \leq \frac{2c_1 e^\epsilon s^2 p_r}{\epsilon^2} \leq \frac{c_3 s}{\epsilon^2}$$

holds for some $c_3 \in \mathbb{R}^+$. Combining two results together, we have the error on mean values bounded by:

$$\frac{sc_2 s + (d'-s)c_3 s}{\epsilon^2} \leq O(\frac{d's}{\epsilon^2}).$$

For proving the error bound on frequencies, we separately consider two formulas $\frac{(p_t+p_r)(1-p_t-p_r)}{(p_t+p_r-2p_f)^2}$ and $\frac{2p_f(1-2p_f)}{(p_t+p_r-2p_f)^2}$. Let C_4 denote the count $\sum_{m'=0}^{m-1} 2^{m'}\binom{s}{m-1-m'}\binom{d'-s-1}{m'}$ in the p_f, the $\frac{1}{p_t+p_r-2p_f}$ can be represented as:

$$\frac{C_1-(1-e^{-\epsilon})C_2}{(1-e^{-\epsilon})(2C_4-C_3)}.$$

According to their combinatoric meanings, we have $2(C_3 - C_4) \leq 2\sum_{m'=0}^{m-2} 2^{m'}\binom{s-1}{m-2-m'}\binom{d'-s-1}{m'} \leq \sum_{m'=1}^{m-1} 2^{m'}\binom{s-1}{m-1-m'}\binom{d'-s-1}{m'-1} \leq C_4$. Further plugging in the results from Lemmas 2 and 3, we get

$$\frac{1}{p_t+p_r-2p_f} \leq \frac{3C_1/C_3}{1-e^{-\epsilon}} \leq c_4\frac{s}{\epsilon^2}$$

holds for some constant value $c_4 \in \mathbb{R}^+$. For the first formula, since $(p_t+p_r)(1-p_t-p_r) \leq (p_t+p_r) \leq (e^\epsilon+1)p_r$ and $p_r = O(\frac{1}{s})$, we have $\frac{(p_t+p_r)(1-p_t-p_r)}{(p_t+p_r-2p_f)^2} \leq c_5\frac{s}{\epsilon^2}$. For the second formula, since $2p_f(1-2p_f) \leq 2p_f \leq 2p_t \leq 2e^\epsilon p_r$, we get $\frac{2p_f(1-2p_f)}{(p_t+p_r-2p_f)^2} \leq c_6\frac{s}{\epsilon^2}$. Combining two results together, we have the error on frequencies bounded by:

$$\frac{sc_5 s + (d'-s)c_6 s}{\epsilon^2} \leq O(\frac{d's}{\epsilon^2}).$$

APPENDIX C
PROOF OF MAXIMUM ABSOLUTE ERROR BOUNDS OF EXSUB MECHANISM

Considering the i-th mean value $\underline{\theta}_i$, according to Lemma 1, the expectation of $\hat{\underline{\theta}}_i - \underline{\theta}_i$ is 0. Furthermore $\hat{\underline{\theta}}_i - \underline{\theta}_i$ is the average of n independent random variables $\frac{[\![H(i_+)=z]\!]-[\![H(i_-)=z]\!]}{p_t-p_r}$, every of which lies in the range:

$$[\frac{-1}{p_r-p_r}, \frac{1}{p_t-p_r}].$$

When $[\![i_+ \in \mathbf{Y_x}]\!] = 1\ or [\![i_- \in \mathbf{Y_x}]\!] = 1$, the variable has variation of $\frac{(p_t+p_r)-(p_t-p_r)^2}{(p_t-p_r)^2}$; when $[\![i_+ \in \mathbf{Y_x}]\!] = 0\ and\ [\![i_- \in \mathbf{Y_x}]\!] = 0$, the variable has variation of $\frac{2p_f}{(p_t-p_r)^2}$. In both cases, assume that $m \leq \frac{d'}{s}$ and $m = \Theta(\frac{d'}{s})$, we have $\frac{(p_t+p_r)-(p_t-p_r)^2}{(p_t-p_r)^2} \leq \frac{c_2}{s}$ and $\frac{2p_f}{(p_t-p_r)^2} \leq \frac{c_3 s}{\epsilon^2}$ holds with some constant $c_2, c_3 \in \mathbb{R}^+$ for any $\epsilon = O(1)$ and any $s \in \mathrm{R}^+$ (see detail in Appendix B).

