Differentially Private Real-Time Release of Sequential Data

XUERU ZHANG, Computer Science and Engineering, The Ohio State University, USA
MOHAMMAD MAHDI KHALILI, Computer and Information Sciences, University of Delaware, USA
MINGYAN LIU, Electrical and Computer Engineering, University of Michigan, USA

Many data analytics applications rely on temporal data, generated (and possibly acquired) sequentially for online analysis. How to release this type of data in a privacy-preserving manner is of great interest and more challenging than releasing one-time, static data. Because of the (potentially strong) temporal correlation within the data sequence, the overall privacy loss can accumulate significantly over time; an attacker with statistical knowledge of the correlation can be particularly hard to defend against. An idea that has been explored in the literature to mitigate this problem is to factor this correlation into the perturbation/noise mechanism. Existing work, however, either focuses on the offline setting (where perturbation is designed and introduced after the entire sequence has become available), or requires a priori information on the correlation in generating perturbation. In this study we propose an approach where the correlation is learned as the sequence is generated, and is used for estimating future data in the sequence. This estimate then drives the generation of the noisy released data. This method allows us to design better perturbation and is suitable for real-time operations. Using the notion of differential privacy, we show this approach achieves high accuracy with lower privacy loss compared to existing methods.

CCS Concepts: • **Security and privacy → Privacy protections**.

Additional Key Words and Phrases: differential privacy, sequential data

ACM Reference Format:
Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. 2021. Differentially Private Real-Time Release of Sequential Data. 1, 1 (June 2021), 28 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The collection and analysis of sequential data are crucial for many applications, such as monitoring web browsing behavior, analyzing daily physical activities recorded by wearable sensors, and so on. Privacy concerns arise when data is shared with third parties, a common occurrence. Toward this end, differential privacy [Dwork 2006] has been widely used to provide a strong privacy guarantee; it is generally achieved by disclosing a noisy version of the underlying data so that changes in the data can be effectively obscured.

To achieve differential privacy in sharing sequential data, a simple approach is to add independent noise to the data at each time instant (Figure 1(a)). This is problematic because of the temporal correlation in the data (see Section 3). A number of studies have attempted to address this issue. For example, [Rastogi and Nath 2010] applies Discrete Fourier Transform (DFT) of the sequence and release a private version generated using inverse DFT with the perturbed DFT

Authors' addresses: Xueru Zhang, Computer Science and Engineering, The Ohio State University, Columbus, OH, USA, zhang.12807@osu.edu; Mohammad Mahdi Khalili, Computer and Information Sciences, University of Delaware, Newark, DE, USA, khalili@udel.edu; Mingyan Liu, Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI, USA, mingyan@umich.edu.

coefficients; [Wang et al. 2017] proposes a correlated perturbation mechanism where the correlated noise is generated based on the autocorrelation of the original sequence; [Kellaris and Papadopoulos 2013] decomposes the sequence into disjoint groups of similar data, and uses the noisy averages of these groups to reconstruct the original sequence; [Xiao and Xiong 2015] constructs a Hidden Markov Model (HMM) from the independent-noise-added data sequence, and releases the sequence inferred from the HMM; method proposed in [Fioretto and Van Hentenryck 2019] first reconstructs the non-sampled data from perturbed sampled points and then solves a convex optimization to improve accuracy. [Ghane et al. 2019] proposes *Trajectory Generative Mechanism* that generates synthetic trajectory data under differential privacy; it first learns a probabilistic graphical generative model that encodes the trajectories, synthetic trajectory is then generated privately based on it. [Gursoy et al. 2018] developes a differentially private synthetic trajectory publisher (DP-Star) for spacial data; it first reduces each trajectory into a sequence of representative points via the minimum description length principle, a synthetic trajectory is then generated privately from these representative points. [Chen et al. 2011] proposes a data-dependent sanitization algorithm that generates a differentially private release for trajectory data; it first constructs a noisy prefix tree to group the sequences with same prefixes into the same branch, sanitized data is then generated from it. This approach was further extended in [Chen et al. 2012] by using a variable n-gram model to sanitize general sequential data. [Hua et al. 2015] considers more general trajectories by removing the implicit assumption used in [Chen et al. 2012, 2011] that the trajectories contain a lot of identical prefixes or n-grams. However, all of the above studies rely on the availability of the entire sequence, so can only be applied offline as post-processing methods. [Fan and Xiong 2014] are the closest to our work, where the sequence is adaptively sampled first; Kalman/particle filters are then used to estimate non-sampled data based on the perturbed sampled data. However, it requires a priori knowledge of the correlation of the sequence.

In this paper we start from sequential data that can be modeled by first-order autoregressive (AR(1)) processes. We consider Gaussian AR(1) process as an example but the idea can be generalized to all (weakly) *stationary* processes. Leveraging time-invariant statistical properties of stationary process, our proposed approach in each time step estimates the unreleased, future data from that already released, using correlation learned over time and not required a priori. This estimate is then used, in conjunction with the actual data observed in the next time step, to drive the generation of the noisy, released version of the data (Figure 1(b)). Both theoretical analysis and empirical results show that our approach can release a sequence of high accuracy with less privacy loss.

Other related works: Our work focuses on *user-level* privacy, where each individual generates a sequence of data and we aim to guarantee its privacy at all times. In contrast, some works in the literature study *event-level* privacy, which only guarantees the privacy at one time step[1]. For example, [Chan et al. 2011; Dwork et al. 2010] propose binary tree mechanism which at each time outputs the sum of the binary data points seen so far under event-level privacy. [Perrier et al. 2018] generalizes these works to continuous data and developed a mechanism based on binary tree mechanism. However, their method is only applicable to data sequences that obey a light-tailed distribution (i.e., distribution whose tail lies below the exponential distribution) and cannot be operated fully online (i.e., a sufficiently large time-lag is required before continual release).

Another line of research aims to develop new privacy notions for correlated data. Among them, some focus on correlation between users [Liu et al. 2016; Yang et al. 2015; Zhu et al. 2015], while others focus on correlation between data at different time steps. For example, [Song et al. 2017] proposes the notion of *pufferfish privacy* which captures data correlation using a Bayesian Network, and Markov Quilt Mechanism was built that releases a data instance at one

[1]User-level privacy is much stronger than event-level privacy because it guarantees the privacy of all events.

time step under the proposed privacy notion. [Cao et al. 2019] proposes *temporal privacy leakage* (TPL) to quantify privacy leakage of a temporal correlated sequence; they assume the probabilistic correlations between data points in every two consecutive time steps are known and based on which develop an algorithm to calculate TPL. A privacy budget allocation mechanism was also developed to convert a traditional differential privacy (DP) mechanism into TPL mechanism. In contrast to these works, our work focuses on the releasing mechanism rather than the privacy definition used to measure its efficacy. Specifically, we proposed an approach that can effectively reduce the total information leakage as data is released. The algorithmic property of our mechanism is orthogonal to the privacy notion used.

Some works focus on developing mechanisms that dynamically allocate privacy budgets over time. For example, [Kellaris et al. 2014] considers a setting where a subsequence of fixed length d is released at each time, and develops two privacy budget allocation mechanisms (i.e., Budget Distibution and Budget Absorption) that dynamically allocate privacy budget over time based on the dissimilarity between the previous released data and new data. [Cao and Yoshikawa 2015] develops dynamic privacy budget allocation and approximation framework, where privacy budget decays exponentially over time and the previously released noisy data may be re-published if they are sufficiently close to the future data.

The rest of the paper is organized as follows. Section 2 presents background and preliminaries. Section 3 introduces the baseline approach and its issues. Our approach is presented and analyzed in Sections 4, 5 and 6. Section 7 presents Discussion. Experiments are presented in Section 8 and Section 9 concludes the paper. All proofs are given in the appendices.

2 PRELIMINARIES

Consider a time-varying sequence $\{Z_t\}_{t=1}^T$, where $Z_t \in \mathbb{R}$ corresponds to a query $Q \in \mathcal{D} \to \mathbb{R}$ over a private dataset D_t at time $t \in \mathbb{N}$, i.e., $Z_t = Q(D_t)$. The dataset $D_t = \{d_t^i\}_{i=1}^N \in \mathcal{D}$ consists of data from N individuals ($N \geq 1$) where d_t^i denotes the data of the i^{th} individual at time step t and $\mathcal{D}$ denotes the set of all possible datasets. Then $d_{1:T}^i = \{d_t^i\}_{t=1}^T$ is the data of the i^{th} individual over T time steps and $D = \{d_{1:T}^i\}_{i=1}^N$ includes sequences of N individuals over T time steps.

We assume $\{Z_t\}_{t=1}^T$ can be modeled as a first-order autoregressive (AR(1)) process [Wei 2006], where the value at each time depends linearly on the value of the immediate preceding time step; we will see that the approach can be generalized to any (weakly) *stationary* process. The goal is to disclose/release this data in real time with privacy guarantees for each individual at all times. We denote by $\{X_t\}_{t=1}^T$ the released sequence. Notationally, we will use X to denote a random variable with probability distribution $\mathcal{F}_X(\cdot)$, x its realization and $\hat{X}(y)$ the estimate of X given observation $Y = y$; finally, $X_{1:t} := \{X_i\}_{i=1}^t$.

2.1 First-order autoregressive process

AR(1) processes are commonly used for modeling a time series, among which Gaussian AR(1) process is one type that is widely used in various domains.

Definition 2.1. **(Gaussian AR(1) process)** $Z_{1:T}$ is a Gaussian AR(1) process [Wei 2006] if:

$$Z_t = \alpha + \rho Z_{t-1} + U_t,\ t \geq 1 \tag{1}$$

where $U_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma_u^2)$, $Z_0 \sim \mathcal{N}(\mu, \sigma_z^2)$ and σ_u^2, α, ρ are constants. If $|\rho| < 1$, then $\{Z_t\}_{t=1}^T$ is a stationary Markov process with the following properties: (1) $Z_t \sim \mathcal{N}(\mu, \sigma_z^2)$ with $\mu = \frac{\alpha}{1-\rho}$ and $\sigma_z^2 = \frac{\sigma_u^2}{1-\rho^2}$; (2) its autocorrelation function is given by $\text{Corr}(Z_t Z_{t-\tau}) = \text{Corr}(\tau) = \rho^{|\tau|}$.

In this paper, Binomial AR(1) process is also studied and all results on Binomial AR(1) process are presented in Appendix A.

2.2 Differential privacy

Definition 2.2. **((ϵ, δ)- differential privacy (DP) [Dwork et al. 2006])** A randomized algorithm $\mathcal{A}(\cdot)$ taking dataset D as input satisfies (ϵ, δ)-differential privacy if for any D, $\widehat{D}$ that are different in at most one individual's data and for any set of possible output $S \subseteq \text{range}(\mathcal{A})$, we have

$$\Pr(\mathcal{A}(D) \in S) \leq \exp(\epsilon) \cdot \Pr(\mathcal{A}(\widehat{D}) \in S) + \delta,$$

where ϵ and $\delta \in [0, 1]$ are privacy parameters and ϵ bounds the privacy loss. Smaller ϵ and δ mean stronger privacy guarantee.

In this study we consider the setting where each individual's data is of a sequential nature and the response to a query Q over N individuals is released over T time steps as it is generated. Within this context, $x_{1:T} = \mathcal{A}(z_{1:T}) = \mathcal{A}(\{Q(D_t)\}_{t=1}^T)$ and a randomized algorithm $\mathcal{A}(\cdot)$ is (ϵ, δ)-differentially private if $\mathcal{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|z_{1:T}) \leq \exp(\epsilon) \cdot \mathcal{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|\widehat{z}_{1:T}) + \delta$ holds for any possible $x_{1:T}$ and any pairs of $z_{1:T}$, $\widehat{z}_{1:T}$ generated from D, $\widehat{D}$, where D, $\widehat{D}$ are datasets differing in at most one individual's sequence.[2] It suggests that the released sequence $x_{1:T}$ should be relatively insensitive to the change of one individual's sequential data, thereby preventing any meaningful inference on any individual from observing $x_{1:T}$. Differential privacy has been widely adopted as the privacy notion and privacy-preserving technique for different applications such as machine learning [Abadi et al. 2016; Khalili et al. 2021b,a; Wei et al. 2020; Zhang et al. 2018a,b, 2020], data mining [Friedman and Schuster 2010; Task and Clifton 2012], data market [Khalili et al. 2019, 2021c; Li et al. 2014; Zheng 2020], etc.

Definition 2.3. (**Sensitivity of query Q at time t**) Consider a query $Q : \mathcal{D} \to \mathbb{R}$ taking a dataset as input, the sensitivity of Q at t is defined as: $\Delta Q_t = \sup_{\forall D_t \sim \widehat{D}_t} |Q(D_t) - Q(\widehat{D}_t)|$, where $D_t, \widehat{D}_t \in \mathcal{D}$ are two datasets at t that are different in at most one individual's data.

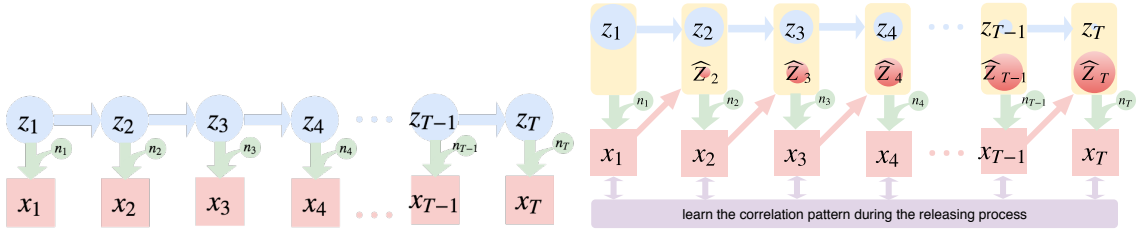
Since $Z_t = Q(D_t)$, ΔQ_t quantifies the maximum impact of an individual on Z_t. In the rest of paper, unless explicitly stated, we consider scenarios where ΔQ_t does not change over time and use the notation $\Delta Q_t = \Delta$. For instance, if $d_t^i \in \{0, 1\}$, $\forall t$ and $Q(D_t) = \sum_{i=1}^N d_t^i$ is the count query (e.g., daily count of patients), then $\Delta = 1$. It is worth noting that our method and analysis can be easily generalized to the case when ΔQ_t also changes over time.

2.3 Minimum mean squared error estimate

The minimum mean squared error (MMSE) estimate of a random variable X given observation $Y = y$ is $\hat{X}(y) = \text{argmin}_h \, \mathbb{E}_X((X - h(Y))^2 | Y = y) = \mathbb{E}(X|Y = y)$. If $h(\cdot)$ is constrained to be linear, i.e., $h(Y) = k_1 Y + k_2$, then the corresponding minimization leads to the linear MMSE (LMMSE) estimate and is given by $\hat{X}(y) = \rho_{XY} \frac{\sigma_X}{\sigma_Y}(y - \mathbb{E}(Y)) + \mathbb{E}(X)$ with a mean squared error (MSE) $= (1 - \rho_{XY}^2)\sigma_X^2$, where ρ_{XY} is the correlation coefficient of X and Y, σ_X^2, σ_Y^2 the variance of X, Y respectively. Using these properties, we have the following result.

Proposition 2.4. *Consider a Gaussian AR(1) process $Z_{1:T}$ defined by (1), the MMSE estimate of Z_{t+1} given $Z_t = z_t$ is $\hat{Z}_{t+1}(z_t) = \mu(1 - \rho) + \rho z_t$, with MSE $\sigma_z^2(1 - \rho^2)$. If we use a perturbed $X_i = Z_i + N_i$, $i \in \{1, \cdots, t\}$, to estimate Z_{t+1},*

[2] If we express D in matrix form, i.e., $D \in \mathbb{R}^{N \times T}$ with $D_{it} = d_t^i$, then D and $\widehat{D}$ are different in at most one row, i.e., the hamming distance between two matrices is at most 1.



(a) Baseline approach (b) Proposed approach

Fig. 1. Comparison of two data release methods: $\{z_t\}_{t=1}^T$ is the true sequence, $\{x_t\}_{t=1}^T$ the released private sequence, $\widehat{Z}_t$ the estimate of z_t learned from x_{t-1}, and $\{n_t\}_{t=1}^T$ the added noise.

where $N_i \sim \mathcal{N}(0, \sigma_n^2)$ is the added noise, then the MMSE estimate of Z_{t+1} given $X_i = x_i$ is

$$\hat{Z}_{t+1}(x_i) = \mu(1 - \rho^{t+1-i}\frac{\sigma_z^2}{\sigma_z^2 + \sigma_n^2}) + \rho^{t+1-i}\frac{\sigma_z^2}{\sigma_z^2 + \sigma_n^2}x_i \ .$$

The (L)MMSE estimates for a Binomial AR(1) process are given in Appendix A.2. Note that for Gaussian AR(1) processes, both MMSE estimates $\hat{Z}_{t+1}(z_t)$, $\hat{Z}_{t+1}(x_i)$ are linear. For Binomial AR(1), the MMSE estimate $\hat{Z}_{t+1}(z_t)$ is also linear, which may not hold for other AR(1) processes. However, due to the simple form of linear MMSE estimate and its applicability to more general random processes, we will solely focus on LMMSE estimates in this study.

3 BASELINE APPROACH

The baseline approach (Figure 1(a)) provides differential privacy for a sequence $z_{1:T}$ by perturbing each z_t directly: $x_t = z_t + perturbation$. The upper bound of the total privacy loss, ϵ_T, can be characterized as a log-likelihood ratio of the released output under two sequences, which can be decomposed as follows:

$$\log \frac{\mathcal{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|z_{1:T})}{\mathcal{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|\hat{z}_{1:T})} = \sum_{t=1}^{T} \log \frac{\mathcal{F}_{X_t|Z_t}(x_t|z_t)}{\mathcal{F}_{X_t|Z_t}(x_t|\hat{z}_t)},$$

where the term $\log \frac{\mathcal{F}_{X_t|Z_t}(x_t|z_t)}{\mathcal{F}_{X_t|Z_t}(x_t|\hat{z}_t)}$ bounds the privacy loss at time t. As the total privacy loss accumulates over T time steps, balancing the privacy-accuracy tradeoff becomes more and more difficult as T increases. As long as the variance of perturbation is finite, as $T \to \infty$, ϵ_T inevitably approaches infinity.

We therefore propose a method that can (i) improve the privacy-accuracy tradeoff significantly, and (ii) bound the total privacy loss over an infinite horizon when the variance of perturbation is finite.

4 THE PROPOSED APPROACH

In our proposed method, data point x_t at time step t is released based on the previously released data x_{t-1} and its true value z_t (shown in Figure 1(b)).

The idea behind our approach is based on two observations: (1) Since x_{t-1} is correlated with z_t through z_{t-1}, we can use x_{t-1} to obtain an estimate[3] of z_t, denoted by $\hat{Z}_t(x_{t-1})$, and release (the perturbed version of) $\hat{Z}_t(x_{t-1})$ instead of z_t. (2) Since differential privacy is immune to post-processing [Dwork et al. 2014], using x_{t-1} to estimate z_t does not

[3]This estimate can be obtained with or without the knowledge of the statistics of the AR(1) process; in the absence of such knowledge one can employ a separate procedure to first estimate the statistics as detailed later in this section.

introduce additional privacy loss. Thus, technically we can release an initial x_1 (the perturbed version of z_1), followed by the sequence $x_t = \hat{Z}_t(x_{t-1}), t > 1$. However, doing so will lead to a fairly inaccurate released sequence compared to the original, for which the privacy loss does not accumulate over time but the estimation error does. To balance the competing needs of accuracy (having the released sequence resemble the true sequence) and privacy, one must calibrate the released version using the true values.

There are different ways to calibrate the released sequence. In this study, we shall examine the use of the convex combination $(1-w_t)\hat{Z}_t(x_{t-1}) + w_t z_t$, and the perturbed version of this as the released x_t. Examples of other approaches to calibrating released sequences are discussed in Section 7.

The weight parameter w_t serves four purposes:

- In addition to the perturbation $N_t \sim \mathcal{N}(0, \sigma_n^2)$, w_t can also be tuned to better balance the privacy-accuracy tradeoff: larger w_t results in a more accurate but less private sequence. In contrast, σ_n^2 is the only means of controlling this tradeoff in the baseline method.
- If MSE is the measure of accuracy, then w_t can also be used to balance the bias-variance tradeoff. For a deterministic sequence $z_{1:T}$ with estimator X_t at t, $Bias(X_t) = \mathbb{E}(X_t) - z_t$ and $MSE(X_t) = \mathbb{E}((X_t - z_t)^2)$. The bias and variance can be controlled jointly by adjusting w_t and σ_n^2, which can result in smaller MSE as $MSE = Variance + Bias^2$. In contrast, $x_{1:T}$ is always unbiased in the baseline method and $MSE = Variance$ always holds.
- If we keep w_t private, then the method can prevent certain attackers from knowing the detail of the perturbation mechanism, resulting in stronger protection (Section 7).
- By adjusting w_t, it is possible to release the sequence spanning an infinite horizon with bounded total privacy loss (Section 5).

Algorithm 1: Function `Est`

Input : $x_{1:t-1}$, $\mathrm{Var}(N_t)$

$\hat{\mu} = \frac{1}{t-1}\sum_{i=1}^{t-1} x_i$

$\hat{\sigma}^2 = \max\{\frac{1}{t-2}\sum_{i=1}^{t-1}(x_i - \hat{\mu})^2 - \mathrm{Var}(N_t), 0\}$

$\hat{\rho} = \frac{\sum_{i=1}^{t-2}(x_i-\hat{\mu})(x_{i+1}-\hat{\mu})}{\sum_{i=1}^{t-2}(x_i-\hat{\mu})^2} + \frac{1}{t-1}\mathbf{1}\{\rho^{true} > 0\}$[4]

Output : $\hat{\rho}, \hat{\mu}, \hat{\sigma}^2$

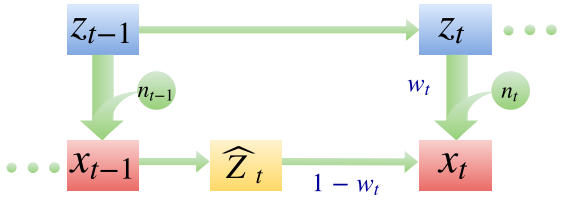


Fig. 2. A two-step illustration of the proposed method: adding noise n_t to the convex combination of estimate $\hat{Z}_t(x_{t-1})$ and true value z_t gives the released x_t.

4.1 Estimate of Z_t with learned correlation

We can estimate the true value Z_t from x_{t-1} using the LMMSE estimate $\hat{Z}_t(x_{t-1})$ given in Section 2. However, it requires the knowledge of mean μ, variance σ^2 and autocorrelation ρ of $Z_{1:T}$, which may be unknown in reality and should be estimated. To avoid revealing more information about $z_{1:T}$, this estimate is obtained using only the released $x_{1:T}$, as shown in Algorithm 1, where both $\hat{\mu}$ and $\hat{\sigma}^2$ are unbiased, and $\hat{\rho}$ is adopted from [Huitema and McKean 1991][5].

[5]The extra term $\frac{1}{t-1}$ is used for correcting the negative bias if there is prior knowledge of the positive autocorrelation $\rho^{true} > 0$.

4.2 Release x_t with estimate $\hat{Z}_t(x_{t-1})$ and true value z_t

Given the estimated parameters $\hat{\mu}, \hat{\sigma}^2$ and $\hat{\rho}$, using results presented in Section 2, the LMMSE estimate $\hat{Z}_t(x_{t-1})$ can be approximated as

$$\hat{Z}_t(x_{t-1}) = \hat{\mu}_{t-1}\left(1 - \hat{\rho}_{t-1}\frac{\hat{\sigma}^2_{t-1}}{\hat{\sigma}^2_{t-1} + \mathrm{Var}(N_t)}\right) + \hat{\rho}_{t-1}\frac{\hat{\sigma}^2_{t-1}}{\hat{\sigma}^2_{t-1} + \mathrm{Var}(N_t)}x_{t-1}\ .$$

Take the convex combination of estimate $\hat{Z}_t(x_{t-1})$ and true value z_t with private weight $w_t \in (0,1)$, and release:

$$x_t = (1-w_t)\hat{Z}_t(x_{t-1}) + w_t z_t + \textit{perturbation}\ .$$

Algorithm 2: Sequential Data Release Algorithm

Input : Sensitivity of query Δ, $\{\mathrm{Var}(N_t)\}_t$
for $t = 1, 2, \cdots, T$ **do**
  Input : true state z_t, weight w_t
  if $t \le 2$ **then**
    $w_t = 1$;
    Release : $x_t = z_t + n_t$.
  else
    $\hat{\rho}_{t-1}, \hat{\mu}_{t-1}, \hat{\sigma}^2_{t-1} = \mathtt{Est}(x_{1:t-1}, \mathrm{Var}(N_t))$;
    $r_t = \hat{\rho}_{t-1}\frac{\hat{\sigma}^2_{t-1}}{\hat{\sigma}^2_{t-1}+\mathrm{Var}(N_t)}$;
    Release : $x_t = (1-w_t)(\hat{\mu}_{t-1}(1-r_t) + r_t x_{t-1}) + w_t z_t + n_t$
Output : privacy parameter (ϵ_T, δ_T)

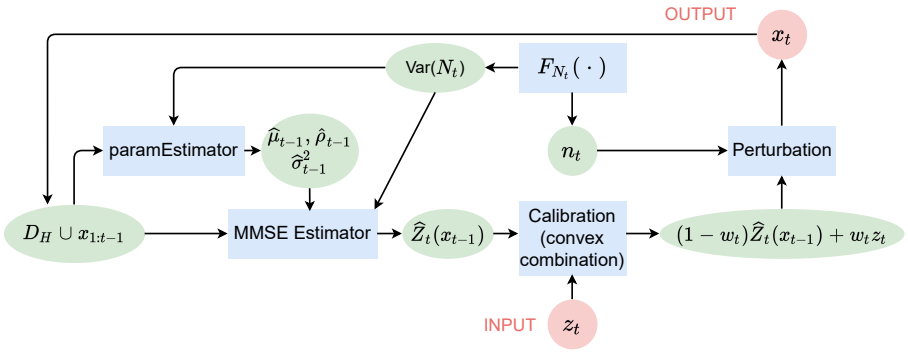


Fig. 3. flowchart of the complete procedure: at each time step t, true data point z_t is first estimated based on previous released data x_{t-1}; the estimation $\hat{Z}_t(x_{t-1})$ is then calibrated with true data z_t using convex combination, i.e., $(1-w_t)\hat{Z}_t(x_{t-1}) + w_t z_t$, $w_t \in (0,1)$; finally, the noisy version of the calibrated result (i.e., x_t) is released.

4.3 Privacy mechanism

The perturbation term in the released data adds privacy protection, and existing literature provides methods on how to generate them. We shall adopt the Gaussian mechanism [Dwork et al. 2014] and bound the privacy loss in terms of perturbation.

Lemma 4.1. *(Gaussian Mechanism) Consider query $Q : \mathcal{D} \to \mathbb{R}$ with sensitivity ΔQ, and the Gaussian mechanism $\mathcal{G}(d) = Q(d) + N$ which adds zero-mean Gaussian noise N with variance σ^2 to the output. If $\sigma \geq \frac{\Delta Q\sqrt{2\log(1.25/\delta)}}{\epsilon}$ for $\epsilon, \delta \in (0,1)$, then it satisfies (ϵ, δ)-differential privacy.*

We also propose a Binomial mechanism in Appendix A.3, which is a generalization (for arbitrary ΔQ) to the version (for the case $\Delta Q = 1$) first proposed in [Dwork et al. 2006]. It is more suitable for discrete settings and doesn't have a restriction on $\epsilon < 1$.

The complete procedure of our method is illustrated in Figure 3 and given in Algorithm 2, where n_t is a realization of Gaussian noise N_t (resp. Binomial noise defined in Appendix A.3) when adopting the Gaussian (resp. Binomial) mechanism. D_H in Figure 3 represents the history data that can be used for estimating parameters but won't be revealed during this time horizon.

Note that the Gaussian/Binomial mechanism only specifies the privacy parameters over one time step. In the next section we specify (ϵ_T, δ_T) over T steps.

5 PRIVACY ANALYSIS

Next, we bound the total privacy loss when $X_{1:T}$ is released using Algorithm 2. Since the total privacy loss is accumulated over T steps, various composition methods can be applied to calculate (ϵ_T, δ_T). We use the moments accountant method from [Abadi et al. 2016] when N_t is Gaussian; the corresponding result is given in Theorem 5.1. In Appendix A.4, we use the composition theorem from [Kairouz et al. 2017] when N_t is Binomial with the corresponding result given in Theorem A.5.

Theorem 5.1. *Let $Z_t = Q(D_t)$ and Δ be the sensitivity of Q, $\forall t$. Consider Algorithm 2 using zero-mean Gaussian noise with $Var(N_t) = \sigma_n^2$, $\forall t$, that takes sequence $z_{1:T}$ as input and outputs $x_{1:T}$. The following holds.*

(i) Given any $\epsilon_T \geq \frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^T w_t^2$, the algorithm satisfies (ϵ_T, δ_T)-differential privacy for

$$\delta_T = \exp\left(\left(\frac{\frac{\Delta^2}{\sigma_n^2}\sum_{t=1}^T w_t^2}{4} - \frac{\epsilon_T}{2}\right)\left(\frac{\epsilon_T}{\frac{\Delta^2}{\sigma_n^2}\sum_{t=1}^T w_t^2} - \frac{1}{2}\right)\right).$$

(ii) Given any $\delta_T \in (0,1)$, the algorithm satisfies (ϵ_T, δ_T)-differential privacy for

$$\epsilon_T = 2\sqrt{\frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^T w_t^2 \log(\frac{1}{\delta_T}) + \frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^T w_t^2}.$$

Theorem 5.1 says that if a sequence of noisy data is released following Algorithm 2 and the noise has variance σ_n^2, then with probability $1 - \delta_T$, the total amount of privacy loss incurred on each individual over T time steps is bounded by ϵ_T. Here $\frac{\sigma_n}{\Delta}$ represents the degree of perturbation and w_t is the weight on the true value. Smaller perturbation and larger weight result in higher privacy loss. Because of the mapping between σ_n^2 and (ϵ_T, δ_T), we have the following result.

Corollary 5.2. *Let $\{w_t\}_{t=1}^T$ be the weights used in generating $x_{1:T}$ in Algorithm 2. To satisfy (ϵ_T, δ_T)-differential privacy, the variance of the Gaussian noise should be:*

$$\sigma_n^2 \geq \frac{\Delta^2\sum_{t=1}^T w_t^2}{2\epsilon_T + 4\ln\frac{1}{\delta_T} - 4\sqrt{(\ln\frac{1}{\delta_T})^2 + \epsilon_T \ln\frac{1}{\delta_T}}}.$$

To guarantee (ϵ_T, δ_T)-differential privacy, the noise magnitude will depend on both w_t and Δ. Larger sensitivity means larger impact of each individual on the released information and thus requires more perturbation for privacy protection; larger weights mean higher reliance on the true value in the released information, thus more perturbation is needed.

Note that Algorithm 2 reduces to the baseline approach when $w_t = 1,\ \forall t$. Theorems 5.1, A.5 and Corollary 5.2 also hold for the baseline method if we set $w_t = 1,\ \forall t$. When the noise variance is finite, using the baseline method we have $\forall \delta_T$, $\epsilon_T \to \infty$ as $T \to \infty$. However, under the proposed method, it is possible that $\lim_{T\to\infty} \epsilon_T < \infty$ by controlling w_t, e.g., by taking $w_t^2 = ar^{t-1}, r \in (0,1)$ as a decreasing geometric sequence, we have $\lim_{T\to\infty} \sum_{t=1}^{T} w_t^2 = \lim_{T\to\infty} \sum_{t=1}^{T} ar^{t-1} = \frac{a}{1-r}$, which leads to a bounded ϵ_T even when $T \to \infty$ (Theorem 5.1).

6 ACCURACY ANALYSIS

In this section, we compare the accuracy of our method and the baseline method using the MSE measure, defined as $\mathbb{E}_{X_{1:T}}(||x_{1:T} - z_{1:T}||^2)$.

For simplicity of exposition, the analysis in this section is based on the assumption that the true values of parameters (ρ, μ, σ^2) of the underlying process are known. Additional error introduced by estimating parameters in Algorithm 1 is examined numerically in Section 8. In addition, we will only present the case of Gaussian AR(1) process and $\mathrm{Var}(N_t) = \sigma_n^2, \forall t$.

Theorem 6.1. *Let the sequence $z_{1:T}$ be generated by the Gaussian AR(1) process $Z_{1:T}$ with $Z_t \sim \mathcal{N}(\mu, \sigma_z^2)$ and $Corr(Z_t Z_{t-\tau}) = \rho^{|\tau|}, \forall t$. Let $x_{1:T}$ be the sequence released by Algorithm 2. Then $\mathbb{E}_{X_{1:T}}(||x_{1:T} - z_{1:T}||^2)$ is given by*

$$\underbrace{\sigma_z^2\left(1 - \rho^2 \frac{\sigma_z^2}{\sigma_z^2 + \sigma_n^2}\right) \sum_{t=1}^{T} (1 - w_t)^2}_{\text{estimation error}} + \underbrace{T\sigma_n^2}_{\text{perturbation error}} .$$

Theorem 6.1 suggests that the total error consists of two parts: (i) estimation error and (ii) perturbation error. For the former, a sequence with stronger autocorrelation (larger ρ) enables more accurate estimate, resulting in lower estimation error. Further, higher weight on the true value z_t (larger w_t), or less perturbation (smaller σ_n^2), also lowers the estimation error.

Theorem 6.2 below further compares the privacy-accuracy tradeoff of the two methods, where MSE is compared under the same privacy parameters (ϵ_T, δ_T).

Theorem 6.2. *Let sequential data $z_{1:T}$ be generated by the Gaussian AR(1) process $Z_{1:T}$ with $Z_t \sim \mathcal{N}(\mu, \sigma_z^2)$ and $Corr(Z_t Z_{t-\tau}) = \rho^{|\tau|}, \forall t$. Let $x_{1:T}^A$, $x_{1:T}^B$ be the sequences released by Algorithm 2 and the baseline method, respectively. Let $(\sigma_n^2)^A$, $(\sigma_n^2)^B$ be the corresponding noise variance. Suppose both outputs satisfy (ϵ_T, δ_T)-differential privacy, then*

$$\frac{T}{(\sigma_n^2)^B} = \frac{\sum_{t=1}^{T} w_t^2}{(\sigma_n^2)^A} = \frac{2\epsilon_T + 4\ln\frac{1}{\delta_T} - 4\sqrt{(\ln\frac{1}{\delta_T})^2 + \epsilon_T \ln\frac{1}{\delta_T}}}{\Delta^2} . \tag{2}$$

Furthermore, $\exists\ \{w_t\}_{t=1}^{T}, w_t \in (0,1)$ and $(\sigma_n^2)^A$, that satisfy Eqn. (2) and with which $x_{1:T}^A$ is more accurate than $x_{1:T}^B$.

Moreover, if a constant weight $w_t = w, \forall t$ is used, then $x_{1:T}^A$ is more accurate than $x_{1:T}^B$ if

$$w > \frac{1 - (\sigma_n^2)^B/\sigma_z^2}{1 + (\sigma_n^2)^B/\sigma_z^2} . \tag{3}$$

As mentioned earlier, when $w_t = 1, \forall t$, Algorithm 2 reduces to the baseline method, and $x^A_{1:T}$ and $x^B_{1:T}$ become equivalent. Theorem 6.2 shows that our method can *strictly* improve the privacy-accuracy tradeoff by controlling $w_t \in (0, 1)$. It also provides the guidance on how to select a constant weight $w_t = w, \forall t$, to guarantee this improvement from Eqn. (3): (i) If $(\sigma_n^2)^B > \sigma_z^2$, i.e., the privacy requirement is high and large perturbation is needed, then our method can always outperform the baseline regardless of the choice of $w \in (0, 1)$. In particular, if choosing $w \to 0$, our method will have large estimation error, but privacy can be provided with insignificant perturbation; the overall error is dominated by the estimation error, which is still smaller than the perturbation error in the baseline. (ii) If $(\sigma_n^2)^B < \sigma_z^2$, then w should be sufficiently large to maintain accuracy.

7 DISCUSSION

Generalization: The proposed method is not limited to AR(1) processes; it can be applied to any (weakly) stationary random process. This is because the LMMSE estimate only depends on the mean, variance and correlation of the random process. The methodologies used in Sections 5 and 6 are also not limited to AR(1) processes. For example, the bound in Theorem 5.1 is not limited to Gaussian AR(1). The error bound derived in Theorem 6.1 only depends on the MSE of $\hat{Z}_t(x_{t-1})$ at each time step (i.e., **term 3** in Appendix B.4). That is, this bound can be generalized to sequences following other random processes as long as we can quantify their MSE. In Section 8, the real-world datasets used in the experiments do not necessarily follow AR(1), but our method is shown to achieve better performance.

Robustness against certain attacks: Differential privacy is a strong privacy guarantee and a worst-case measure, as it bounds privacy loss over all possible outputs and inputs. In practice, how much information about $z_{1:T}$ can really be inferred by an attacker depends on how strong it is assumed to be. An attacker is able to infer more information with higher confidence if they know the exact perturbation mechanism used in generating $x_{1:T}$, i.e., $\Pr(X_t|Z_t)$. Specifically, real data sequence $z_{1:T}$ and observations $x_{1:T}$ form a Hidden Markov Model (HMM); if an attacker knows the transition probability $P(Z_{t+1}|Z_t)$ and emission probability $P(X_t|Z_t)$, then they can infer hidden states $z_{1:T}$ based on observations $x_{1:T}$ using dynamic programming such as the Viterbi algorithm [Forney 1973].

Therefore, if an attacker knows the noise distribution $\mathcal{N}(0, \sigma_n^2)$, then they will know $\Pr(X_t|Z_t)$ automatically with the baseline method, i.e., $X_t|Z_t \sim \mathcal{N}(Z_t, \sigma_n^2)$. However, with our method, $X_t|Z_t \sim \mathcal{N}(w_t Z_t + (1 - w_t)\hat{Z}_t(x_{t-1}), \sigma_n^2)$; thus if w_t is *private* and unknown to the attacker, then $\Pr(X_t|Z_t)$ cannot be readily inferred even when they know the noise distribution. As a result, in practice our method can prevent this class of attackers from knowing the details of the perturbation mechanism, thus can be stronger.

Impact of estimating parameters from a noisy sequence: The analysis in Section 6 shows that when the true parameters of the underlying process are known, our algorithm can always outperform the baseline method. However, these may be unknown in reality and need to be estimated from the released sequence using Algorithm 1, which leads to additional estimation error. Nevertheless, this can still outperform the baseline method. Consider the extreme case where $(\sigma_n^2)^A \to +\infty$. The LMMSE estimate from the noisy data $\hat{Z}_t(x_{t-1}) \to \mathbb{E}(Z_t) \approx \hat{\mu}_{t-1}$. Since the added noise is zero-mean, with enough released data $\hat{\mu}_{t-1}$ can attain sufficient accuracy. Then x_t determined by both $\hat{\mu}_{t-1}$ and true z_t before adding noise becomes a filtered version of the true sequence, and its accuracy after adding noise will still be higher than the baseline method under the same privacy measure; this point is further validated by experiments in Section 8.

Other approaches to calibrating released sequence: We have used the convex combination of estimate $\hat{Z}_t(x_{t-1})$ and true data z_t to calibrate the released data. This method is effective and easy to use and analyze. In particular, the weight in the convex combination provides an additional degree of freedom and serves four purposes (Section 4).

There are also other approaches to calibrating the released sequence. For example, we can leverage all released points to estimate new data, and use a sequence of estimates to calibrate, i.e., $\sum_{i=1}^{t-1} w_i \hat{Z}_t(x_i) + w_t z_t$. One could also use a non-linear combination to calibrate, e.g., $w_t z_t + (1 - w_t)\sqrt{z_t \hat{Z}_t(x_{t-1})}$.

8 EXPERIMENTS

In this section, we compare the privacy-accuracy tradeoff of our method with other methods using real-world datasets. Unless explicitly mentioned, fixed weights, $w_t = w, \forall t$, are used in the proposed method.

Methods: For comparison, in addition to the baseline method, we also consider the following.

- *Baseline-Laplace*: Laplace noise $n_t \sim Lap(0, \frac{T\Delta}{\epsilon_T})$ is added to z_t independently at each time step.
- *FAST without sampling* [Fan and Xiong 2014] [6]: Laplace noise $n_t \sim Lap(0, \frac{T\Delta}{\epsilon_T})$ is first added to z_t, then a posterior estimate of each z_t using the Kalman filter is released. Since it assumes the time series follows a random process $Z_{t+1} = Z_t + U_t$ with $U_t \sim \mathcal{N}(0, \sigma_u^2)$, to use the Kalman filter it requires σ_u^2 to be known in advance. Moreover, it also needs to use a Gaussian noise $\tilde{n}_t \sim \mathcal{N}(0, \sigma_{app}^2)$ to approximate the added Laplace noise n_t. In our experiments, σ_{app}^2 is chosen based on the guidelines provided in [Fan and Xiong 2014] and σ_u^2 that gives the best performance is selected using exhaustive search.
- *DFT* [Rastogi and Nath 2010]: Discrete Fourier Transform is applied to the entire sequence first, then among T Fourier coefficients $DFT(z_{1:T})_j = \sum_{i=1}^{T} \exp(\frac{2\pi\sqrt{-1}}{T} ji)x_i$, $j \in [T]$, it selects the top d and perturbs each of them using Laplace noise $\frac{\sqrt{dT}\Delta}{\epsilon_T}$. Lastly, it pads $T - d$ 0's to this perturbed coefficients vector and applies Inverse Discrete Fourier Transform. In our experiments, d that gives the best performance is selected from $\{1, \cdots, T\}$ using exhaustive search.
- *BA* and *BD* [Kellaris et al. 2014]: Two privacy budget allocation mechanisms, Budget Distribution (BD) & Budget Absorption (BA), are used to dynamically allocate privacy budget over time based on the dissimilarity between the previously released data and the new data. The new private data is released at each time step only when the data is sufficiently different from the previously released data; otherwise, the previous data is recycled and released again. The idea is to improve accuracy by allocating more privacy budgets to the most important data points.
- *Binary tree mechanism* [Chan et al. 2011]: binary tree mechanism at each time outputs the approximate sum of the data points seen so far, i.e., $B_t = \sum_{i=1}^{t} z_i +$ noise, while preserving **event-level** privacy[7]. The idea is to first internally group the data arrived so far based on the binary representation of current time t, the partial sum (*p-sum*) of data within each group can be computed and perturbed. Finally, sum over all these noisy *p-sum*'s gives the result. To compare with our method, we generate sequence $b_{1:T}$ based on the summations released by binary tree mechanism $B_{1:T}$, i.e., $\forall t, b_t = B_t - B_{t-1}$. Then the performance of sequence $b_{1:T}$ is compared with the sequence released by our method $x_{1:T}$.

Real-world Datasets: We use the following datesets in our experiments.

[6] FAST samples $k < T$ points and allocates privacy budget ