SaO_2: Optimal Online Locally Private Data Analytics with Stronger Shuffle Amplification

Shaowei Wang, Jin Li
Institute of Artificial Intelligence and Blockchain, Guangzhou University
Guangzhou, China
{wangsw,lijin}@gzhu.edu.cn

Bangzhou Xin
Chinese Academy of Engineering Physics
Mianyang, China
xbw401@gmail.com

Wei Yang
University of Science and Technology of China
Hefei, China
qubit@ustc.edu.cn

ABSTRACT

Online data analytics with privacy protection has broad applications in practice. Despite many efforts devoted to this problem under local differential privacy, it still suffers large gaps from the offline counterpart in terms of utility and functionality. This work shows private data analytics can be conducted online without excess utility loss and even has stronger amplification effects in the shuffle model.

We design an optimal and streamable mechanism for locally private sparse vector estimation, which empowers diverse online analytics (i.e., mean and range query) on general streaming binary vectors (e.g., multi-dimensional binary, categorical, or set-valued data from users). The mechanism exploits the negative correlation of occurrence events in the sparse vector to reach optimal error rate, meanwhile requiring only streamable computations on the input during its data-dependent phase. To further break the error barrier due to local differential privacy, we analyze shuffle privacy amplification bounds of the proposed streamable mechanism, and show that online responding & shuffling enables strictly stronger privacy amplification effects than the classical offline shuffle model. We validate the performance of our proposals under both synthetic and real-world datasets, and show 40% $\sim$ 60% error reductions over state-of-the-art approaches.

PVLDB Reference Format:
Shaowei Wang, Jin Li, Bangzhou Xin, and Wei Yang. SaO_2: Optimal Online Locally Private Data Analytics with Stronger Shuffle Amplification. PVLDB, 16(1): XXX-XXX, 2023.
doi:XX.XX/XXX.XX

1 INTRODUCTION

Online data analytics helps track user status and behaviors over time. It plays a vital role in continual decision making and service quality improving (e.g., for healthcare [45], location-based services [7], and other prevalent Internet services [11, 16]), but is facing with severe privacy challenges. The temporal user data, such as demographic data (e.g., ages, and locations) and activity data (e.g., sensor readings, and visited pages/Apps), reveal sensitive information about individuals. Meanwhile, lawful privacy-related regulations are increasingly strengthened world-widely, for instances, the GDPR in the Europe Union [37], the CCPA in California [21], and the Personal Information Protection Law in China [10].

The local differential privacy [14, 26] is widely adopted for online user analytics in industry (e.g., by Google Chrome [16], operating systems from Microsoft [11] and Apple [33, 34]). In contrast to sporadic/offline data collecting that is well-studied in the community (see [48, 50] for reviews), the online version has several unique merits and poses extra challenges:

(1) accumulated privacy loss. The overall privacy leakage grows with the number of reports. Adversaries may exploit multiple correlated reports to infer more accurate information [47].

(2) real-time responding. The advantage of online analytics stems from the real-time feedback, which however requires private mechanisms being streamable. As a comparison, an offline mechanism begins sanitation after observing the complete input.

One direct approach is dividing the budget ϵ of local privacy into T parts and sanitizing datum at each timestamp $t \in [T]$ independently. However, the mean squared error will scale with $O(T/(1/T)^2) = O(T^3)$ (hides the $1/\epsilon^2$ factor). Researchers recently observe that the user data often changes few times during the lifetime [15], and propose to exploit the sparsity in change. They represent the streaming data as sparse ternary vectors recording the temporal changes, then feed into streamable locally private mechanisms, which sample multiple dimensions or non-zero entries and sanitize them independently. As a result, they reduce the error factor to $O(T^2 \log^2 T)$ (e.g., in [30, 51, 52]) or $O(Ts^2 \log^2 T)$ (e.g., in [15, 22, 49]), where s is the sparsity parameter denoting the number of changes in a stream.

Though much progress has been made on online locally private data analytic, it is far legged behind the offline counterpart (w.r.t. utility performance and functionality). **1)** For the core sub-problem of locally private ternary vectors estimation, a recent work [40] proposes using local hash [41] on all non-zero entries then releasing one hashed index, to reach the minimax lower error bound of $O(Ts/\epsilon^2)$ (i.e., reach the bound $O(Ts \log^2 T/\epsilon^2)$ for the original problem), which surpasses current online streamable mechanisms by a factor of T/s or s; another recent work [53] proposes to map all non-zero entries with random weights to one bucket and then adds noises on the summated weights, to match the optimal maximum absolute error and to reach the mean squared error bound of $O(Ts \log n/\epsilon^2)$. **2)** At the meantime, to break the error barrier inherited in local privacy, offline private data analytics has embraced the shuffle model [1, 3, 5, 20], where each user can hide the private view among views from other users and adopt a higher budget in the local. The data utility in the shuffle privacy model has the potential to approximate the central model of differential

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 16, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

Table 1: Comparison of ϵ-LDP mechanisms for the core sub-problem of sparse ternary vector estimation with d' dimensions and s non-zero entries. The *comput.* column presents computational costs; the *comm.* presents communication costs. The MSE presents mean squared error bounds. The online column indicates whether the mechanism can be explicitly made online.

approaches	user comput.	user-server comm.	server comput.	MSE$\times(n\epsilon^2)$	optimal?	online?
PrivKV [51]	$O(\log d')$	$O(\log d')$	$O(n)$	$O(d'^2)$	✗	✓
KVUE [32]	$O(\log d')$	$O(\log d')$	$O(n)$	$O(d'^2)$	✗	✓
PCKV-GRR [22]	$O(\log d')$	$O(\log d')$	$O(n)$	$O(d'^2)$	✗	✓
PCKV-UE [22]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
EFMRTT [15]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
ToPL [42]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
DDRM [49]	$O(d')$	$O(d')$	$O(nd')$	$O(d's^2)$	✗	✓
Collision [40]	$O(s)$	$O(\log s)$	$O(nd')$	$O(d's)$	✓	✗
SUCCINCT [53]	$O(s)$	$O(\log s)$	$O(nd')$	$O(d's\log n)$	✓	✗
this work	$O(d')$	$O(\min(d', d'/s\log d')$	$O(nd')$	$O(d's)$	✓	✓

privacy [4]. The area of online locally private data analytics in the shuffle model is inadequately explored. **3)** Besides mean estimation on binary/categorical value considered by existing online private mechanisms, the offline studies support more types of user data (e.g., multi-dimensional vector [38, 43]) and analyzing tasks (e.g., range queries [8, 27]). Real-world user data is often in multi-dimensional forms, such as demographic and activity data are jointly collected to study their occurrences; the server may perform range queries that provide summarized statistics over time (e.g., App usage frequency within a week) and enable sophisticated temporal analyses.

1.1 Our Contributions

In this work, we bring up a general and optimal protocol for online locally private data analytics, supporting multi-dimensional data and various analyzing tasks (i.e., mean query, frequency query, and range query). In contrast to pessimistic results in previous studies, our results show that online private analytics suffers no excess utility loss than their offline counterparts (neither in constant factor), and may even enjoy stronger privacy amplification effects in the shuffle privacy model.

Theorem 1.1. *Consider n users each holds $\mathbf{x}_{i,t} \in \{0,1\}^d$ at timestamp $t \in [T]$, assuming there are at most s changes across T timestamps for every user, there is a streamable local ϵ-differential private mechanism that obtains an unbiased estimator of $\{\frac{1}{n}\sum_{i\in[n]}\mathbf{x}_{i,t}\}_{t\in[T]}$ with mean squared error $O(\frac{dTs\log^2 T}{n\epsilon^2})$. It incurs $O(d)$ memory consumption, $O(dT)$ computational costs, and $O(\frac{dT}{s})$ communication costs on the user side.*

Without loss of generality, we assume every user possesses a streaming binary vector $\{0,1\}^{d\times T}$, in which each datum $\{0,1\}^d$ at timestamp $t \in [T]$ covers binary/categorical/set-valued data as special cases. After representing the streaming data as hierarchical ternary vectors $\{-1,0,1\}^{d'}$ indicating its temporal residues at different granularity, we sanitize one of the ternary vectors in the hierarchical tree with the newly proposed ExSub mechanism online. The ExSub mechanism exploits the negative correlation of two options $\{-1,1\}$ in each entry (i.e., two options never show up together in the input) and outputs an **ex**clusive **sub**set of entries, to minimize estimation error and to match the minimax error lower bound. Moreover, the ExSub mechanism is designed to use streamable primitives only, and the exclusive subset can be sampled with a recursive implementation of *sequential random sampling* [31]. Finally, the server reconstructs statistics of interests from estimators of ternary vectors. Taking into consideration the importance of each hierarchical level for a specific task, we give calibrated hierarchy selection strategies and thus reduce the estimation error. We describe the properties of the proposal in the Theorem 1.1 informally, and compare it with existing approaches in Table 1 for the core sub-problem of ternary vector aggregation. Lastly, in the shuffle model, we propose online shuffling that breaks linkage between multiple responses from one user, and indicate it induces stronger effects of privacy amplification than the classical shuffling.

To summarize, the contributions of this work are as follows:

- We present a protocol $\mathsf{SaO_2}$ for online locally private data analytic, which handles general multi-dimensional data (e.g., binary, categorical, set-valued data, and multi-dimensional vector) and various analyzing tasks (i.e., mean estimation, frequency estimation, and range queries).
- We design an optimal locally private mechanism ExSub for the core building block of the protocol: sparse ternary vector aggregation. The mechanism maximizes data utility meanwhile only relying on streamable computations. Compared with best known prior mechanisms, it reduces 10%-30% error.
- We provide optimized hierarchy selection strategies to simultaneously reduce user-side computational/communication costs and increase server-side estimation utility.
- For breaking the error barrier of local privacy, we propose online shuffling for data analytics and derive a privacy amplification upper bound. The new bound shows that online shuffling provides strictly more privacy amplification than the classical shuffling.
- Through extensive experiments on both synthetic and real-world datasets, we show our proposals reduce 40%-60% error when compared with existing online approaches.

1.2 Organization

The remaining content is organized as follows. Section 2 retrospects related works. Section 3 gives background knowledge and the problem formulation. Section 4 presents the general protocol

for online private data analytics. Section 5 describes the design of ExSub mechanism and provides both offline and online implementations. Section 6 analyzes the privacy amplification effect of online shuffling. Section 7 reports experimental results. Finally, Section 8 concludes the paper.

2 RELATED WORKS

In this part, we retrospect locally private data analytics protocols and privacy amplification results in the classical shuffle model.

2.1 Online Locally Private Protocols

The prevalent RAPPOR [16] uses a Bloom-filter and binary randomized response [12] to condense & sanitize user data, it reuses private views until local data has changed (i.e., memoization). For continuous data stream, Ding *et al.* [11] discretize local values to sparsify temporal changes and then use randomized response for sanitizing discrete value. The work [24] partitions users into multiple groups each with the same report pattern, and then sanitizes each report independently. These approaches (i.e., [11, 16, 24]) use the memoization technique to reduce privacy consumption in continual collection, but only ensure a weaker version of local differential privacy that is restricted to the data domain with same change pattern (i.e., the timestamps of s changes in the data stream must be the same). However, the pattern of changes is sensitive information about users, and requires rigorous protection.

For streaming online data analyses with rigorous local differential privacy protection, one straightforward way is splitting the privacy budget into T parts and sanitizing each value at every timestamp. The work [15] introduces the hierarchical tree structure with height $O(\log T)$ for recording temporal changes of the local data, then sanitizes one of the $\{-1, 1\}$ change with whole budget ϵ and thus improves the estimation error bound from the factor $O(T^2)$ to $O(s^2 \log^2 T)$. The work [42] studies online numerical data collection, they propose to sanitize each clipped value independently and the mean squared error scales with $O(s^2 \log^2 T)$. Latterly, the DDRM [49] proposes to select k changes from all s changes, and then sanitizes every selected change with budget $\frac{\epsilon}{k}$, the error also scales with $O(s^2 \log^2 T)$. The theoretical work [29] proposes randomizing every value with the binary randomized response meanwhile truncating sensitive outputs (e.g., binary vectors that are too close or too far to the input), to save budget to $O(\sqrt{s})$. However, their proposal is only of theoretical interests, and has extremely poor empirical performance (see Section 7). As opposed to existing works with sub-optimal utility, our proposal reaches optimal error rate of $O(s \log^2 T)$ and handles universal binary vector streams (or slowly changing vector streams), such as set-valued streams and multi-dimensional categorical streams.