According to the Bernstein inequalities on n zero-mean bounded random variables [41], we have:

$$\mathbb{P}[|\hat{\underline{\theta}}_i - \underline{\theta}_i| \geq \frac{\alpha}{n}] \leq 2\exp(-\frac{\alpha^2/2}{n^2\mathrm{Var}[\hat{\underline{\theta}}_i - \underline{\theta}_i] + \alpha/(3p_t-3p_r)}).$$

Assume that $m \leq \frac{d'}{s}$ and $m = \Theta(\frac{d'}{s})$, we have $1/(p_t-p_r) \leq \frac{c_1 s}{\epsilon}$ holds for any $\epsilon = O(1), s \in \mathbb{R}^+$ with some constant $c_1 \in \mathbb{R}^+$ (see Equation 2).

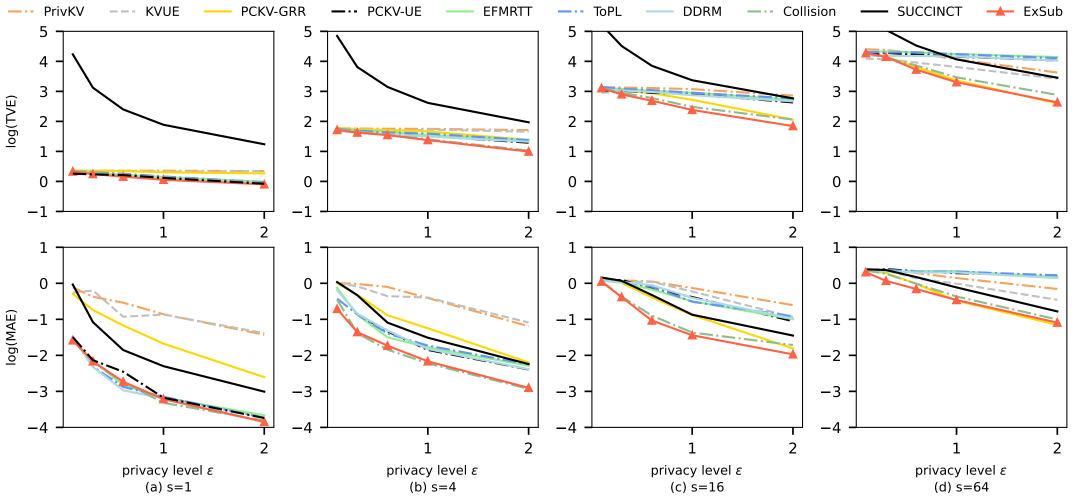
When $\alpha \leq \frac{3c_3 n}{c_1\epsilon}$, we get $\mathbb{P}[|\hat{\underline{\theta}}_i - \underline{\theta}_i| \geq \frac{\alpha}{n}] \leq 2\exp(-\frac{\epsilon^2\alpha^2/2}{2c_3 ns})$. Consequently, with probability $1-\beta$, we get $|\hat{\underline{\theta}}_i - \underline{\theta}_i| \leq \sqrt{\frac{2c_3 s\log(2/\beta)}{\epsilon^2 n}}$ (when $\sqrt{\frac{2c_3 s\log(2/\beta)}{\epsilon^2 n}} \leq \frac{3c_3 n}{c_1\epsilon}$).

Now consider all $i \in [1:d]$, applying the union bound on d' tail probabilities, we conclude that: if $\sqrt{\frac{2c_3 s\log(2d'/\beta)}{\epsilon^2 n}} \leq \frac{3c_3 n}{c_1\epsilon}$, with probability $1-\beta$, then the inequality $\max_{j=1}^{d}|\hat{\underline{\theta}}_i - \underline{\theta}_i| \leq O(\sqrt{\frac{s\log(d'/\beta)}{\epsilon^2 n}})$ holds.

APPENDIX D
COMPLEMENTARY EXPERIMENTAL RESULTS

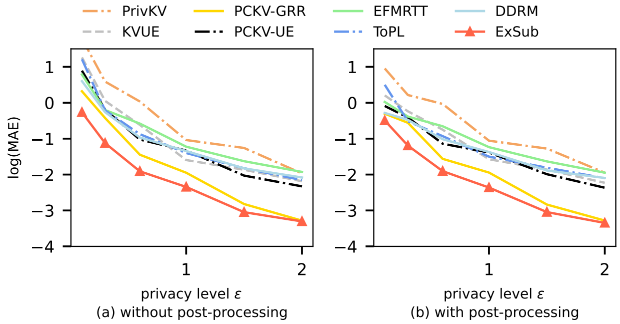
FutureRand [29] focuses on asymptotic estimation performance. Here, we extend this work by presenting experimental results with a large sparsity parameter in Table IV. In line with the results shown in Table II, the FutureRand mechanism suffers from significantly more errors than other mechanisms, even in an asymptotic setting (for instance, when $\epsilon \rightarrow 0$ or s is large). The considerable empirical error of FutureRand makes it impractical in these experimental settings.

