Differentially Private Streaming Data Release under Temporal Correlations via Post-processing

Xuyang Cao[1], Yang Cao[1], Primal Pappachan[2], Atsuyoshi Nakamura[1], and
Masatoshi Yoshikawa[3]

[1] Graduate School of IST, Hokkaido University, Japan
[2] Portland State University, USA
[3] Faculty of Data Science, Osaka Seikei University, Japan

Abstract. The release of differentially private streaming data has been
extensively studied, yet striking a good balance between privacy and util-
ity on temporally correlated data in the stream remains an open problem.
Existing works focus on enhancing privacy when applying differential pri-
vacy to correlated data, highlighting that differential privacy may suffer
from additional privacy leakage under correlations; consequently, a small
privacy budget has to be used which worsens the utility. In this work,
we propose a post-processing framework to improve the utility of differ-
ential privacy data release under temporal correlations. We model the
problem as a maximum posterior estimation given the released differen-
tially private data and correlation model and transform it into nonlinear
constrained programming. Our experiments on synthetic datasets show
that the proposed approach significantly improves the utility and accu-
racy of differentially private data by nearly a hundred times in terms of
mean square error when a strict privacy budget is given.

Keywords: Differential Privacy · Data Correlations · Time-series Stream
· Continual Data Release · Post-processing.

1 Introduction

Data collection and analysis in many real-world scenarios are performed in a
streaming fashion, such as location traces [23], web page click data [13], and
real-time stock trades. However, releasing data continuously may result in pri-
vacy risks. To this end, *differentially private streaming data release* have been
thoroughly studied [3,4,8,9,12,13,14,15,16,24]. The curator of the database can
use a differentially private mechanism, such as Laplace Mechanism (LM), that
adds noises to the query results at each time point for satisfying a formal privacy
guarantee called ϵ-Differential Privacy (ϵ-DP) [10], where ϵ is the parameter (i.e.,
privacy budget) controlling trade-off between privacy protection and utility of
data release. A small ϵ indicates a high level of privacy and thus requires adding
a larger amount of noise. Taking location traces as an example to elaborate,
Fig. 1 (a) (c) (d) illustrate how differentially private location statistics are re-
leased using LM at each time point where (a) represent real-time location raw

t=	1	2	3	...
u1	loc_2	loc_2	loc_3	...
u2	loc_1	loc_2	loc_3	...
u3	loc_4	loc_1	loc_2	...
u4	loc_3	loc_3	loc_4	...

(a) Raw Data

(b) Temporal Correlations

Counts Query

t=	1	2	3	...
loc_1	1	1	0	...
loc_2	1	2	1	...
loc_3	1	1	2	...
loc_4	1	0	1	...

(c) True Counts

Laplace Noise

t=	1	2	3	...
loc_1	2.5	4.6	4.0	...
loc_2	3.1	0.0	-0.2	...
loc_3	-2.0	2.2	3.0	...
loc_4	4.0	2.0	1.1	...

(d) Private Counts

Fig. 1. Scenario: Differentially Private Streaming Data Release.

data sets (i.e., values of longitude and latitude of residence, company, shopping mall respectively) of users in a database D collected by devices with GPS sensors (i.e., GPS, GNSS) [23], (c) are the true counts of each location computed by a count query function $f(D)$ and (d) what will be released and sent to the public are streaming private counts through a differentially private mechanism such as Laplace Mechanism (LM) [11].

However, recent studies [5,6,18,25,28,29] reveal that, when the data are correlated, more noises have to be added to prevent leakages which deteriorates the utility. They point out that differential privacy algorithms suffer extra privacy leakage on correlated data and develop techniques to enhance differential privacy with a smaller ϵ. In the context of streaming data release, a Markov chain could be used to model the temporal correlations. For example, as shown in Fig. 1 (b), temporal correlation is manifested as the transition probabilities between different locations, which can be obtained through public information such as road networks or traffic data. Based on the temporal correlation presented in Figure 1(b), we have the probability of users proceeding from location $loc1$ to $loc2$ will be $Pr(l^{t+1} = loc2 | l^t = loc1) = 1$ if we have the knowledge that another road is congested. Cao et al. [5,6] quantified such a private leakage and proposed a special privacy definition on temporal correlated data named $\alpha\text{-}DP_{\mathcal{T}}$, to calibrate a smaller privacy budget in order to cover the extra privacy leakage caused by temporal correlations. Song et al. [25] proposed Wasserstein Mechanism for Pufferfish privacy (i.e., a privacy notion that generalizes differential privacy) and Markov Quilt Mechanism specifically when correlation between data is described by a Bayesian Network or a Markov chain. Similar to [5,6], they calculate an

Fig. 2. Existing studies [5, 25] propose approaches for enhancing DP on temporally correlated data; however, these methods sacrifice utility. This work tackles this problem by utilizing temporal correlations as prior knowledge about the data for post-processing purposes.

enlarged ϵ to enhance the privacy but sacrifice more utility. Hence, the challenge is how to boost the utility of differentially private streaming data release on temporally correlated data.