2.2 Offline Locally Private Protocols

After the general hierarchical difference transformation [15] that utilizes the sparsity in change (see Section 4), the online data analytics can be reduced to the problem of online private sparse vector aggregation. The seminal work [51] on offline private sparse ternary data (i.e., key-value data) proposes to uniform-randomly select one dimension from total d' dimensions, and then sanitizes the value of the selected dimension with full privacy budget ϵ, the subsequent work [32] provides a slightly variant of transition matrix to sanitize the ternary value $\{-1, 0, 1\}$. However, since the effective number of users on each dimension decreased from n to n/d', their error dependencies on dimension are $O(d'^2)$. The PCKV [22] randomly selects one non-zero entry from all s non-zero entries, and then sanitizes the corresponding dimension (among d' dimensions) and value with categorical locally private mechanisms (e.g., Generalized Randomized Response [46], Unary Encoding [16]). Their errors scale with $O(d's^2)$ for PCKV-UE or $O(d'^2)$ for PCKV-GRR. The underlying ternary mechanism of DDRM [49] is an extension of the PCKV-UE selecting k non-zero entries, the error bound still scales with $O(\frac{d'}{(\epsilon/k)^2}(\frac{s}{k})^2) = O(d's^2)$. Latterly, the Collision mechanism [40] maps all non-zero entries to a Bloom-filter with local hash [41], then outputs one index of the Bloom-filter. It reaches the optimal error dependence $O(d's)$, but the outputting probability of each index in the Bloom-filter is not determinable until all non-zero entries are ready. The most recent work [53] proposes to map all non-zero entries to one bucket with pseudo-random weight -1 or $+1$, and then clips the bucketed value with norm $O(\sqrt{\log n})$ and adds Laplace noises. It reaches optimal maximum absolute error and reaches mean squared error dependence $O(d's \log n)$, but is also not streamable. When trying to adopt previous mechanisms for online analytics, some sub-optimal approaches [32, 51, 52] can be made online via streamable uniform sampling. Meanwhile, for current mechanisms achieving optimal utility (i.e., [40, 53]), they require entire information of non-zero entries to begin sanitation, rendering them intractable for streamable implementation. It might seem that the local privacy must sacrifice utility for online implementation. Fortunately, as our proposals suggested, it is not true.

2.3 Privacy Amplification via Shuffling

The shuffle model [3] lies in the middle ground of local and central model of differential privacy. It potentially reaches more balanced utility and privacy/security trade-offs. The shuffle model employs a semi-trusted shuffler (i.e., anonymous channels, edge servers) to uniform-randomly shuffles (semi)-private messages from users, before exposing them to the server. When the message from each user is sanitized by a local mechanism with privacy level ϵ, the seminal work [15] shows each message is covered by "privacy blanket" and actually satisfies $(\sqrt{144\epsilon^2 \log(1/\delta)/n}, \delta)$-DP. This phenomenon is known as privacy amplification via shuffling [1, 5]. Recently, the work [18] gives asymptotically optimal privacy amplification bounds for shuffled messages, via clone (mixture) analyses on privatized messages. Another line of works on the shuffle privacy model discards the constraint that message(s) from one user must be locally private, and allows each user to send multiple messages to the shuffler (e.g., in [2, 6, 20]). In the setting of online real-time data collecting and analyzing, the *online shuffling* naturally arises, which shuffles datum from users at every timestamp independently. Our analyses in Section 6 demonstrate that the online shuffling bridges the advantages of the traditional single-message and multi-message shuffle models, as it ensures that multiple messages from each user satisfy local ϵ-DP thus prevents attacks from the shuffler or other parties when anonymity fails, and surpasses the intrinsic error due to the limitation of sending one message [19].

3 BACKGROUND

We let $[n]$ denote $\{1, ..., n\}$ and $[n_1 : n_2]$ denote $\{n_1, n_1 + 1, ..., n_2\}$. The $[\![statement]\!]$ denotes Iversion function, the value is 1 if the *statement* is true and is 0 otherwise.

3.1 Differential Privacy

Let $\mathbf{x}_{j,t} \in \{0,1\}^d$ denote the binary vector of the user j at the timestamp $t \in [T]$, and let $\mathbf{x}_{j,R}$ denote the data $\{\mathbf{x}_{j,t}\}_{t \in R}$ given a timestamp subset $R \subseteq [T]$, we aim at protecting the privacy of the whole streaming data $\mathbf{x}_j = \mathbf{x}_{j,[T]} \in \{0,1\}^{d \times T}$ for every user j. We assume 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 normally much smaller than $d \cdot T$.

We denote the dataset $[\mathbf{x}_1, ..., \mathbf{x}_n]$ from n users as S. For two datasets $S, S' \in \{0,1\}^{d \times T \times n}$ that are of the same size and differ in one row, they are called *neighboring datasets*. Let K denote a randomized mechanism for sanitizing a dataset, the centralized differential privacy with parameter (ϵ, δ) is as follows.

Definition 3.1 (Central (ϵ, δ)-DP [14]). Let $\mathcal{D}_K$ denote the output domain, a randomized mechanism K satisfies (ϵ, δ)-differential privacy iff for any neighboring datasets S, S', the $K(S)$ and $K(S')$ are (ϵ, δ)-indistinguishable. That is, for any outputs $\mathbf{t} \subseteq \mathcal{D}_K$,

$$\mathbb{P}[K(S) \in \mathbf{t}] \leq \exp(\epsilon) \cdot \mathbb{P}[K(S') \in \mathbf{t}] + \delta.$$

The local differential privacy protects data privacy locally in the user side. Similar to the central version, it poses indistinguishability constraints on the data domain of one user (i.e., neighboring datasets with one user).

Definition 3.2 (Local ϵ-DP [26]). 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 $\mathbf{z} \in \mathcal{D}_K$,

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

Let K denote the local randomizer of each user, and $t_i = K(\mathbf{x}_j)$ the local private view from user j, when semi-trusted shufflers (or anonymous channels) $\mathcal{S}$ lie between users and the server, the server only observes the uniform-randomly permuted messages $\{t_1, ..., t_n\} = \mathcal{S}[t_1, ..., t_n]$ from n users and the privacy is amplified from the centralized perspective. The shuffle (ϵ, δ)-differential privacy is defined as follows.

Definition 3.3 (Shuffle (ϵ, δ)-DP [15]). Given a shuffler $\mathcal{S}$, the randomized mechanism K satisfies shuffle (ϵ, δ)-differential privacy iff the outputting unordered set $\{t_1, t_2, ..., t_n\}$ satisfies centralized (ϵ, δ)-DP constraints for any neighboring datasets.

Exponential Mechanism. A universal tool to design (local) ϵ-differential private mechanisms is exponential mechanism [28], which outputs $\mathbf{z}$ given an input S with probability proportional to

$$\exp\left(\frac{\mathsf{utility}(S, \mathbf{z}) \cdot \epsilon}{2\Delta}\right).$$

The utility is some arbitrary function and $\Delta = \max_{z,S,S'} |\mathsf{utility}(S, \mathbf{z}) - \mathsf{utility}(S', \mathbf{z})|$. The constant 2 in the denominator coming from varying probability normalization factors, when the normalization factor is the same for all input (e.g., in the ExSub mechanism in Section 5), it can be safely removed.

3.2 Shuffle Privacy Analyses via Clone Analogy

We retrospect the classical clone method [18] of locally private mechanisms for analyzing the shuffle privacy amplification effect. Without loss of generality, we assume two neighboring datasets $S = \{x_1 = a, x_2, ..., x_n\}$ and $S' = \{x_1 = b, x_2, ..., x_n\}$ differ at the 1st user. Consider a local ϵ-differential private mechanism K taking as input data from some domain $\mathcal{D}_{\mathcal{X}}$. The clone method utilizes the fact that the distribution of $K(x_j)$ is a mixture distribution of $K(a)$ or $K(b)$ (for any $a, b, x_j \in \mathcal{D}_{\mathcal{X}}$):

$$K(x_j) = \begin{cases} K(a), & \text{with probability } e^{-\epsilon}/2; \\ K(b), & \text{with probability } e^{-\epsilon}/2; \\ \mathcal{W}_{a,b}(x_j), & \text{else.} \end{cases}$$

The $K(a)$ or $K(b)$ coming from all users are called clones, and their counts is denoted as C. The clone probability from each user is $p = \frac{1}{2e^{\epsilon}} + \frac{1}{2e^{\epsilon}} = \frac{1}{e^{\epsilon}}$. A special case is when $x_j = a$, the $K(x_j)$ is a clone of $K(a)$ and $K(b)$ with probability $\frac{1}{e^{\epsilon}+1}$ and $\frac{1}{e^{\epsilon}+1}$ respectively; so as $x_j = b$. The clone method builds a connection between the shuffle privacy amplification bound and the number of clones (in Lemma 3.4).

Lemma 3.4 (Shuffle Privacy via Clone Method [18]). *Let $p \in [0,1]$ denote the clone probability and $q \in [0.5, 1]$ denote the mutual-clone probability, consider the process that we first sample $C \sim Binomial(n-1, p)$, then $A \sim Binomial(C, \frac{1}{2})$, and sample $B \sim Bernoulli(q)$. If random variables $P = (A + B, C - A + 1 - B)$ and $Q = (A + 1 - B, C - A + B)$ are (ϵ_c, δ)-indistinguishable, then the unordered messages $T_a = \{\mathcal{K}(a), \mathcal{K}(x_2), ..., \mathcal{K}(x_n)\}$ and $T_b = \{\mathcal{K}(b), \mathcal{K}(x_2), ..., \mathcal{K}(x_n)\}$ are (ϵ_c, δ)-indistinguishable.*

The variable A (or $C - A$) records the number of clones $K(a)$ (or $K(b)$) from $n - 1$ users other than the 1st user. By bounding on the divergence between $(A + B, C - A + 1 - B)$ and $(A + 1 - B, C - A + B)$ where $p = \frac{1}{e^{\epsilon}}$ and $q = \frac{e^{\epsilon}}{e^{\epsilon}+1}$, the clone method [18, Appendix A] shows that they are (ϵ_c, δ)-indistinguishable with

$$\epsilon_c = \log\left(1 + \frac{e^{\epsilon} - 1}{e^{\epsilon} + 1} \cdot \frac{2\sqrt{\log(4/\delta)\Phi/2} + 1}{\Phi/2 - \sqrt{\log(4/\delta)\Phi/2}}\right),$$

where $\Phi = \frac{n}{e^{\epsilon}} - \sqrt{\frac{3n}{e^{\epsilon}} \log(4/\delta)}$ is the lower $\frac{\delta}{2}$-bound on the number of clones C (i.e., $C \geq \Phi$ with probability at least $1 - \delta/2$).

3.3 Online Mean and Range Analytics

Online user data analytics aims to provide real-time statistics about 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 issue range queries, such as the total value over a 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 perspective of users, they occasionally respond with some information at several (or all) timestamps. We let $\mathbf{z}_{j,t}$ denote the message user i responses at the timestamp t; if they responded no information, we deemed it as $\perp$. When enhanced with differential privacy, all messages $\{\mathbf{z}_{j,1}, ..., \mathbf{z}_{j,T}\}$ across timestamps $[T]$ are ensured to be locally ϵ-DP. That is, the output $\mathbf{z}$ in the Definition 3.2 represents $\{\mathbf{z}_{j,1}, ..., \mathbf{z}_{j,T}\}$. In our protocol, every message $\mathbf{z}_{j,t}$ belongs to $\{-1, +1, \perp\}^d$.

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.

$t' \leftarrow 0$
$prediction \leftarrow \{0\}^d$
for $t \in [T]$ **do**
  if $t \mod r^h = 0$ **then**
    $t' \leftarrow t' + 1$
    $\mathbf{R}_{j,t',h} \leftarrow \mathbf{x}_{j,t} - prediction$
    $prediction \leftarrow \mathbf{x}_{j,t}$
    yield $\mathbf{R}_{j,t',h}$
  end
end

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**
    $v \leftarrow v + \sum_{t''=\lfloor \frac{t}{r^{h+1}} \rfloor \cdot r+1}^{t'} \hat{\mathbf{R}}_{j,t'',h}$
  end
end
return v

4 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 information from online ternary statistics.

(1) The ternary representation. Following best-practices for utilizing sparsity in change [15, 49], 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 $\{-1, 0, 1\}^d$.

The 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 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 $\widehat{\mathbf{x}}_{j,t}$ costs $O(r \log_r T)$ 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 mean squared estimation error grows at least linearly with the number of non-zero entries (see Section 2) and every element at level h can be used at most r^h times for recovering the binary vector, every user in our framework selects only one hierarchy level to response. A side consequence is that the memory/computation/communication costs on the user side is reduced by about a factor of H.

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 parameters (e.g., d, T, s, ϵ), and is deferred to the fourth component.

(3) Online private ternary aggregation. After the ternary residue representation and hierarchical selection, every user now concerns with streaming ternary vectors $\{\mathbf{R}_{j,t',h}\}_{t'\in[T_h]}$ at the level h, which has at most s non-zero entries. The user now sanitizes the streaming ternary vectors with an online private mechanism, and submits outputting messages when timestamp $t \mod r^h = 0$ holds. The server receives 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:

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

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

Providing the query task and the error characterization of the ternary privatization mechanism, we can now specify the hierarchy selection portions $\mathbf{W} \in \Delta_H$. Given a fixed $\mathbf{W}$, the number of users selecting the level h is $n \cdot \mathbf{W}_h$. According to the Theorem 5.3 on the proposed ExSub, the mean squared error of each residue estimator $\hat{\mathbf{R}}_{i,t',h}$ is bounded by $O(\frac{s}{\epsilon^2})$ (see detail in Appendix B), 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 sampling error that is negligible when ϵ is small).

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 [9], according to the variance bound on summed variables: $\mathrm{Var}[A+B] \le 2\mathrm{Var}[A] + 2\mathrm{Var}[B]$, 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):

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

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

Prefix Range queries with $t_1 \equiv 0$. Every $\mathbf{R}_{*,t',h}$ contributes at most $(r-1)r^h$ times in one range query, and appears in at most $T - r^h$ queries. Therefore, the factor on each residue variance is $\sum_{j=1}^{T}(r^h)^2 = O((r-1)^2 r^{2h}(T-r^h))$. The error of all queries $\{\mathbf{x}_{*,[0:t_2]}\}_{t_2=1}^{T}$ is bounded by $O(\sum_h \frac{dsT(r-1)^2 r^h (T-r^h)}{n \cdot \mathbf{W}_h \cdot \epsilon^2})$. Therefore, specifying $\mathbf{W}_h = \frac{r^h(T-r^h)}{\sum_{h'} r^{h'}(T-r^{h'})}$ approximately minimizes the error, and the minimum is $O(\frac{dsT^3(r-1)^2 \log_r T}{n \cdot \epsilon^2})$.

All Range queries with $t_1 \in [T], t_2 \in [t_1 : T]$. Every $\mathbf{R}_{*,t',h}$ contributes at most $(r-1)r^h$ times in one range query and can present in at most $T^2/4$ queries, thus the factor on each residue variance is $O((r-1)^2 r^{2h} \cdot T^2)$. 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 $O(\sum_h \frac{dsT^3(r-1)^2 r^h}{n \cdot \mathbf{W}_h \cdot \epsilon^2})$. Therefore, assigning $\mathbf{W}_h = \frac{r^h}{\sum_{h'} r^{h'}}$ approximately minimizes the error, and the minimum is $O(\frac{dsT^4(r-1)^2 \log_r T}{n \cdot \epsilon^2})$.

We note that similar protocols can be seen in [15, 29, 49], but they pay no effort to range queries. Specially, in the shuffle model, each user sends messages to the shuffler (instead of directly to the server) at each timestamp. See Section 6 for detail.

5 THE EXSUB MECHANISM

We now present the locally private mechanism for the streaming ternary vectors $\{\mathbf{R}_{j,t',h}\}_{t' \in [T_h]}$ at the level h, coined as Exclusive Subset mechanism (ExSub). Before delving into the online implementation, we here focus on the design and properties (privacy and utility guarantees) of its offline version, which sees the whole streaming ternary vectors in advance.

Data Preparation. We deem $\{\mathbf{R}_{j,t',h}\}_{t' \in [T_h]}$ from the level h as a ternary vector $\mathbf{R}$ with $d \cdot T_h$ dimensions and at most s non-zero entries. For simplicity in analyses, we append s stub entries to the vector, and let d' denote the length $d \cdot T_h + s$. We then take a set perspective on the sparse ternary vector. Let i_- and i_+ be symbols indicating that the i-th element of $\mathbf{R}$ equals to -1 and 1 respectively, we express the ternary vector as a subset:

$$\mathbf{S_R} = \{i_- \mid i \in [d \cdot T_h] \text{ and } \mathbf{R}_i = -1\} \cup \{i_+ \mid i \in [1, d'] \text{ and } \mathbf{R}_i = 1\}.$$

Specifically, when the vector has less than s non-zero entries, we add the following stub symbols to $\mathbf{S_R}$ to ensure there are exact s elements:

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

5.1 Mechanism Design

Existing streamable mechanisms build upon either uniform random selection on dimensions [32, 51] or on non-zero entries [15, 22, 42, 49], then apply sanitation independently on the selected partial data. It simplifies the mechanism design, but suffers an extra $O(d/s)$ or $O(s)$ error factor from optimal mechanisms [40, 53]. On the other hand, in current optimal mechanisms, every non-zero entry or every zero entry is not treated equally (after the local hash is specified), and the private signal for each entry is not determinable until all entries are ready. Our proposed ExSub mechanism is permutation invariant (w.r.t. input/output entries simultaneously) by design, thus every entry can determine the signaling parameter before knowing the remaining entries. The signaling parameter varies with the numbers of observed non-zero and zero entries in input/output by now, therefore enabling improved utility compared with independent sanitation in existing streamable mechanisms.

Let $\mathcal{Z}$ denote the symbol domain $\{1_-, 1_+, ..., d'_-, d'_+\}$, the input data $\mathbf{S_R}$ is then a subset of $\mathcal{Z}$ with exact cardinality s. Operated on the domain of $\mathcal{Z}$, the ExSub mechanism probabilistically outputs a subset $\mathbf{Z} \subseteq \mathcal{Z}$ with a fixed size m. As i_- and i_+ never co-exist in the input, they do not show up together in the output either. We denote the output domain as:

$$\mathcal{Z}^{m\pm} = \{\mathbf{Z} \mid \mathbf{Z} \subseteq \mathcal{Z},\ |\mathbf{Z}| = m \text{ and } |\{i_-, i_+\} \cup \mathbf{Z}| \leq 1\ \forall j \in [d']\}$$

Guided by principles of exponential mechanism [28] and the extremal property [25, 39] for local differential privacy, when 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 $\exp(-\epsilon)$. Here, the closeness criterion between $\mathbf{Z}$ and $\mathbf{S_R}$ is whether $\mathbf{Z}$ has common elements with $\mathbf{S_R}$. That is, we define a binary utility function in the framework of exponential mechanism as follows.

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

The design of the (d', s, ϵ, m)-ExSub mechanism is described in Definition 5.1. The concrete choice of output cardinality m is deferred to Section 5.4.

Definition 5.1 ((d', s, ϵ, m)-ExSub mechanism). For ϵ-LDP ternary vector data sanitization and estimation, take an s-sparse vector $\mathbf{R}$ as input, the ExSub mechanism randomly outputs $\mathbf{Z} \in \mathcal{Z}^{m\pm}$ according to the following probability design:

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

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.

An example. To understand the mechanism, we give an instance where $s = 1$, $\epsilon = \log(2)$, and $m = 2$. Suppose a user holds local residue $\mathbf{R} = [0, -1] \in \{-1, 0, 1\}^2$, then the stubbed domain size is $d' = 3$ and the set representation of $\mathbf{R}$ is $\mathbf{S_R} = \{2_-\}$. In the $(3, 1, \log(2), 2)$-ExSub mechanism, it will select each of the following outputs with probability 1/8 (i.e., when 2_- appears):

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

and select each of the following outputs with probability 1/16:

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

5.2 Offline Implementation

Since the output universe $\mathcal{Z}^{m\pm}$ grows exponentially with d' and m, naively traverse the universe and select one output Z is intractable. In this part, we present an efficient implementation in Algorithm 3 based on uniform sampling without replacement. The underlying idea is dividing the output universe into $\frac{(m+2)(m+1)}{2}$ disjoint subgroups each is indexed by two variables: $num_t \in [0, m]$ and $num_r \in [0, m - num_t]$. The num_t indicates the cardinality of the intersection between $\mathbf{S_R}$ and $\mathbf{Z}$, and the num_r indicates the cardinality of the intersection between $\{i_{-b} \mid i_b \in \mathbf{S_R}\}$ and $\mathbf{Z}$. Each element in the same subgroup has an identical output probability $\frac{e^{-[\![num_t = 0]\!] \cdot \epsilon}}{\Omega}$.

In Algorithm 3, we first select a subgroup (i.e., num_t, num_r) according to the size l of the subgroup and the probability $\frac{e^{-[\![num_t = 0]\!] \cdot \epsilon}}{\Omega}$

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

Input: A ternary vector $\mathbf{R}$ represented in set form $\mathbf{S_R} \in \mathcal{Z}^s$.
Output: A private view $\mathbf{Z} \in \mathcal{Z}^{m\pm}$ that satisfies ϵ-LDP.

```
// Data-independent phase
1  Ω = 2^m C(d', m) + (e^{-ε} − 1) Σ_{m'=0}^{m} 2^{m−m'} C(s, m') C(d−s, m−m')
2  r ← Uniform(0.0, 1.0),  acc ← 0.0
3  for num_t ← 0 to m do
4      for num_r ← 0 to m − num_t do
5          l ← C(s, num_t) C(s−num_t, num_r) C(d'−s, m−num_t−num_r) · 2^{m−num_t−num_r}
6          acc ← acc + 1/(Ω · e^{ε·[[num_t=0]]}) · l
7          if acc ≥ r then break
8      end
9      if acc ≥ r then break
10 end
// Data-dependent phase
11 nonzeros ← UniformSample(S_R, num_t + num_r)
12 trues ← UniformSample(nonzeros, num_t)
13 reverses ← {i_{−b} | i_b ∈ nonzeros\trues}
14 allfalses ← {i_b | j ∈ [d'], b ∈ {−1, 1}, i_b ∉ S_R and i_{−b} ∉ S_R}
15 falses ← UniformSample(allfalses, m − num_t − num_r)
16 Z ← trues ∪ reverses ∪ falses
17 return Z
```

of each output in the subgroup. The overall probability of selecting the subgroup is $l \cdot \frac{e^{-[\![num_t=0]\!]\cdot\epsilon}}{\Omega}$. Second, at line 11, given the total cardinality $num_t + num_r$ of the intersection between $\mathbf{Z}$ and $\mathbf{S_R} \bigcup \{i_{-b} \mid i_b \in \mathbf{S_R}\}$, we uniformly sample $num_t + num_r$ symbols from $\mathbf{S_R}$ as *nonzeros*. Third, we uniformly sample num_t symbols from *nonzeros* to form the intersection $trues = \mathbf{Z} \bigcap \mathbf{S_R}$ at line 12, and uniformly sample num_r symbols from the reversed elements of *nonzeros trues* to form the intersection $reverses = \mathbf{Z} \bigcap \{i_{-b} \mid i_b \in \mathbf{S_R}\}$ at line 13. Finally, we sample $m - num_t - num_r$ symbols from $allfalses = \mathcal{Z}/(\mathbf{S_R} \bigcup \{i_{-b} \mid i_b \in \mathbf{S_R}\})$ to form $falses = \mathbf{Z} \bigcap allfalses$, and ensemble the eventual private view $\mathbf{Z}$. In a nutshell, the last three steps uniformly select one output $\mathbf{Z}$ from the subgroup (num_t, num_r). By the definition of num_t and num_r and the fact that every sampling subroutine is uniformly random, we ensure every output in the subgroup is uniformly sampled.

Each subroutine UniformSample$(Symbols, num)$ samples num elements from a set *Symbols* with known cardinality, thus runs in linear time (e.g., with selection sampling [17, 23] or Reservoir sampling [36]). Selecting a subgroup (num_t, num_r) from $\Theta(m^2)$ subgroups incurs computational costs $O(m^2)$, uniformly sampling $O(m)$ elements from a $O(d')$-size set incurs costs $O(d')$, thus the overall running time is $\Theta(\max(d', m^2)$. Since selecting a subgroup is data-independent, the computation can be delegated to the server or other third parties, and the running time is reduced to $\Theta(d')$.

5.3 Online Implementation

We now present an online implementation of the ExSub with the same functionality as Algorithm 3, except taking $\mathbf{x}_{i,t}$ as streaming input and outputting every $\perp$ or i_b in real-time.

A key observation about ExSub is that selecting the subgroup (num_t, num_r) is independent from input data, and all data-dependent procedures are built upon uniform sampling with known cardinalities. Therefore, before timestamp 1, we pre-compute the data-independent part: selecting a subgroup (num_t, num_r). As the streaming data incomes since timestamp 1, for s non-zero entries, we uniformly sample $num_t + num_r$ of them in an online fashion. Inside the sampling, we uniformly sample num_t elements out of $num_t + num_r$ symbols instantly. At the same time, for $d' - s$ zero entries, we uniformly sample num_f symbols. Note that after (num_t, num_r) is selected, the parameters (i.e., the input/output cardinality) of three uniform samplings are fixed, making streamable sampling possible. Uniform-random sampling a combination from a fixed population in real-time is known as *sequential random sampling* in the literature. We implement these sub-procedures with the famous *selection sampling* or *Algorithm S* [17, 23], which selects each element with probability equal to the number of elements left to sample divided by the number of remaining elements. Alternative approaches that are more efficient can be found in [31].

We present the overall online implementation in Algorithm 4, which runs several (recursive) sequential random samplings. The probability of every final output is the same as the offline one in Algorithm 3, thus it enjoys the same utility guarantee. Furthermore, since the timing of submitting every symbol i_b in Algorithm 4 is the current timestamp j itself, it leaks no extra information thus satisfies the same privacy guarantee as Algorithm 3. The overall computational costs of every user are $O(d \cdot \frac{T}{r^h})$, the memory costs are $O(d + 5) = O(d)$, and the communication costs are $O(m \cdot logd)$.

5.4 Estimating Value and Frequency

We proceed to analyze the behavior of the ExSub mechanism, and give unbiased estimators. Considering a symbol $i_b \in \mathcal{Z}$, there are three cases regarding the input $\mathbf{S_R}$: 1) $i_b \in \mathbf{S_R}$; 2) $i_{-b} \in \mathbf{S_R}$; 3) $i_b \notin \mathbf{R_x}$ *and* $i_{-b} \notin \mathbf{R_x}$. By the design of ExSub mechanism, the probability that i_b appears in the output $\mathbf{Z}$ differs under the three cases. For any $j \in [1 : d']$ and $b \in \{+, -\}$, we define the conditional probabilities as true/false/reverse positive rates as follows.

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

The discrepancy in transition probability enables the server to partially disguise between three cases about input and to derive unbiased estimators of value and frequency in Proposition 5.2 (see Appendix A for proof).

Proposition 5.2 (Unbiased estimators). *Given the private view* $\mathbf{Z}$ *about ternary vector* $\mathbf{R}$ *from* ExSub *mechanism, 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*

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

Input: Streaming ternary vector $\mathbf{Y}_{j,t',h}$ at the hierarchical level h, the vector's overall length $d' = d \cdot T_h + s$.

Output: Streaming output $\mathbf{Z}_{t'} \in \mathcal{Z}^{m\pm}$ that satisfies ϵ-LDP.

Get num_t, num_r as in lines 1-10 of the offline 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 $\text{Uniform}(0.0, 1.0) < \frac{num_f}{num_{rf}}$ **then**
          if $\text{Uniform}(0.0, 1.0) < 0.5$ **then**
            $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_+\}$
          else $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_-\}$
          $num_f \leftarrow num_f - 1$
        end
      else
        $num_{rp} \leftarrow num_{rp} - 1$
        if $\text{Uniform}(0.0, 1.0) < \frac{num_t + num_r}{num_{rp}}$ **then**
          if $\text{Uniform}(0.0, 1.0) < \frac{num_t}{num_t + num_r}$ **then**
            $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_b\}$
            $num_t \leftarrow num_t - 1$
          else
            $\mathbf{Z}_{t'} \leftarrow \mathbf{Z}_{t'} \cup \{i_{-b}\}$
            $num_r \leftarrow num_r - 1$
          end
        end
      end
    end
    $\mathbf{Z}_{t'} = \{(i + t' \cdot d)_{b'} \mid i_{b'} \in \mathbf{Z}_{t'}\}$
    yield $\mathbf{Z}_{t'}$
  end
end

$\widehat{\overline{\mathbf{R}}}_j = \frac{[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]}{p_t - p_r}$; *an unbiased estimator of frequency* $\underline{\mathbf{R}}_j = [\![i_+ \in \mathrm{S_R}]\!] + [\![i_- \in \mathrm{S_R}]\!]$ *is* $\widehat{\underline{\mathbf{R}}}_j = \frac{[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!] - 2p_f}{p_t + p_r - 2p_f}$.

We proceed to analyze the utility guarantee. Based on the error formulation of ExSub mechanism with fixed m (see Appendix B), we further choose an appropriate value $m = \Theta(d'/s)$ in Theorem 5.3 (refer to Appendix B for proof). Therefore, the mean squared error is approximately minimized to $O(\frac{d's}{\epsilon^2})$. The choice of m has a significant impact on the performance (see Figure 1), we set it to $\lceil \frac{d'}{e^\epsilon s + s + 2} \rceil$ empirically.

Theorem 5.3 (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 $\sum_{i=1}^{d'} |\widehat{\overline{\mathbf{R}}}_i - \overline{\mathbf{R}}_i|_2^2 \le O(\frac{d's}{\epsilon^2})$; the (d', s, ϵ, m)-ExSub mechanism with $m = \lceil d'/(e^\epsilon s + 2s + 1) \rceil$ satisfies $\sum_{j=1}^{d'} |\widehat{\underline{\mathbf{R}}}_j - \underline{\mathbf{R}}_j|_2^2 \le O(\frac{d's}{\epsilon^2})$.*

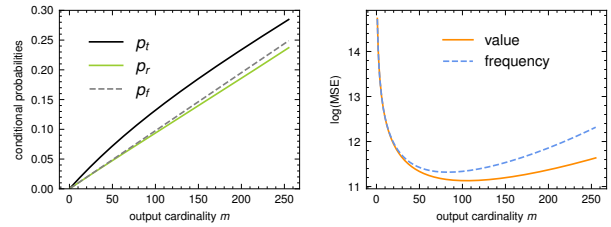

Figure 1: The true/reverse/false positive rates and (natural logarithm) of mean squared error (MSE) on value/frequency of $(d' = 512, s = 4, \epsilon = 0.5, m)$-ExSub mechanism with $n = 1$.

We now derive the expected maximum absolute error of the ExSub for mean estimation, and show it is rate optimal. Based on the (discrete) probability distributions of observed variables $[\![i_+ \in \mathbf{Z}]\!] - [\![i_- \in \mathbf{Z}]\!]$, we present the maximum absolute error bounds of the ExSub mechanism in Theorem 5.4 (see Appendix C for proof). The error is bounded by $\tilde{O}(\frac{1}{\epsilon}\sqrt{\frac{s}{n}})$.

Theorem 5.4 (Maximum Absolute Error of Mean Estimation). *With privacy budget $\epsilon = O(1)$, $m \le d'/s$, and $m = \Theta(d'/s)$, for mean value estimation on n users, the estimator from (d', s, ϵ, m)-ExSub mechanism is bounded as*

$$\max_{i=1}^{d'} |\hat{\underline{\theta}}_i - \underline{\theta}_i| \le O(\sqrt{\frac{s \log(d'/\beta)}{\epsilon^2 n}})$$

with probability $1 - \beta$.

Recently, for the mean estimation of s-sparse ternary vector under ϵ-LDP, works [40, 53] derive the minimax mean squared error bound (i.e., $O(d's/n^2)$) and minimax maximum absolute error bound (i.e., $O(\sqrt{s \log(d'/s)/(\epsilon^2 n)})$) respectively. Combining the upper error bounds in Theorem 5.3 and 5.4, we conclude that the ExSub mechanism is minimax optimal under the measurement of both mean squared error and maximum absolute error. Compared to attainable mechanisms in [40, 53], our ExSub mechanism can be made online, and has 10%-30% empirical improvements (see Section 7). Besides, though the mechanism in [53] is optimal in terms of maximum absolute error, it suffers an extra factor of $\log(n/\beta)$ in the mean squared error and provides only biased estimators.

We note that both the data preparation and the element-wise estimation of value and frequency are streamable, thus the whole ExSub mechanism can be implemented online.

6 PRIVACY AMPLIFICATION WITH ONLINE SHUFFLING

Despite the advantage of minimum trust in other parties, the local privacy is criticized for injecting large amount of noises. For example, the locally private ternary vector mean estimation invokes worst-case error $O(d's/(n\epsilon^2))$ inevitably. The shuffle model [3, 15] amplifies local privacy via anonymity, and thus strides over the error barrier. In this part, we give privacy amplification bound of

locally private analytics, and show online shuffling has strictly stronger amplification effects than the classical one.

Online Shuffling. In the classical offline shuffle model in Section 3.1, every private view from other users is shuffled as a whole. As contrast, in online data analytics, the server is supposed to receive responses in real time, thus the shuffler has to shuffle partial private views $[\mathbf{z}_{1,t}, ..., \mathbf{z}_{n,t}]$ at every timestamp t. We here take a further step and propose to shuffle every symbol of $\mathbf{z}_{j,t}$ independently. We refer to it as online shuffling.

Collaborative Clones. Recall the clone method for the classical shuffle model in Section 3.2, every ϵ-private view from other users is a clone of user 1 with probability $e^{-\epsilon}$. Since ExSub mechanism is ϵ-private, the amplified privacy is at most the bound in Lemma 3.4. Besides, notice that in the online shuffling, all symbols (i.e., i_b) from one user is no longer linked. The break of linkage of multiple symbols ensures better anonymity, and provides an opportunity for more clones. Considering two users $j, j+1$ employing the (d', s, ϵ, m)-ExSub mechanism with output size $m = 2$, let $[i_b, i'_{b'}]$ denote the 2 symbols outputted from $K(x_1)$. When either $K(x_j)$ or $K(x_{j+1})$ outputs the joint-message $[i_b, i'_{b'}]$ (i.e., they are not the clone of $K(a)$), the $K(x_j)$ may outputs i_b (or $i'_{b'}$) and the $K(x_{j+1})$ may outputs $i'_{b'}$ (or i_b) simultaneously, and they collaboratively provide $[i_b, i'_{b'}]$ (i.e, a clone of $K(a)$).

Formally, we let $\mathcal{Z}^{m'}_{c,c'}$ denote the domain of multiset on $\mathcal{Z}$ with cardinality m', maximum count $\max_{j\in[d'],b\in\{-1,1\}} \#i_b \leq c$, and $\max_{j\in[d']} \#i_+ + \#i_- \leq c'$. Let $\tilde{S}$ denote the process of online shuffling, then one output $\tilde{S}(K(x))$ from $(d', s, \epsilon, 2)$-ExSub mechanism belongs to $\mathcal{Z}^2_{1,1}$ and two online-shuffled outputs $\tilde{S} \circ K(x_j, x_{j+1})$ belong to $\mathcal{Z}^4_{2,2}$. The following Lemma implies that $\tilde{S} \circ K(x_j, x_{j+1})$ is one or two clones of $K(x_1)$. See Appendix D for proof.

Lemma 6.1 (Clones from two online shuffled $(d', s, \epsilon, 2)$-ExSub messages). *For any $x_1, x'_1 \in \{a, b\} \subseteq \mathcal{X}$ and any $x_j, x_{j+1} \in \mathcal{X}$, the distribution of $S(K(x_j), K(x_{j+1}))$ is a mixture:*

$$\tilde{S}\circ K(x_j, x_{j+1}) = \begin{cases} \tilde{S} \circ K(x_1, x'_1), & \text{with prob. } \frac{1}{e^{2\epsilon}}; \\ \tilde{S}(K(x_1), \mathcal{M}(x_j, x_{j+1})), & \text{with prob. } \frac{2(1-e^{\epsilon})}{e^{\epsilon}} + p_{cc}; \\ \mathcal{W}(x_j, x_{j+1}), & \text{else,} \end{cases}$$

where the collaborative clone probability is:

$$p_{cc} = \frac{d'(d'-1)}{e^{\epsilon}} \left(\frac{s + (2d' - s - e^{\epsilon} - 2)(e^{-\epsilon} - e^{-2\epsilon})}{\Omega}\right)^2.$$

Enlarged Number of Clones. We now analyze the total number of $K(x_1)$ clones coming from $\tilde{S}(K(x_2), ..., K(x_n))$, which is equivalent to $\tilde{S}(\tilde{S} \circ K(x_2, x_3), ..., \tilde{S} \circ K(x_{n-1}, x_n))$ due to behavior of uniform-random permutation. For facilitating the tail bounding of clones, we define a distribution $Paired(\epsilon)$ over $\{0, 1, 2\}$ as follows:

$$Paired(\epsilon) = \begin{cases} 2, & \text{with probability } e^{-2\epsilon}; \\ 1, & \text{with probability } 2(1 - e^{-\epsilon})/e^{\epsilon} + p_{cc}; \\ 0, & \text{else.} \end{cases}$$

According to the clone count from every two online-shuffled messages $\tilde{S}(K(x_j), K(x_{j+1}))$ in Lemma 6.1, the total number of clones is the summation of $(n-1)/2$ samples from $Paired(\epsilon)$. Combining the following Theorem 6.2 (see Appendix E for proof) and Lemma 3.4, we conclude that the $\tilde{S}(K(x_2), ..., K(x_n))$ satisfies (ϵ'_c, δ)-DP. Compared to the classical shuffling, the online shuffling has a strictly larger $\delta/2$-bound on number of clones and has thus strictly stronger amplification effects (i.e., $\epsilon'_c < \epsilon_c$ from Section 3.2).

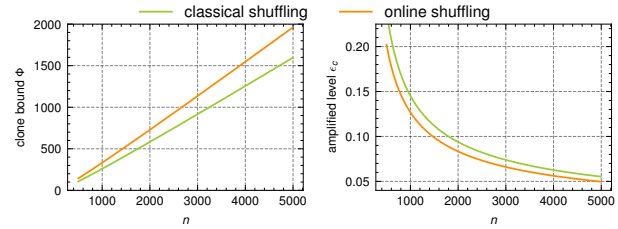

Figure 2: Comparison of privacy amplification with classical shuffling and online shuffling for n users adopting $(d' = 1000, s = 20, \epsilon = 1, m = 2)$-ExSub mechanism.

Theorem 6.2 (Privacy Amplification of Online Shuffled $(d', s, \epsilon, m = 2)$-ExSub Messages). *Let $q \in [0.5, 1]$ denote the mutual-clone probability, consider the process that we first sample $(n-1)/2$ variables $c_{(i)} \sim Paired(\epsilon)$, and let $C = \sum_{i=1}^{(n-1)/2} c_{(i)}$. Then, we sample $A \sim Binomial(C, \frac{1}{2})$, and sample $B \sim Bernoulli(q)$. When $n \geq 16e^{\epsilon}\log(4/\delta)$ and $\epsilon \leq \log(3)$, the random variables $P = (A + B, C - A + 1 - B)$ and $Q = (A + 1 - B, C - A + B)$ are (ϵ'_c, δ)-indistinguishable with:*

$$\epsilon'_c = \log\left(1 + \frac{e^{\epsilon} - 1}{e^{\epsilon} + 1} \cdot \frac{2\sqrt{\log(4/\delta)\Phi'/2} + 1}{\Phi'/2 - \sqrt{\log(4/\delta)\Phi'/2}}\right),$$

where $\Phi' = \frac{n}{e^{\epsilon}} + \frac{nd'(d'-1)}{2e^{\epsilon}}\left(\frac{s+(2d'-s-e^{\epsilon}-2)(e^{-\epsilon}-e^{-2\epsilon})}{\Omega}\right)^2 - \sqrt{\frac{3n\log(4/\delta)}{e^{\epsilon}}}$.

Since $\Omega \leq 4d'(d'-1)$, we have the collaborative clone probability $p_{cc} \geq \frac{(2d'-e^{\epsilon}-2)^2}{8d'(d'-1)} \frac{(1-e^{-\epsilon})^2}{e^{3\epsilon}} \approx \frac{(1-e^{-\epsilon})^2}{2e^{3\epsilon}}$ when d' is large. This implies the p_{cc} is almost domain size independent thus fits all levels of the hierarchical-tree based online aggregation. To illustrate the effect of online shuffling, we plot the tail bound and the amplified privacy level ϵ_c (w.r.t. $\delta = 10^{-5}$) in Figure 2, with comparison to the classical one in Lemma 3.4. It is observed that online shuffling offers more than 20% effective clones and saves about 10% privacy budget.

7 EXPERIMENTAL EVALUATION

Our experimental evaluation focuses on the utility performance of locally private protocols. We start assessment in offline settings to comprehensively compare the ExSub with existing approaches, then evaluate them for online mean and range queries. Competing offline approaches include the PrivKV mechanism [51], the KVUE mechanism [32], the PCKV mechanism with generalized randomized response as the base randomizer (denoted as PCKV-GRR), the PCKV mechanism with unary encoding as the base randomizer[22] (denoted as PCKV-UE), the Collision mechanism [40], and the succinct mean estimation protocol [53] (denoted as SUCCINCT). Competing online protocols include the hierarchical-tree-based approach from Erlingsson *et al.* [15] (denoted as EFMRTT), the ToPL approach [42], the DDRM mechanism [49], and other streamable mechanisms originally designed for offline settings: PCKV-GRR and PCKV-UE. We are noted that several online protocols [11, 16, 24] adopted much weaker privacy protection on streaming data (see Section 2), thus are beyond comparison. The FutureRand mechanism [29] is claimed to be asymptotically optimal in theoretic, but the actual estimation

errors are an order more than other approaches' (See Table 2 and more results in Appendix F).

7.1 Datasets & Metrics

We use two real-world datasets: Stock [1] and Trajectory [2]. We also synthesize datasets with diverse parameters to cover more practical scenarios. The Stock dataset contains historical daily prices of 7136 U.S. stocks from 2014 to 2017. We split the dataset into 160000 records each containing close prices in 32 consecutive trading days, and preprocess the prices into ternary values $\{-1, 0, 1\}$ indicating whether the accumulated drop or rise since last recorded change (e.g., $-1, 1$ in the stream) are over 2%. The Trajectory dataset records 442 taxi trajectories running in the city of Porto, in Portugal. We cut the rectangular area $[-8.65, -8.55] \times [41.1, 41.2]$ (w.r.t. longitude and latitude) to 3×4 cells, and split the dataset into 1044693 trajectories each contains 32 celled locations in the area, similarly as [49]. In the synthesized dataset, the ternary residue vector of each user is independent-randomly generated: s non-zeros entries are uniform-randomly selected from $d \cdot T$ entries.

We use total variation error and maximum absolute error as utility metrics. For mean queries, the total variation error is:

$$\text{TVE} = \sum\nolimits_{t \in [T]} |\widehat{\overline{\mathbf{x}}}_{*,t} - \overline{\mathbf{x}}_{*,t}|_1,$$

where the true mean value at the timestamp t is $\overline{\mathbf{x}}_{*,t}$ is $\sum_{j=1}^{n} \mathbf{x}_{j,t}/n$; the maximum absolute error is:

$$\text{MAE} = \max\nolimits_{t \in [T]} |\widehat{\overline{\mathbf{x}}}_{*,t} - \overline{\mathbf{x}}_{*,t}|_{+\infty}.$$

The above metrics also apply to the range queries and intermediate residue statistics. Every reported result is the average of 100 independent simulations. For offline queries, we post-process the intermediate residue estimators $\hat{\overline{\mathbf{R}}}_*$ and $\hat{\underline{\mathbf{R}}}_*$ from all protocols by projecting them to a capped $\Delta_{d'}$ simplex [44]; one exception is the SUCCINCT protocol [53] that is only able to estimate the mean residue value $\hat{\overline{\mathbf{R}}}_*$, for which we post-process by truncating each mean residue value to $[-1, 1]$. In typical settings, we present experimental results both without and with post-processing.

7.2 Offline Queries

We start with offline experiments under extensive synthesized settings, and study the effects of various parameters on the performance of locally private mechanisms.

Effects of dimensions $d \cdot T$. Simulated with $n = 10000$, $s = 8$ and overall dimension $d' = d \cdot T$ ranging from 64 to 512, the error results on mean residue value are presented in Figure 3. We observe that the ExSub dominates existing approaches in every setting. When the domain size is relatively small (e.g., $d' = 64$), the PCKV-GRR has close performances to optimal mechanisms (i.e., Collision/SUCCINT/ExSub), while the gap grows large as the domain size increases. Similar phenomena also exist for PrivKV/KVUE, validated our theoretical analyses indicating the PCKV-GRR/PrivKV/KVUE has sub-optimal dependence on the domain size.

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

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

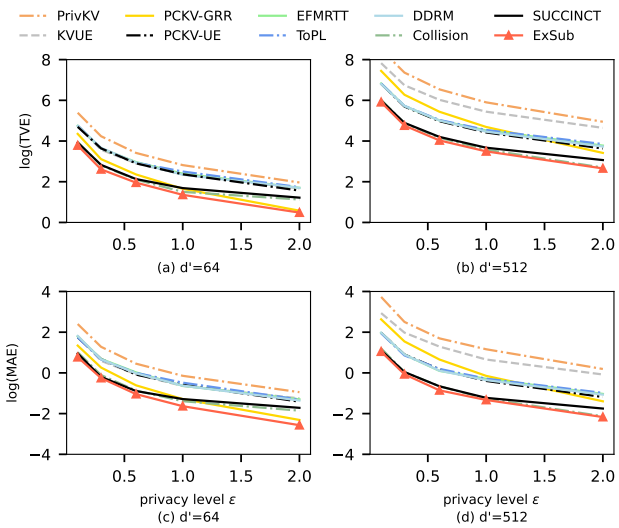

Figure 3: Error results without post-processing with $n = 10000$, $m = 8$ and dimension parameter d' varies from 64 to 512.

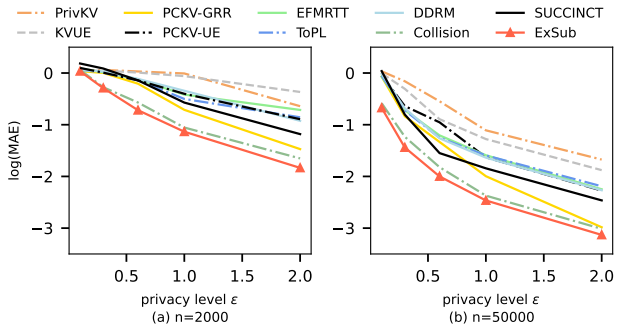

Figure 4: MAE results with post-processing with $d' = 128$, $m = 8$ and number of users n varies from 2000 to 50000.

Effects of users n. Simulated with $d' = 128$ and $s = 8$, we vary the number of users from 2000 to 50000, and present the results on mean residue value in Figure 4. The ExSub unanimously beats existing approaches. When the user population gets larger (i.e., the effect of post-processing decreases), the performance gap becomes more significant. The SUCCINT protocol uses clip operation to truncate extreme values, thus having excellent performance when estimators are not post-processed (e.g., in Figure 3), but the performance after post-processing is not satisfiable.

Effects of sparsity s. Simulated with $n = 10000$ and $d' = 256$, we vary the sparsity parameter s from 1 to 64, and plot the results on mean residue value in Figure 5. As the sparsity parameter gets larger, the performance gap between PCKV-GRR/PrivKV/KVUE and ExSub gets smaller, and the performance gap between PCKV-UE/EFMRTT/ToPL/DDRM and ExSub gets larger. This confirms our theoretical analyses indicating PCKV-UE/EFMRTT/ToPL/DDRM suffers sub-optimal dependence on the sparsity parameter.

Results under extreme privacy budgets. Simulated with $n = 10000$, $d' = 128$, and $s = 8$, we list the error results of optimal mechanisms in Table 2 under extremely small or large privacy budgets, which might be interesting for evaluating the performance

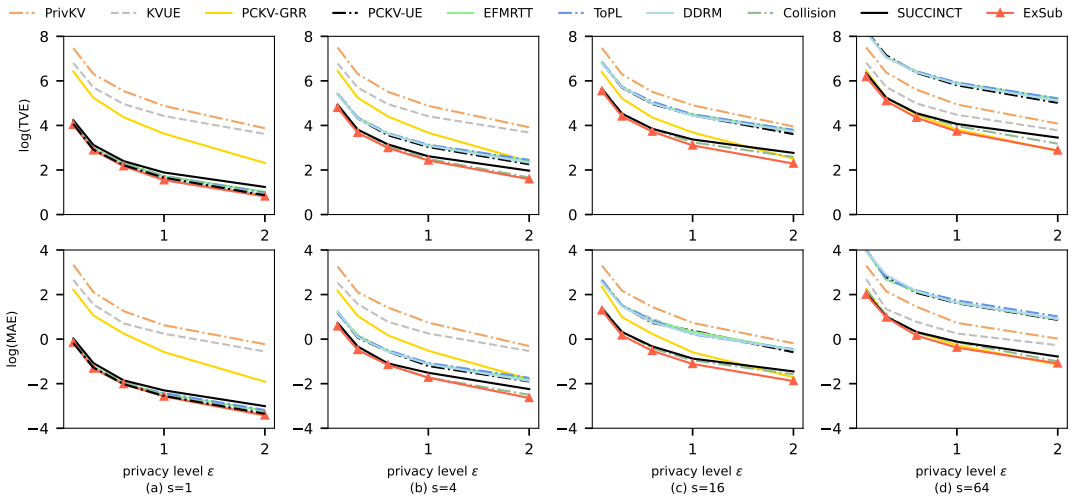

Figure 5: Error results without post-processing with $n = 10000$, $d' = 256$ and sparsity parameter s varies from 1 to 64.

Table 2: 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 [40]	4.6e+3	463.7	4.13	1.15	0.54
SUCCINCT [53]	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 [40]	128.1	12.6	0.11	0.034	0.021
SUCCINCT [53]	131.7	13.7	0.14	0.077	0.077
ExSub	**113.7**	**10.5**	**0.094**	**0.026**	**0.014**

in asymptotic settings and in the shuffle model. It is observed that the ExSub has a minimum constant factor on error bounds in every settings, and outperforms existing optimal mechanisms by 10%-30%.

Effects of fan out parameter r. Simulated 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 3 presents the error results of the ExSub and the DDRM mechanism for mean queries. The fan-out parameter $r = 4$ performs slightly better than $r = 2$, but the gap is negligible. Since fan out with $r > 2$ causes large factors on the error bounds for range queries (see Section 4), we suggest using $r = 2$.

7.3 Online Mean Queries

This part dedicates to evaluate the performance under online settings with real-world datasets.

On Stock dataset. We truncate the number of significant changes in stock prices to 6 (i.e., $s = 6$). Since the hierarchical residue tree is not applicable to the integer-valued domain here, we directly use ternary privatization mechanisms on the $\{-1, 0, 1\}$ rise/drop record with 32 timestamps. We present the results in Figures 6 and 12. The ExSub outperforms existing approaches by about 30%.

Table 3: MAE Results on mean queries without post-processing under vary fan-out parameter ($n = 50000$, $d = 1$, $T = 128$, $s = 8$).

	$\epsilon = 0.1$			
	$r = 2$	$\mathbf{r = 4}$	$r = 8$	$r = 16$
DDRM [49]	13.7	**13.5**	14.6	14.8
ExSub	4.85	**4.83**	5.06	6.06
	$\epsilon = 1.0$			
	$r = 2$	$r = 4$	$r = 8$	$r = 16$
DDRM [49]	13.7	**13.5**	14.6	14.8
ExSub	0.44	**0.43**	0.48	0.45

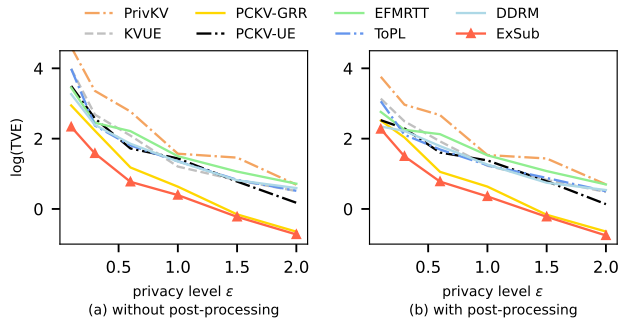

Figure 6: TVE results for mean queries on the Stock dataset.

On Trajectory dataset. We truncate the number of changes in celled locations to 4 (i.e., $s = 2 \cdot 4$), and use fan-out parameter $r = 2$ for continual location distribution estimation. We present the error results in Figure 7. The ExSub reduces about 50% when compared to existing approaches.

7.4 Online Range Queries

In this part, we evaluate the performance for range queries and demonstrate the necessity of calibrated portion strategies from Section 4. Since the hierarchical structure is not applicable to the

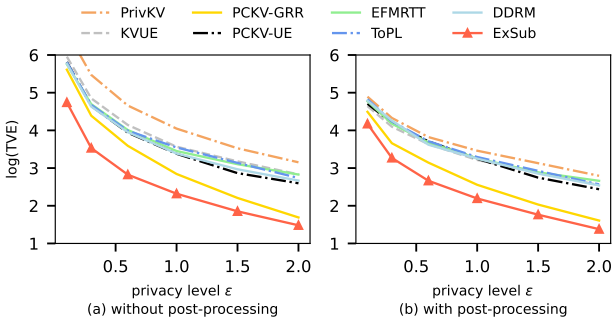

Figure 7: TVE results for mean queries on Trajectory dataset.

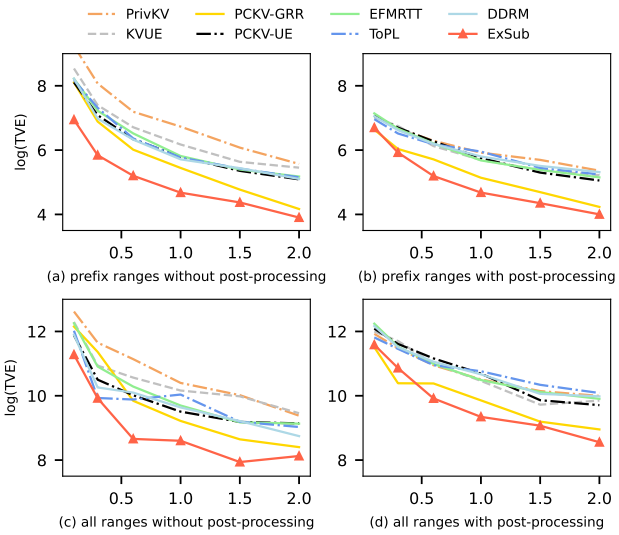

Figure 8: TVE results on range queries of Trajectory dataset.

Stock dataset, we focus on the range query over the Trajectory dataset, such as aggregating location heatmap over one day.

We present the results with uniform portions $\mathbf{W}_h = 1/H$ in Figure 8. The ExSub mechanism outperforms other approaches by about 50%. We also experiment with calibrated portions, which are deduced by a more sophisticated analysis on the weight of each level h, to avoid overestimating the 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 $(\sum_{t'=0}^{(r-1)r^h} t'^2) + (\sum_{t'=(r-1)r^h}^{T-r^h} (r-1)^2 r^{2h} = (T - r^{h+1} + \frac{(r-1)r^h((r-1)r^h+1)(2(r-1)r^h+1)}{6(r-1)^2 r^{2h}}) r^{2h}$, thus letting $\mathbf{W}_h = (T - r^{h+1} + \frac{(r-1)r^h((r-1)r^h+1)(2(r-1)r^h+1)}{6(r-1)^2 r^{2h}}) r^h$ approximately minimizes error. The results with this portion strategy are marked as ExSub-*calibrated* in Figure 9 for trajectory dataset and in Figure 10 for a synthetic dataset with relatively long sequence (i.e., $T = 128$). When compared with the uniform portion strategy (denoted as ExSub-*uniform*), we observe that the calibrated strategy reduces about 20% error for prefix/all range queries, meanwhile incurring slightly more error for mean queries. This indicates the portion strategy must calibrate to the task at hand.

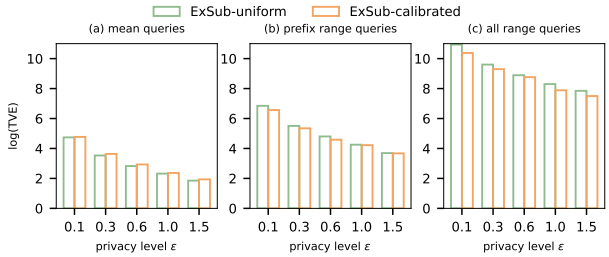

Figure 9: TVE results comparison on Trajectory dataset.

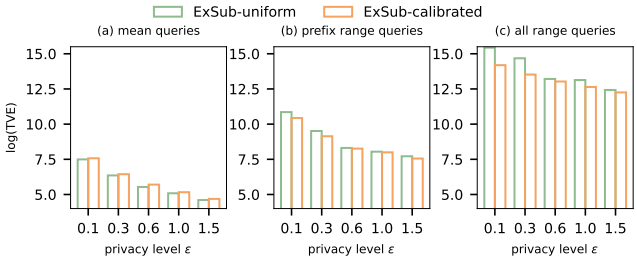

Figure 10: TVE results comparison on a synthetic dataset with $n = 100000$, $d = 8$, $t = 128$, and $m = 20$.

7.5 Experimental Summary

In brief, the ExSub mechanism shows premium utility performances in both offline and online data analytics. It reduces about 10%-30% estimation error even when compared to existing optimal offline mechanisms (i.e., Collision [40], SUCCINCT [53]), and reduces more than 40% error when compared to existing online mechanisms. We believe it is a theoretically-/empirically-guaranteed replacement for (almost) all settings.

8 CONCLUSION AND DISCUSSION

We presented a general and optimal protocol SaO_2 for online locally private data collecting and analyses, covering rich types of user data (i.e., multi-dimensional binary vectors) and various statistical queries (e.g., mean estimation, range queries). Our core mechanism exploits the mutually exclusive relation for matching error lower bounds of the underlying ternary vector aggregation problem, and only uses primitives that allow online computation and responding (i.e., uniform sampling with fixed population size). We also revealed an interesting phenomenon in the shuffle model: online shuffling offers strictly stronger privacy amplification effects than its offline version. Through extensive experiments, we demonstrated that the proposals beat existing approaches by about 50% in typical settings.

Discussion on the gap between offline and online private estimations. In contrast to enormous theoretical error characterizations for offline private analytics (e.g., locally private minimax error bounds in [13]), there were no such results for their online version in the literature. This work provided some optimistic instances on the fundamental utility gap between offline and online analytics, and showed *the gap is zero for a broad class of problems* (as ternary vector compromises categorical and set-valued data). It is promising to study other common problems (e.g., numerical streams with bounded ℓ_2-norm).

REFERENCES

[1] Borja Balle, James Bell, Adria Gascón, and Kobbi Nissim. 2019. The privacy blanket of the shuffle model. *CRYPTO* (2019).

[2] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. 2020. Private summation in the multi-message shuffle model. *CCS* (2020).

[3] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. 2017. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th symposium on operating systems principles*. 441–459.

[4] Albert Cheu. 2021. Differential privacy in the shuffle model: A survey of separations. *arXiv preprint arXiv:2107.11839* (2021).

[5] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. 2019. Distributed differential privacy via shuffling. *EUROCRYPT* (2019).

[6] Albert Cheu and Maxim Zhilyaev. 2022. Differentially private histograms in the shuffle model from fake users. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 440–457.

[7] Chi-Yin Chow and Mohamed F Mokbel. 2011. Trajectory privacy in location-based services and data publication. *ACM Sigkdd Explorations Newsletter* 13, 1 (2011), 19–29.

[8] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. 2019. Answering range queries under local differential privacy. *VLDB* (2019).

[9] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. 2019. Answering range queries under local differential privacy. *Proceedings of the VLDB Endowment* 12, 10 (2019), 1126–1138.

[10] Rogier Creemers and Graham Webster. 2021. Translation: Personal Information Protection Law of the People's Republic of China—Effective Nov. 1, 2021. *DigiChina Project, August* 20 (2021).

[11] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting telemetry data privately. *Advances in Neural Information Processing Systems* 30 (2017).

[12] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2013. Local privacy and statistical minimax rates. *FOCS* (2013).

[13] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2018. Minimax optimal procedures for locally private estimation. *J. Amer. Statist. Assoc.* (2018).

[14] Cynthia Dwork. 2008. Differential privacy: A survey of results. *International Conference on Theory and Applications of Models of Computation* (2008), 1–19.

[15] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. Amplification by shuffling: From local to central differential privacy via anonymity. *SODA* (2019).

[16] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. *CCS* (2014).

[17] CT Fan, Mervin E Muller, and Ivan Rezucha. 1962. Development of sampling plans by using sequential (item by item) selection techniques and digital computers. *J. Amer. Statist. Assoc.* 57, 298 (1962), 387–402.

[18] Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2022. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 954–964.

[19] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. 2021. On the power of multiple anonymous messages: Frequency estimation and selection in the shuffle model of differential privacy. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 463–488.

[20] Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. 2020. Private aggregation from fewer anonymous messages. *EUROCRYPT* (2020).

[21] Eric Goldman. 2020. An introduction to the california consumer privacy act (CCPA). *Santa Clara Univ. Legal Studies Research Paper* (2020).

[22] Xiaolan Gu, Ming Li, Yueqiang Cheng, Li Xiong, and Yang Cao. 2020. PCKV: Locally Differentially Private Correlated Key-Value Data Collection with Optimized Utility. *USENIX Security* (2020).

[23] Terence G Jones. 1962. A note on sampling a tape-file. *Commun. ACM* 5, 6 (1962), 343.

[24] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. 2018. Local differential privacy for evolving data. *Advances in Neural Information Processing Systems* 31 (2018).

[25] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2014. Extremal mechanisms for local differential privacy. *Advances in neural information processing systems* 27 (2014).

[26] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[27] Tejas Kulkarni. 2019. Answering range queries under local differential privacy. *SIGMOD* (2019).

[28] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. *FOCS* (2007).

[29] Olga Ohrimenko, Anthony Wirth, and Hao Wu. 2022. 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*. 237–249.

[30] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2016. Heavy hitter estimation over set-valued data with local differential privacy. *CCS* (2016).

[31] Michael Shekelyan and Graham Cormode. 2021. Sequential Random Sampling Revisited: Hidden Shuffle Method. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3628–3636.

[32] Lin Sun, Jun Zhao, Xiaojun Ye, Shuo Feng, Teng Wang, and Tao Bai. 2019. Conditional Analysis for Key-Value Data with Local Differential Privacy. *arXiv preprint arXiv:1907.05014* (2019).

[33] Abhradeep Guha Thakurta, Andrew H Vyrros, Umesh S Vaishampayan, Gaurav Kapoor, Julien Freudiger, Vivek Rangarajan Sridhar, and Doug Davidson. 2017. Learning new words. (March 14 2017). US Patent 9,594,741.

[34] Abhradeep Guha Thakurta, Andrew H Vyrros, Umesh S Vaishampayan, Gaurav Kapoor, Julien Freudinger, Vipul Ved Prakash, Arnaud Legendre, and Steven Duplinsky. 2017. Emoji frequency detection and deep link frequency. (July 11 2017). US Patent 9,705,908.

[35] James Victor Uspensky. 1937. *Introduction to mathematical probability*. McGraw-Hill Book Company, New York.

[36] Jeffrey S Vitter. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)* 11, 1 (1985), 37–57.

[37] Paul Voigt and Axel Von dem Bussche. 2017. *The EU General Data Protection Regulation (GDPR)*. Vol. 18. Springer.

[38] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 2019. Collecting and analyzing multidimensional data with local differential privacy. *ICDE* (2019).

[39] Shaowei Wang, Liusheng Huang, Yiwen Nie, Pengzhan Wang, Hongli Xu, and Wei Yang. 2018. PrivSet: Set-Valued Data Analyses with Locale Differential Privacy. *INFOCOM* (2018).

[40] Shaowei Wang, Jin Li, Yuqiu Qian, Jiachun Du, Wenqing Lin, and Wei Yang. 2021. Hiding Numerical Vectors in Local Private and Shuffled Messages.. In *IJCAI*. 3706–3712.

[41] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*. 729–745.

[42] Tianhao Wang, Joann Qiongna Chen, Zhikun Zhang, Dong Su, Yueqiang Cheng, Zhou Li, Ninghui Li, and Somesh Jha. 2021. Continuous release of data streams under both centralized and local differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1237–1253.

[43] Tianhao Wang, Bolin Ding, Jingren Zhou, Cheng Hong, Zhicong Huang, Ninghui Li, and Somesh Jha. 2019. Answering multi-dimensional analytical queries under local differential privacy. In *Proceedings of the 2019 International Conference on Management of Data*. 159–176.

[44] Weiran Wang and Canyi Lu. 2015. Projection onto the capped simplex. *arXiv preprint arXiv:1503.01002* (2015).

[45] Yichuan Wang, LeeAnn Kung, and Terry Anthony Byrd. 2018. Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological forecasting and social change* 126 (2018), 3–13.

[46] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.

[47] Yonghui Xiao and Li Xiong. 2015. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1298–1309.

[48] Xingxing Xiong, Shubo Liu, Dan Li, Zhaohui Cai, and Xiaoguang Niu. 2020. A comprehensive survey on local differential privacy. *Security and Communication Networks* 2020 (2020).

[49] Qiao Xue, Qingqing Ye, Haibo Hu, Youwen Zhu, and Jian Wang. 2022. DDRM: A Continual Frequency Estimation Mechanism with Local Differential Privacy. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[50] Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam. 2020. Local differential privacy and its applications: A comprehensive survey. *arXiv preprint arXiv:2008.03686* (2020).

[51] Qingqing Ye, Haibo Hu, Xiaofeng Meng, and Huadi Zheng. 2019. PrivKV: Key-Value Data Collection with Local Differential Privacy. *IEEE S&P* (2019).

[52] Qingqing Ye, Haibo Hu, Xiaofeng Meng, Huadi Zheng, Kai Huang, Chengfang Fang, and Jie Shi. 2021. PrivKVM*: Revisiting Key-Value Statistics Estimation with Local Differential Privacy. *IEEE Transactions on Dependable and Secure Computing* (2021).

[53] Mingxun Zhou, Tianhao Wang, TH Hubert Chan, Giulia Fanti, and Elaine Shi. 2022. Locally Differentially Private Sparse Vector Aggregation. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 1565–1565.

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 $\frac{\mathbb{E}[[\![i_+ \in \mathbf{Z}]\!] + [\![i_- \in \mathbf{Z}]\!]] - 2p_f}{p_t + p_r - 2p_f} \equiv \underline{\mathbf{R}}_j$.

B 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 B.1 as a formula of true/false/reverse positive rates.

Lemma B.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 5.3 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 B.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 B.2 and B.3, and arrive that $2C_2d' \geq C_1(2d' - ms)$ 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 B.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 $(2^m\binom{d'}{m} - \sum_{m'=0}^{m} 2^{m'}\binom{s}{m-m'}\binom{d-s}{m-m'}) \leq s2^{m-1}\binom{d'-1}{m-1}$.

Assume 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 B.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 $2^{m'} \binom{s}{m-m'}\binom{d-s}{m'} = \frac{s}{m-m'} 2^{m'} \binom{s-1}{m-1-m'}\binom{d-s}{m-m'} \leq s 2^{m'} \binom{s-1}{m-1-m'}\binom{d-s}{m'}$. 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 B.2 and B.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:

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

C PROOF OF MAXIMUM ABSOLUTE ERROR BOUNDS OF EXSUB MECHANISM

Considering the i-th mean value $\underline{\theta}_i$, according to Lemma B.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 [35], 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.

D PROOF OF COLLABORATIVE CLONES FROM TWO ONLINE SHUFFLED MESSAGES

According to classical clone technique, the $K(x_j)$ or $K(x_{j+1})$ is a clone of $K(x)$ with probability $e^{-\epsilon}$:

$$K(x_j) = \begin{cases} K(x), & \text{with probability } e^{-\epsilon}; \\ \mathcal{W}_x(x_j), & \text{else.} \end{cases}$$

Applying the definition of (d', s, ϵ, m)-ExSub, the distribution of $\mathcal{W}_x(x_j)$ is

$$\mathbb{P}[\mathcal{W}_x(x_j) = z] = \frac{\exp(-[\![z \bigcap x_j = \phi]\!]) - \exp(-[\![z \bigcap x = \phi]\!] - 1)}{(1-e^{-\epsilon})\Omega}. \tag{3}$$

Given the fact that $\mathcal{W}_x(x_j)$ and $\mathcal{W}_x(x_{j+1})$ appear together with probability $(1-e^{-\epsilon})^2$, to reach the proposition, it is enough to prove that $\tilde{S} \circ \mathcal{W}(x_j, x_{j+1})$ is a clone of $K(x)$ with probability $\frac{p_{cc}}{(1-e^{-\epsilon})^2}$. To this end, we first show $\tilde{S} \circ \mathcal{W}(x_j, x_{j+1})$ is a clone of uniform distribution over the domain $\mathcal{Z}_{1,1}^2$ with probability $\frac{e^{\epsilon}p_{cc}}{(1-e^{-\epsilon})^2}$:

$$\tilde{S} \circ \mathcal{W}(x_j, x_{j+1}) = \begin{cases} \tilde{S}(\mathrm{uniform}(\mathcal{Z}_{1,1}^2), \mathcal{M}'(x_j, x_{j+1})), & \text{with prob. } \frac{e^{\epsilon}p_{cc}}{(1-e^{-\epsilon})^2}; \\ \mathcal{W}'(x_j, x_{j+1}), & \text{else.} \end{cases}$$

For any possible output $\{i_b, i'_{b'}\}$ from uniform$(\mathcal{Z}^2_{1,1})$, one can construct it by concatenate two outputs $\{j^*_{b^*}, i_b\}$ and $\{j^{*'}_{b^{*'}}, i'_{b'}\}$ from $\mathcal{W}(x_j)$ and $\mathcal{W}(x_{j+1})$ respectively. Thus we can redistribute the probabilities from domain $\mathcal{Z}^2_{1,1} \times \mathcal{Z}^2_{1,1}$ to domain $\mathcal{Z}^2_{1,1}$:

$$\begin{aligned}&\mathbb{P}[\{i_b, i'_{b'}\} \subseteq \tilde{S}(\mathcal{W}(x_j), \mathcal{W}(x_{j+1}))]\\ \geq\ &\mathbb{P}[i_b \in \mathcal{W}(x_j)]\mathbb{P}[i'_{b'} \in \mathcal{W}(x_{j+1})] + \mathbb{P}[i'_{b'} \in \mathcal{W}(x_j)]\mathbb{P}[i_b \in \mathcal{W}(x_{j+1})]\end{aligned}$$

Based on the formula of $\mathcal{W}(x_j)$ in Equation 3, we present a lower bound on the probability that i_b showing up in $\mathcal{W}(x_j)$ as follows.

Lemma D.1. *Given the the definition of $\mathcal{W}(x_j)$ in previous paragraphs, for any $i_b \in \mathcal{Z}$ and any $x_j \in \mathcal{X}$, we have:*

$$\mathbb{P}[i_b \in \mathcal{W}(x_j)] \geq \frac{s(1 + e^{-2\epsilon} - e^{-\epsilon}) + (2d' - 2 - e^{\epsilon})(e^{-\epsilon} - e^{-2\epsilon})}{(1 - e^{-\epsilon})\Omega}.$$

Proof. We separately consider three cases: $[\![i_+ \in \mathbf{Y}_{\mathbf{x_j}}]\!] = 1$, $[\![i_- \in \mathbf{Y}_{\mathbf{x_j}}]\!] = 1$, $[\![i_+ \in \mathbf{Y}_{\mathbf{x_j}}]\!] = 0$ *and* $[\![i_- \in \mathbf{Y}_{\mathbf{x_j}}]\!] = 0$ and and let k denote the number of output $\{j^*_{b^*}, i_b\}$ that has common elements with both $\mathbf{Y_x}$ and $\mathbf{Y_{x_j}}$. For the first case, observe that there are total $2d' - s$ outputs $\{j^*_{b^*}, i_b\}$ that have common elements with $\mathbf{Y_{x_j}}$, then the $(1 - e^{-\epsilon})\Omega \cdot \mathbb{P}[i_b \in \mathcal{W}(x_j)]$ equals to $k(1 - e^{-\epsilon}) + (2d' - s - k)(1 - e^{-2\epsilon})$ that is not lower than $(2d' - s)(1 - e^{-\epsilon})$. For the second case, there are total $s - 1$ outputs $\{j^*_{b^*}, i_b\}$ that have common elements with $\mathbf{Y_{x_j}}$, thus $(1-e^{-\epsilon})\Omega\cdot\mathbb{P}_i[i_b \in \mathcal{W}(x_j)]$ equals to $k(1-e^{-\epsilon})+(s-1-k)(1-e^{-2\epsilon})+(2d'-s-1-(s-k))(e^{-\epsilon}-e^{-2\epsilon})$ that is not lower than $(s-1)(1-e^{-\epsilon})+(2d'-s-1)(e^{-\epsilon}-e^{-2\epsilon})$. For the last case, there are total s outputs $\{j^*_{b^*}, i_b\}$ that have common elements with $\mathbf{Y_{x_j}}$, thus $(1-e^{-\epsilon})\Omega\cdot\mathbb{P}_i[i_b \in \mathcal{W}(x_j)]$ equals to $k(1-e^{-\epsilon})+(s-k)(1-e^{-2\epsilon})+(2d'-s-1-(s-k))(e^{-\epsilon}-e^{-2\epsilon})$ that is not lower than $s(1-e^{-\epsilon})+(2d'-s-1)(e^{-\epsilon}-e^{-2\epsilon})$. Combining three case together, we have the lower bound as $\frac{(s-1)(1-e^{-\epsilon})+(2d'-s-1)(e^{-\epsilon}-e^{-2\epsilon})}{(1-e^{-\epsilon})\Omega}$. □

Therefore, the $\tilde{S}(\mathcal{W}(x_j), \mathcal{W}(x_{j+1}))$ is a clone of uniform$(\mathcal{Z}^2_{1,1})$ with probability at least:

$$\begin{aligned}&\frac{\mathbb{P}[\{i_b, i'_{b'}\} \subseteq \tilde{S}(\mathcal{W}(x_j), \mathcal{W}(x_{j+1}))]}{4/(2d'(d'-1))}\\ \geq\ &d'(d'-1))(\frac{s(1 + e^{-2\epsilon} - e^{-\epsilon}) + (2d' - 2 - e^{\epsilon})(e^{-\epsilon} - e^{-2\epsilon})}{(1 - e^{-\epsilon})\Omega})^2,\end{aligned}$$

The denominator 4 comes from the fact that each intermediate output $\{j^*_{b^*}, i_b, j^{*'}_{b^{*'}}, i'_{b'}\}$ is used at most 4 times in redistribution:

$$\{i_b, i'_{b'}, \cancel{j^*_{b^*}}, \cancel{j^{*'}_{b^{*'}}}\} \qquad \{j^*_{b^*}, j^{*'}_{b^{*'}}, \cancel{i_b}, \cancel{i'_{b'}}\}$$
$$\{i_b, j^{*'}_{b^{*'}}, \cancel{j^*_{b^*}}, \cancel{i'_{b'}}\} \qquad \{j^*_{b^*}, i'_{b'}, \cancel{i_b}, \cancel{j^{*'}_{b^{*'}}}\}$$

where the first two elements in each multi-set form a target element that might belongs to $\mathcal{Z}^2_{1,1}$. Further since uniform$(\mathcal{Z}^2_{1,1})$ is a clone of $K(a)$ or $K(b)$ with probability at least $e^{-\epsilon}$, we conclude that the $\tilde{S}(\mathcal{W}(x_j), \mathcal{W}(x_{j+1}))$ is a clone of $K(x_1)$ with probability $\frac{p_{cc}}{(1-e^{-\epsilon})^2}$.

E PROOF OF ONLINE SHUFFLE PRIVACY AMPLIFICATION

Let p denote $e^{-\epsilon}$, to prove the result, we only need to show the $\frac{\delta}{2}$-bound of the variable $C = \sum_{i=1}^{(n-1)/2} c_{(i)}$ is never lower than Φ'. The key observation is that the mean value of $Paired(\epsilon)$ is $np + \frac{np_{cc}}{2}$ and the moment generating function is upper bounded by $TwoSingles(\epsilon)$, which corresponds to the number of clones from two independent private views in the classical clone analogy:

$$TwoSingles(\epsilon) = \begin{cases} 2, & \text{with probability } e^{-2\epsilon};\\ 1, & \text{with probability } 2(1 - e^{-\epsilon})/e^{\epsilon};\\ 0, & \text{else.}\end{cases}$$

We first show the moment-generating function of $Paired(\epsilon) - \mathbb{E}[Paired(\epsilon)]$ is bounded by $TwoSingles(\epsilon) - \mathbb{E}[TwoSingles(\epsilon)]$ when $e^{\epsilon} \leq 3$. Formally, consider the moment-generating function of the $Paired(\epsilon) - \mathbb{E}[Paired(\epsilon)]$:

$$\begin{aligned}M_{paired,p,p_{cc}} = &((1-p)^2 - p_{cc})e^{r(-2p-p_{cc})}\\ &+ (2p(1-p) + p_{cc})e^{r(1-2p-p_{cc})} + p^2 e^{r(2-2p-p_{cc})},\end{aligned}$$

its partial derivation on p_{cc} is $M'_{paired,p,p_{cc}} = e^{-r(2p+p_{cc})}(-1 - e^{2r}p^2 r + r(-(-1+p)^2 + p_{cc}) + e^r(1 + 2(-1+p)pr - rp_{cc}))$. Notice that $(-1 - e^{2r}p^2 s + r(-(-1+p)^2 + p_{cc}) + e^r(1 + 2(-1+p)pr - rp_{cc}))$ is a quadratic function on p, assuming $\log(1 - 1/2) < r < 0$ the coefficient of p^2 is not lower than 0. Given the fact that the second-order derivative $\frac{\partial M'_{paired,p,p_{cc}}}{\partial p}(p = 1/3) \leq 0$, and $M_{paired,1/3,p_{cc}} \leq 0$ holds (i.e., $1/(1 - e^r) - \sqrt{(1 - sp_{cc})/((-1 + e^r)r)} \leq 1/3$ is the smaller root and $1/(1 - e^r) + \sqrt{(1 - rq)/((-1 + e^r)r)} > 1$ is the larger root of the quadratic function), we have $M'_{paired,p,p_{cc}} \leq 0$ holds (when $p > 1/3, 0 \leq p_{cc} \leq (1-p)^2$ and $\log(1 - 1/2) < r < 0$). Consequently, we get $M_{Paired,p,p_{cc}} \leq M_{Paired,p,0}$. When $p_{cc} = 0$, the $Paired(\epsilon)$ is equivalent to the $TwoSingles(\epsilon)$, which implies $C = \sum_{i=1}^{(n-1)/2} c_{(i)} \sim Binomial(n-1, p)$.

Applying the Markov's inequality on the variable e^{rC}, for a margin $\alpha \geq 0$, we have $\mathbb{P}[C < np + nq/2 - \beta] \leq \frac{(M_{paired,p,p_{cc}})^{(n-1)/2}}{e^{r\alpha}} \leq \frac{(M_{TwoSingles,p})^{(n-1)/2}}{e^{r\alpha}} \leq \frac{(M_{\text{Bernoulli},p})^{(n-1)}}{e^{r\beta}} \leq e^{-e^{\epsilon}\alpha^2/(3(n-1))}$. The last step is induced by set up $s = \log(1 - e^{\epsilon}\alpha/n)$, which is larger than $\log(1 - 1/2)$ when $n \geq 16e^{\epsilon}\log(4/\delta)$.

F COMPLEMENTARY EXPERIMENTAL RESULTS

The FutureRand [29] concerned with the asymptotic estimation performance. We here further present experimental results with a large sparsity parameter in Table 4. Similar to the results in Table 2, the FutureRand mechanism suffers significantly more errors than other mechanisms even in the asymptotic setting (e.g., when $\epsilon \to 0$ or s is large). The huge empirical error of FutureRand renders it impractical in these experimental settings.

We also give complementary experimental results in Figures 11, 12 (on the Stock dataset) and Figures 13, 14 (on the Trajectory dataset), for filling the missing error measurements in the main content. As expected, the performance gaps between competitive mechanisms are mostly consistent under various metrics (e.g., MAE/TVE, and with/without post-processing).

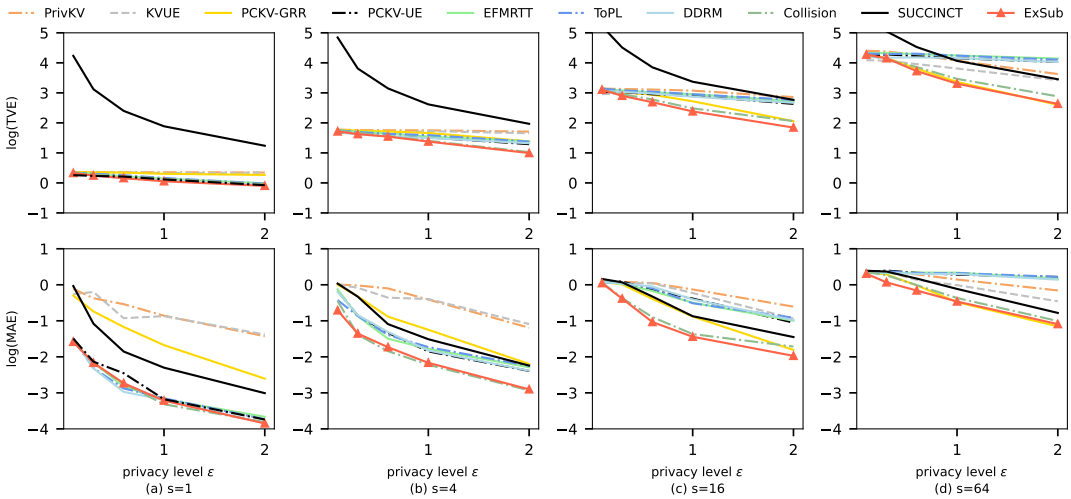

Figure 11: Error results with post-processing with $n = 10000$, $d' = 256$ and sparsity parameter s varies from 1 to 64.

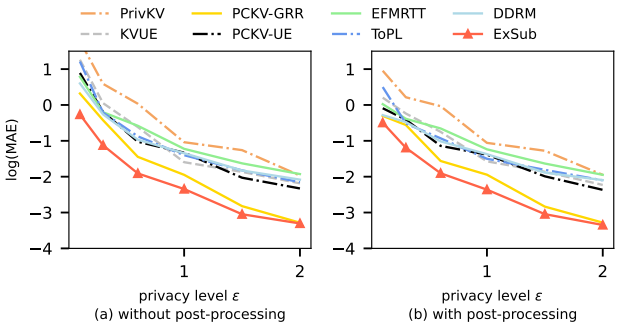

Figure 12: MAE results for mean queries on Stock dataset.

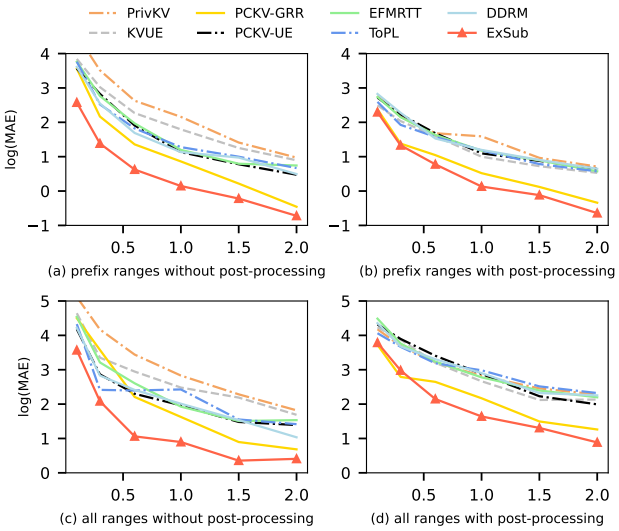

Figure 13: MAE results on range queries of Trajectory dataset.

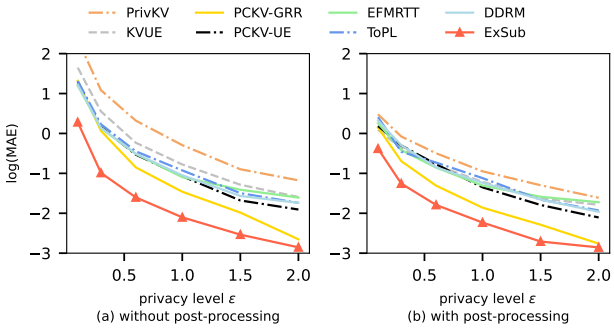

Figure 14: MAE results for mean queries on Trajectory dataset.

Table 4: 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 [40]	6.7e+4	6.8e+3	61.9	17.0	10.4
SUCCINCT [53]	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 [40]	526.4	55.6	0.47	0.17	0.14
SUCCINCT [53]	555.4	55.3	0.66	0.41	0.35
ExSub	426.3	45.2	0.41	0.14	0.11