Additionally, we provide supplementary experimental results in Figures 10, 11 (on the Stock dataset) and Figures 12, 13 (on the Trajectory dataset) to complete the missing error measurements in the main content. As expected, the performance gaps between competitive mechanisms are largely consistent across various metrics (such as MAE/TVE, and with or without post-processing).



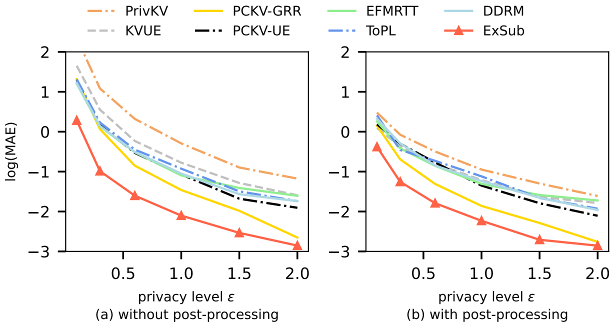
(a) s=1 (b) s=4 (c) s=16 (d) s=64

Fig. 10. Error results with post-processing with $n = 10000$, $d' = 256$ and sparsity parameter s varies from 1 to 64.



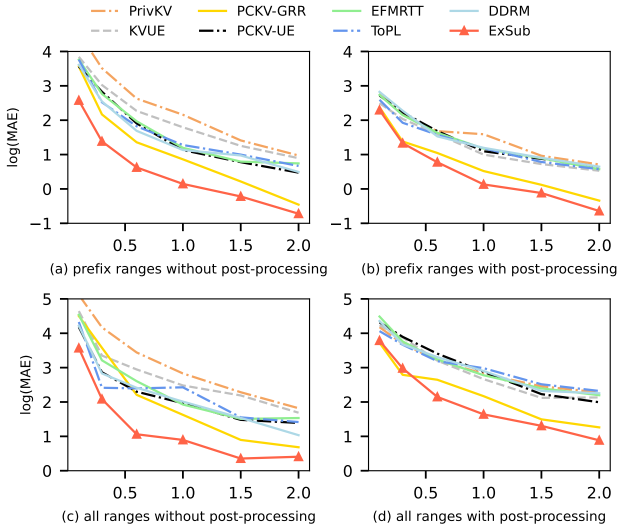
(a) without post-processing (b) with post-processing

Fig. 11. MAE results for mean queries on Stock dataset.



(a) without post-processing (b) with post-processing

Fig. 13. MAE results for mean queries on Trajectory dataset.



(a) prefix ranges without post-processing (b) prefix ranges with post-processing

(c) all ranges without post-processing (d) all ranges with post-processing

Fig. 12. MAE results on range queries of Trajectory dataset.

TABLE IV
ERROR RESULTS WITHOUT POST-PROCESSING UNDER EXTREMELY SPARSITY PARAMETERS ($n = 50000$, $d' = 1024$, $s = 100$).

	TVE results				
	$\epsilon = 0.001$	$\epsilon = 0.01$	$\epsilon = 1.0$	$\epsilon = 3.0$	$\epsilon = 5.0$
FutureRand [29]	3.7e+5	3.6e+4	341.8	117.2	84.2
Collision [22]	6.7e+4	6.8e+3	61.9	17.0	10.4
SUCCINCT [23]	6.7e+4	6.8e+3	85.6	54.9	43.2
ExSub	5.4e+4	5.5e+3	49.8	11.3	6.91
	MAE results				
	$\epsilon = 0.001$	$\epsilon = 0.01$	$\epsilon = 1.0$	$\epsilon = 3.0$	$\epsilon = 5.0$
FutureRand [29]	3161.1	300.4	2.71	0.90	0.72
Collision [22]	526.4	55.6	0.47	0.17	0.14
SUCCINCT [23]	555.4	55.3	0.66	0.41	0.35
ExSub	426.3	45.2	0.41	0.14	0.11