Our approach to addressing the aforementioned issue involves capitalizing on the existing temporal correlations as prior knowledge about the original data through post-processing. Although *post-processing* [?, 17, 20, 22] has been extensively researched as a means to enhance the utility of differential privacy, current methods are ill-equipped to deal with temporal correlations. Post-processing primarily aims to refine differentially private (noisy) results by enforcing them to comply with certain ground-truth constraints or prior knowledge about the data. For instance, *deterministic* consistency constraints between data points are frequently employed in previous studies to represent inherent properties of the data (e.g., released counts in histograms should be integers). In this study, we apply the post-processing technique to improve the utility of differentially private streaming data release in the presence of temporal correlations. By accounting for temporal correlations along with other consistency constraints, we strive to obtain the most accurate current counts which could be estimated from previous private counts while approximating the true current counts.

In this study, we formulate post-processing as a nonlinear optimization problem within the Maximum A Posteriori (MAP) framework, accounting for both probabilistic constraints of temporal correlations and deterministic consistency constraints. Similar to [5, 6, 25], we assume that temporal correlations are public knowledge and are expressed by a transition matrix. As illustrated in Fig. 2, our approach leverages the transition matrix to enhance the utility of differentially private counts. Thus, we pose the problem of determining the most plausible counts that satisfy the constraints (both probabilistic and deterministic) and exhibit the least distance from the released private data. To model this probabilistic distribution, we employ the knowledge of Laplace noise distribution and introduce a Markov chain model to calculate the distribution of true counts. Finally, extensive experiments demonstrate and validate the effectiveness of our methods.

To summarize, our contributions are as follows:

- To the best of our knowledge, this paper presents the first attempt to enhance the utility of differentially private data release under temporal correlations. We propose a post-processing framework using maximum a posteriori estimation, which incorporates both probabilistic correlations and deterministic constraints.
- We implement the post-processing framework for temporal correlations in the differentially private continual data release. Specifically, we formulate this problem as constrained nonlinear programming, which can be solved using off-the-shelf optimization software.
- Our experiments on synthetic data demonstrate the effectiveness of the proposed approach. We show that the utility of differentially private data is significantly improved, with nearly a 100-fold reduction in mean square error under a strict privacy budget, while preserving temporal correlations between data.

We would like to note that this work is an extension of our previous poster paper [2], in which we briefly presented the idea without delving into technical details. This paper provides a clearer and more in-depth exploration of our previous work, offering a comprehensive understanding of the proposed MAP-based post-processing framework.

2   Related Work

Several well-studied methods exist to enhance the utility of differentially private data, as post-processing is an effective tool. In this section, we review related works on improving the utility of private data through post-processing.

One of the most widely studied approaches for utility enhancement is the utilization of *consistency constraints* [17, 20] in data (e.g., the *sum* of the released data should be a fixed number, or the released values should be *integer* in the case of counting queries). In our location traces scenario, these consistency constraints can be expressed by the sum of location records or the total number of users as a fixed value (e.g., n) for each time point, with counts always being integers. Previous works formulate the problem as a least squares estimation (LSE) problem [17] or a maximum likelihood estimation (MLE) problem [20], demonstrating the effectiveness of such post-processing approaches.

Hay et al. [17] focused on improving the accuracy of private histograms through post-processing, solving an LSE problem given consistency constraints such as *sum*, *sorted*, and *positive* to find the 'closest' private histograms that also satisfy these constraints. Furthermore, Lee et al. [20] considered noise distribution (Laplace distribution in their scenario) to boost the utility of private query results. They formulated their post-processing problem as an MSE problem and employed the ADMM algorithm to solve the programming problem.

However, when publishing statistics continually, the data points are often temporally correlated. The post-processing methods mentioned above only focus on single-time data release and cannot efficiently capture probabilistic temporal

correlations. Moreover, it remains unclear how to formulate *probabilistic* correlations as constraints, as existing works assume *deterministic* constraints as prior knowledge about the data. We also observe that many existing works on differentially private streaming data release neither provide a formal privacy guarantee under temporal correlations [3,4,8,12,13,14,15,16,24] nor offer reasonable utility for private outputs. Therefore, our study represents the first attempt to enhance the utility of DP with formal privacy under temporal correlations.

3 Preliminaries

3.1 Differential Privacy

Informally, the DP notion requires any single element in a dataset to have only a limited impact on the output. Namely, if D and D' are two *neighboring* databases, the difference in outputs of executing a randomized algorithm on these databases should be minimal [21].

Definition 1. *(ϵ-DP) A randomized mechanism $\mathcal{M}$ is said to satisfy ϵ-DP, where $\epsilon \geq 0$, if and only if for any neighboring datasets D and D' that differ on one element, we have*

$$\forall T \subseteq Range(\mathcal{M}) : Pr(\mathcal{M}(D) \in T) \leq e^\epsilon Pr(\mathcal{M}(D') \in T)$$

where $Range(\mathcal{M})$ represents the set of all possible outputs of the algorithm of mechanism $\mathcal{M}$, the parameter ϵ represents the privacy budget.

3.2 The Laplace Mechanism

The Laplace Mechanism [11] is the first and probably most widely used mechanism for DP. It satisfies ϵ-DP by adding noise to the output of a numerical function [21].

Definition 2. *(Global sensitivity) Let $D \approx D'$ denote that D and D' are neighboring. The global sensitivity of a query function f, denoted by Δ, is given below*

$$\Delta = \max_{D \approx D'} |f(D) - f(D')|$$

According to the definition of DP, the probability density function of the noise should have the property that if one moves no more than Δ units, the probability should increase or decrease no more than e^ϵ. The distribution of noise that naturally satisfies this requirement is $Lap(\frac{\Delta}{\epsilon})$ [21], which denotes a Laplace distribution with location 0 and scale $\frac{\Delta}{\epsilon}$.

Theorem 1. *(Laplace Mechanism, LM) For any function f, the Laplace mechanism A_f that adds i.i.d noise to each function output f satisfies ϵ-DP.*

$$A_f(D) = f(D) + Lap\left(\frac{\Delta}{\epsilon}\right).$$

Commonly, we denote the scale parameter using $\lambda = \frac{\Delta}{\epsilon}$.

Table 1. Notations

D	A bounded database
Loc	Value domain of locations of all users
l_i^t	The location information of $user_i$ at time t, $user_i \in U$, $l_i^t \in$ **Loc**
$\mathcal{M}$	A differential privacy mechanism over D
R	The set of real continual time-series query outputs
$\tilde{\mathbf{R}}$	The set of added-noise continual time-series query outputs
R^t	The set of query outputs at time t, $R^t \subseteq \mathbf{R}$
r_l^t	A specific query output at time t and location l, $l \in$ **Loc**, $r_l^t \in R^t$
$\mathcal{T}$	The transition matrix of locations
$\mathbf{P^t}$	The possibility of locations for a single user at time t
$Pr(\hat{\mathbf{R}})$	The joint distribution of possible private counts

4   Problem Statement

This section will introduce and formulate the primary issue we aim to address. First, below we present the notations used throughout this paper. We use D to represent a *bounded database* consisting of n users. We prefer to use bold letters to indicate vectors. We use r_l^t to denote a specific query output at a given time point t and location l. $\mathcal{T}$ represents a transition matrix modeling temporal correlations between data. More detailed notations are in Table 1.

Temporally Correlated Stream Data. In our scenario of location traces mentioned above, we assume that n people (labeled from 1 to n) staying at m locations (labeled from 1 to m) respectively at single time point t (shown in Fig. 1 (a)). Let **Loc** denote the sets of locations. Naturally, the data at each time point are temporally correlated: for each user, her current location depends on the previous location in the form a transition matrix $\mathcal{T}$. Without loss of generality, we assume the transition matrix is the same for all users and is given in advance since it can be learned from public information such as road networks. This assumption follows existing works [5, 25].

Differentially Private Stream Data Release. A server collects users' real-time locations l^t at time t in a database D, and aims to release differentially private query results over D. In particular, we consider a query function $f : D \to \mathbf{N}^m$ that counts the total number of people at each location over the entire publishing time T, denoted as $f(D)$. The query outputs are represented by $\mathbf{R} = (R^1, \ldots, R^t, \ldots, R^T)$ and $R^t = (r_1^t, \ldots, r_m^t)$. Many existing works, such as [3,4,8,

Table 2. Transition Matrix

$Pr(l^{t+1}|l^t)$	Loc1	Loc2	Loc3
Loc1	0.33	0.33	0.34
Loc2	0.80	0.10	0.10
Loc3	0.05	0.90	0.05

12, 13, 14, 15, 16, 24], have considered a similar problem setting as ours. However, due to temporal correlations, increased noise is added to the true answers to preserve strict privacy [5, 25], which reduces the utility of the released private counts. Our question is: *can we leverage the temporal correlations to improve the utility of differentially private data via post-processing* (while preserving the enhanced privacy as [5, 25])?

5   Methodology

In this section, we will explain how to formulate the post-processing problem for streaming data release under temporal correlations. To address the above-mentioned challenge, we use post-processing, allowing us to refine the private counts using publicly known prior knowledge.

Intuition. Our core idea is that the temporal correlations can be seen as probabilistic constraints on the data. We can formulate the problem as determining the most probable query outputs $\hat{\mathbf{R}}$ that satisfy such constraints when given $\tilde{\mathbf{R}}$, leveraging the knowledge of $\mathcal{T}$ as shown in Fig. 1 (b). Specifically, we aim to solve the programming problem of maximizing $Pr(\hat{\mathbf{R}}|\tilde{\mathbf{R}})$, subject to the *transition matrix* and other *consistency constraints*. Our method will demonstrate that the estimation depends on the noise distribution and the joint distribution of true counts, which are determined by the mechanism used and the inherent correlations within the raw data.

5.1   Maximum A Posterior Estimation Framework for Correlated Data

Firstly, we propose a Maximum A Posterior (MAP) Estimation framework to assist formulating probabilistic post-processing problem.

Definition 3. *(MAP Framework) Let D be a bounded database with n records. A post-processing approach is feasible under a framework $\mathcal{F}(\mathcal{M},\mathcal{C})$ if for all noisy query results $\tilde{Q} \in \mathcal{O}$ through a given privacy mechanism $\mathcal{M}$, we have*

$$P(\hat{Q}|\tilde{Q}) = \frac{P(\tilde{Q}|\hat{Q})P(\hat{Q})}{P(\tilde{Q})}, \tag{1}$$

$$\hat{Q}^* = \arg\max_{\hat{Q}} P(\tilde{Q}|\hat{Q})P(\hat{Q}) \tag{2}$$

where $\mathcal{C}$ represents correlations between data for all true query results Q, $\mathcal{O}$ is denoted as all possible output set of $\mathcal{M}(D)$, $\hat{Q}$ and $\hat{Q}^*$ are variable and our desired 'closest' query result which also meets correlation $\mathcal{C}$ respectively.

We apply MAP Framework to solve post-processing problem of streaming data release under temporal correlations. Given a mechanism $\mathcal{M}$(i.e., Laplace Mechanism here) and temporal correlations $\mathcal{C}$ between data, the 'closest' private counts $\hat{\mathbf{R}}$ is tended to be obtained by calculating the maximum of the posterior possibility under MAP framework $\mathcal{F}(\mathcal{M},\mathcal{C})$

$$Pr(\hat{\mathbf{R}}|\tilde{\mathbf{R}}) = \frac{Pr(\tilde{\mathbf{R}}|\hat{\mathbf{R}})Pr(\hat{\mathbf{R}})}{Pr(\tilde{\mathbf{R}})} \tag{3}$$

subjecting to the correlations $\mathcal{C}$ and other constraints if exist. For convenience, the logarithm form of the above formula is applied

$$\ln Pr(\hat{\mathbf{R}}|\tilde{\mathbf{R}}) = \ln Pr(\tilde{\mathbf{R}}|\hat{\mathbf{R}}) + \ln Pr(\hat{\mathbf{R}}) - \ln Pr(\tilde{\mathbf{R}}) \tag{4}$$

Therefore, the objective 'closest' query outputs (achieve the maximum of (3)) after post-processing will be

$$\hat{\mathbf{R}}^* = \arg \max_{\hat{\mathbf{R}}} \{\ln Pr(\tilde{\mathbf{R}}|\hat{\mathbf{R}}) + \ln Pr(\hat{\mathbf{R}})\} \tag{5}$$

when the private counts $\tilde{\mathbf{R}}$ is given.

In essence, the first term and the second term of right side of (5) come from $\mathcal{M}$ and $\mathcal{C}$ respectively. What makes it different from prior works is that we focus on calculating the joint distribution of private counts $Pr(\hat{\mathbf{R}})$ which are simply viewed as a uniform distribution, namely a constant, in most of previous works. We point out that it cannot be omitted when there are correlations between data especially under temporal correlations.

5.2   Calculation of Terms of Objective Equation

The next steps are how to calculate the left two terms in the right side of (5).

Calculation of the first term For this term, it tells us that noises should be considered while improving accuracy and [20] also points out that we are able to formulate it into a L_1 function if the noises come from LM[4]. Thus, we formulate the first term of (5) in the following

$$\ln Pr(\tilde{\mathbf{R}}|\hat{\mathbf{R}}) = -\frac{1}{\lambda}||\tilde{\mathbf{R}} - \hat{\mathbf{R}}||_{L_1} + Const. \tag{6}$$

---

[4] Please note that our method can be applied to other mechanisms. However, for the duration of this article, we have temporarily chosen to default to the Laplace Mechanism.

Calculation of the second term A *Markov Chain* model is introduced to calculate the possibilities of single user's locations released in continual time-series stream because the possibility of present location only relies on previous one. With the transition matrix and a prior distribution of locations of single user at $t = 1$, we are able to calculate user's probability distribution of location at any time. We introduce two policies to obtain the prior distribution: (a) the first one is to use the normalized frequency of private counts at $t = 1$, $\tilde{R}^1$ (*frequency p-d*); (b) another is to simply use a uniform distribution (*uniform p-d*) instead. Consequently, we can derive all the possibilities of moving next locations at each time t, $\mathbf{P}^t$, expressed as below:

$$\mathbf{P}^t = \mathbf{P}^{t-1}\mathcal{T} \tag{7}$$

for each $t \in \{2, \ldots, T\}$. However, a joint distribution of users' locations should be calculated when given a bounded database containing data of n users.

Note that all of n users are independent here which means their next actions will not be influenced by others. With the probability distribution of location of single user at each time, therefore, the joint distribution of all location counts at specific time point can be expressed by a *multinomial distribution*

$$Pr(R^t) = n! \prod_l \frac{(\mathbf{P}_l^t)^{r_l^t}}{r_l^t!} \tag{8}$$

for each $R^t \subseteq \mathbf{R}$ where n represents total number of users.

Recall the *Stirling's Approximation*

$$\ln x! \approx \frac{\ln 2\pi x}{2} + x \ln \frac{x}{e} \tag{9}$$

We apply the approximation (9) to mitigate our calculation

$$\ln Pr(R^t) \approx \ln n! + \sum_l (r_l^t \ln \mathbf{P}_l^t - \frac{\ln 2\pi r_l^t}{2} - r_l^t \ln \frac{r_l^t}{e}) \tag{10}$$

Naturally, our 'closest' query answer $\hat{\mathbf{R}}$ also obeys this multinomial distribution the same as true query answer.

5.3 Nonlinear Constrained Programming

We conclude our method of formulating this post-processing problem under temporal correlations into a nonlinear constrained programming problem. By calculating the minimum estimation of $-\ln Pr(\hat{\mathbf{R}}|\tilde{\mathbf{R}})$ and combining with (6) (10),

we finally transform (5) into a nonlinear constrained programming as below

$$\text{Minimize } \frac{1}{\lambda}||\tilde{\mathbf{R}} - \hat{\mathbf{R}}||_{L_1}$$

$$-\sum_{t=1}^{T}\sum_{l}(r_l^t \ln \mathbf{P}_l^t - \frac{\ln 2\pi r_l^t}{2} - r_l^t \ln \frac{r_l^t}{e})$$

$$\text{Subject to } \sum r_l^t = n, \text{ for each } t \in \{1, 2, \ldots, T-1, T\}$$

$$r_l^t \geq 0, \text{ for each } t \in \{1, 2, \ldots, T-1, T\}$$

where $\hat{\mathbf{R}} = \big((r_1^1, \ldots, r_m^1), \ldots, (r_1^T, \ldots, r_m^T)\big)$.

Then, we point out that this nonlinear constrained programming is solvable. By introducing augmented *Lagrangian* to our objective function (O.F.), there are many convergence results proved in the literature (e.g. *ADMM* [1]) where we could prove the O.F. will finally converge as dual variables converge. Also, variables r_l^t must satisfy $\sum r_l^t = n$ and $r_l^t \geq 0$ simultaneously. Thus, the boundary of r_l^t is $n \geq r_l^t \geq 0$.

Asymptotical analysis. As shown in derived objective function, there are two terms which represent the contribution from Mechanism applied to true counts and Correlations between true counts respectively. As ϵ approaches zero, the first term, namely $\frac{1}{\lambda}||\tilde{\mathbf{R}} - \hat{\mathbf{R}}||_{L_1}$ will also approach to zero because of coefficient λ. In other words, the second term

$$-\sum_{t=1}^{T}\sum_{l}(r_l^t \ln \mathbf{P}_l^t - \frac{\ln 2\pi r_l^t}{2} - r_l^t \ln \frac{r_l^t}{e})$$

will matter the most to objective function when a stricter privacy budget ϵ is given. Also, we'd like to analyze what the objective function will perform if a 'weak' level correlation is given (note that we will provide a mathematical definition of levels of correlations in our Experiments part) such that probabilities of proceed to the next location from previous ones is a fixed value, namely $\mathbf{P}_l^t$ is a uniform distribution. Then, the second term is able to be 'ignored' and the first term

$$\frac{1}{\lambda}||\tilde{\mathbf{R}} - \hat{\mathbf{R}}||_{L_1}$$

will thus matter the most to solutions. We should point out that our framework will result in an MLE problem such that post-processing problem mentioned by [20] if there is no correlations amount original data.

6   Experiments

In this section, we present experimental results that demonstrate the effectiveness of our proposed MAP framework for post-processing continuous data release under temporal correlations. To validate our method, we apply it to both synthetic and real-world datasets, and evaluate its performance in terms of accuracy

and utility. Furthermore, we have made our code available on GitHub[5], enabling other researchers to reproduce our experiments and extend our work. For statistical significance, all experiments are performed 50 times and the mean values are reported as the final results.

Environment. The experiments were executed on CPU: $Intel(R)Core(TM)$ $i7 - 11370H$ @$3.30GHz$ with $Python$ version 3.7.

Nonlinear Programming Solver. The solver used for solving nonlinear constrained programming is $Gurobi\ Optimizer\ version$ 10.0.1 API for $Python$.

Level of Temporal Correlations. To evaluate the performance of our post-processing method under different temporal correlations, we introduce a method to generate transition matrix in different *levels*. To begin with, we default a transition matrix indicating the "strongest" correlations which contains probability 1.0 in its diagonal cells. Then, we utilize *Laplacian smoothing* [26] to uniform the possibilities of $n \times n$ transition matrix $\mathcal{T}^S$ of 'strongest' correlations. Next, let p_{ij} denote the element at the ith row and jth column of $\mathcal{T}^S$. The uniformed possibilities $\hat{p}_{i,j}$ can be generated from (11), where s $(0 \leq s < \infty)$ is a positive parameter that controls the levels of uniformity of probabilities in each row. That's, a smaller s means stronger level temporal correlations. Also, We should note that, different s are only comparable under the same n.

$$\hat{p}_{i,j} = \frac{p_{ij} + s}{\sum_{j=1}^{n}(p_{ij} + s)} \tag{11}$$

6.1 Utility Analysis

In this subsection, we conduct a utility analysis using the objective function of the nonlinear constrained programming approach described above. The objective function consists of two parts: the noise distribution and the joint distribution of query answers under temporal correlations. The key to the effectiveness of our post-processing method in achieving high utility lies in its ability to recover the correlations between data that are blurred by incremental noise added to the original query answers. For example, it enables the preservation of the correlation that 'the current number of people staying at loc1 must equal the previous number of people staying at loc2' by solving the relevant nonlinear constrained programming problem. As a result, the similarity between the post-processing query answers and the original query answers is improved significantly.

Moreover, we introduce MSE and $Possibility$ as metrics to measure the utility of optimal counts instead of MSE solely for supporting the validation of our MAP post-processing method. For instance, synthetic streaming binary counts, such that total number of locations is $n_{loc} = 3$, total number of users is $n_{user} = 1$, are going to be released under $\epsilon-$DP. And the temporal correlations are known to the public which can be expressed by transition matrix $\mathcal{T} = \begin{Bmatrix} 0.0\ 0.0\ 1.0 \\ 0.5\ 0.0\ 0.5 \\ 0.0\ 1.0\ 0.0 \end{Bmatrix}$ which also represents the basic temporal correlation used for

---
[5] https://github.com/DPCodesLib/DBSec23

(a) MSE of Binary Counts Release

(b) Possibilities of Varying

Fig. 3. a) Scenario: Streaming Binary Counts Released under Temporal Correlations; b) Possibility of Proceeding to Current Counts from Prior Counts of Post-processing Results (under $1-$DP)

generating synthetic datasets. Then, we post-process and release optimal counts using post-processing methods of MLE with a ADMM algorithm [20] and our MAP framework respectively illustrated by Fig. 3(a). The red line of Fig. 3(a) represents optimal results that drop temporal correlations obtained by calculating MLE problem while the greed and blue lines represent the optimal results obtained from our method of MAP framework under two different strategies. The details of them will be revealed in the following subsections. When calculating the possibilities of achieving current counts from previous counts (shown in Fig. 3(b)), however, we note that many possibilities of post-processing points of dropping temporal correlations are lower than cut-off line (10^{-10}) which will be seen as 'impossible events' if possibility is smaller than 10^{-10}. It proves that our MAP framework is able to preserve the probabilistic properties owned by original data, namely temporal correlations, compared with prior post-processing methods.

We will now explore the tradeoff between privacy and utility. The objective function reveals that the privacy budget ϵ is a weight parameter affecting the noises' part, but it has no impact on the correlations' part. This means that the correlations' part is dominant when the privacy budget is strict, while the noises' part replaces it when the budget is lax. As a result, the utility is always preserved under any given privacy budget, since the method always preserves known correlations when calculating the 'closest' private counts.

6.2   Synthetic Datasets

To thoroughly examine the feasibility and effectiveness of our MAP framework and related post-processing methods, we conduct an evaluation on various synthetic datasets. This evaluation aims to provide a comprehensive understanding of the performance of our approach under different scenarios and to validate its potential for practical applications.

(a) MSE over ϵ under $\epsilon-$DP   (b) MSE over α for under $\alpha-$DP$_\mathcal{T}$

Fig. 4. a) MSE over ϵ under $\epsilon-$DP; total release time is $T = 500$, total number of users is $n_{user} = 200$ and level of correlations is $s = 0$. b)MSE over α for under $\alpha-$DP$_\mathcal{T}$; total release time is $T = 500$, total number of users is $n_{user} = 200$ and level of correlations is $s = 0.01$.

MSE vs Privacy Budget ϵ or α

Here, we compare the performance of our post-processing method by varying privacy budget ϵ or α from 0.2 to 2.0 (with $step = 0.2$) at a given total publishing time T in different mechanisms, $\epsilon - DP$ and $\alpha - DP_\mathcal{T}$, respectively. Note that we must choose a prior distribution(p-d) for P^1 when $t = 1$, and our strategy is to use the frequency of $\tilde{R}^1$ or a uniform distribution to substitute for it. And the results, shown in Fig. 4 (a) and Fig. 4 (b), illustrate that our post-processing method significantly improves the utility and accuracy of outputs while achieving a desired privacy budget both in $\epsilon - DP$ and $\alpha - DP_\mathcal{T}$.

The red line which represents prior method of MLE using ADMM only considers utilizing public knowledge of mechanisms instead of both mechanisms and correlations to boost utility of released counts. The blue line and green line are the results after our post-processing given two different policy to choose p-d. As shown in figure, MSE become smaller while increasing privacy budget ϵ. And our methods perform better than MLE method by decreasing MSE nearly hundred times at any given fixed ϵ.

MSE vs Total Release Time

We vary the total publishing time T from 100 to 200 ($step = 10$) to examine the performance of our post-processing method under both ϵ-DP and α-DP$_\mathcal{T}$, using the same methods for generating the synthetic datasets as described above. We use default privacy budgets of $\alpha, \epsilon = 1.0, 1.0$.

The results of our experiments, as shown in Figure 5, indicate that the mean squared error (MSE) values of ϵ-DP and α-DP$_\mathcal{T}$ increase significantly as the total release time is extended. However, our post-processing method demonstrates a remarkable boost in utility, as both of its policies consistently yield lower MSE values than the method that drops temporal correlations. These findings underscore the effectiveness of our post-processing method in preserving the correla-

Fig. 5. MSE over Total Release Time under $\alpha-\mathrm{DP}_{\mathcal{T}}$ and $\epsilon - DP$; privacy budget is $\alpha = 1.0$, $\epsilon = 1.0$ and level of correlations is $s = 0.01$.

tions between raw data, and the importance of considering temporal correlations when designing and evaluating different data release mechanisms.

MSE vs Different Temporal Correlations

In this subsection, we finally check the performance of our post-processing method upon different intensities of temporal correlations. We default the privacy budget and total publishing time as $\alpha = 1.0$ and $T = 500$ respectively. Note that it will have relatively higher temporal correlations if users have a higher possibility from present location to the next specific location (e.g., $Pr(l_i^t|l_i^{t-1}) = 1.0$). Therefore, we firstly generate a transition matrix $\mathcal{T} = \left\{ \begin{array}{ccc} 0.0 & 0.0 & 1.0 \\ 0.5 & 0.0 & 0.5 \\ 0.0 & 1.0 & 0.0 \end{array} \right\}$.

Then, we apply (11) to generate different level degree of correlations, weak correlations, medium correlations and strong correlations corresponding to $s = 1$, $s = 0.1$, $s = 0.01$ respectively.

And the results, shown in Fig. 6, reveal validation of this post-processing method by giving prominent improvement in accuracy. We also compare a special post-processing method that drops temporal correlations which means that the joint distribution of query results $Pr(R)$ is a constant. And the results show that the post-processing method with temporal correlations will achieve higher utility with a lower MSE.

Fig. 6. MSE over Different Levels of Temporal Correlations under $\alpha-\mathrm{DP}_{\mathcal{T}}$ and $\epsilon-\mathrm{DP}$; privacy budget is $\alpha = 1.0$ and $\epsilon = 1.0$, total number of users is $n_{user} = 200$ and total release time is $T = 500$.

These experiments also highlight the essential role of the MAP framework, demonstrating that correlations between raw data can significantly impact the results and cannot be disregarded in both the mechanism design and post-processing stages.

7   Conclusion

In this paper, we have shown that temporal correlations are often present in differential privacy data releases and proposed a MAP framework to address the post-processing problem in this context. Our experiments demonstrate the effectiveness of incorporating temporal correlations into the post-processing step, resulting in significant improvements in accuracy and utility.

Furthermore, our work suggests that the MAP framework can be a useful tool for addressing other post-processing problems involving correlated data, such as Bayesian DP and Pufferfish Privacy Mechanisms. While our approach assumes independence between users, this may not always hold true in practice. Future work could explore how to extend our framework to address post-processing for streaming data releases under temporal correlations when users are correlated.

Overall, our work contributes to advancing the state of the art in differential privacy data releases by providing a new perspective on post-processing under temporal correlations and opens up new avenues for future research in this area.

Acknowledgments

References

1. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
2. Xuyang Cao, Yang Cao, Masatoshi Yoshikawa, and Atsuyoshi Nakamura. Boosting utility of differentially private streaming data release under temporal correlations. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 6605–6607, 2022.
3. Yang Cao and Masatoshi Yoshikawa. Differentially Private Real-Time Data Release over Infinite Trajectory Streams. In *2015 16th IEEE International Conference on Mobile Data Management (MDM)*, volume 2, pages 68–73, June 2015.
4. Yang Cao and Masatoshi Yoshikawa. Differentially Private Real-Time Data Publishing over Infinite Trajectory Streams. *IEICE TRANSACTIONS on Information and Systems*, E99-D:163–175, January 2016.
5. Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. Quantifying Differential Privacy under Temporal Correlations. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 821–832, April 2017.
6. Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. Quantifying differential privacy in continuous data release under temporal correlations. *IEEE Transactions on Knowledge and Data Engineering*, 31(7):1281–1295, 2019.
7. Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. Differentially private high-dimensional data publication via sampling-based inference. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 129–138, 2015.
8. Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. PeGaSus: Data-Adaptive Differentially Private Stream Processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1375–1388, 2017.
9. Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. Real-world trajectory sharing with local differential privacy. *Proceedings of the VLDB Endowment*, 14(11):2283–2295, jul 2021.
10. Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.
11. Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
12. Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 715–724, 2010.
13. Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, 2014.
14. Liyue Fan, Li Xiong, and Vaidy Sunderam. FAST: Differentially Private Real-time Aggregate Monitor with Filtering and Adaptive Sampling. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 1065–1068, 2013.

15. Arik Friedman, Izchak Sharfman, Daniel Keren, and Assaf Schuster. Privacy-Preserving Distributed Stream Monitoring. In *NDSS*, 2014.

16. Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. Differentially Private Event Sequences over Infinite Streams. *Proc. VLDB Endow.*, 7:1155–1166, August 2014.

17. Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially-private queries through consistency. In *36th International Conference on Very Large Databases (VLDB)*, 2010.

18. Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A Framework for Mathematical Privacy Definitions. *ACM Trans. Database Syst.*, 39:3:1–3:36, January 2014.

19. Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):1–36, 2014.

20. Jaewoo Lee, Yue Wang, and Daniel Kifer. Maximum likelihood postprocessing for differential privacy under consistency constraints. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 635–644, New York, NY, USA, 2015. Association for Computing Machinery.

21. Ninghui Li, Min Lyu, Dong Su, and Weining Yang. Differential privacy: From theory to practice. *Synthesis Lectures on Information Security, Privacy, and Trust*, 8(4):1–138, 2016.

22. Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pages 4435–4444. PMLR, 2019.

23. Heeket Mehta, Pratik Kanani, and Priya Lande. Google maps. *International Journal of Computer Applications*, 178(8):41–46, May 2019.

24. Darakhshan Mir, S. Muthukrishnan, Aleksandar Nikolov, and Rebecca N. Wright. Pan-private Algorithms via Statistics on Sketches. In *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '11, pages 37–48, 2011.

25. Shuang Song, Yizhen Wang, and Kamalika Chaudhuri. Pufferfish privacy mechanisms for correlated data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1291–1306, 2017.

26. O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, page 175–184, New York, NY, USA, 2004. Association for Computing Machinery.

27. Ziang Wang and Jerome P Reiter. Post-processing differentially private counts to satisfy additive constraints. *Transactions on Data Privacy*, 14:65–77, 2021.

28. Bin Yang, Issei Sato, and Hiroshi Nakagawa. Bayesian differential privacy on correlated data. In *Proceedings of the 2015 ACM SIGMOD international conference on Management of Data*, pages 747–762, 2015.

29. Tianqing Zhu, Ping Xiong, Gang Li, and Wanlei Zhou. Correlated differential privacy: Hiding information in non-iid data set. *IEEE Transactions on Information Forensics and Security*, 10(2):229–242, 2015.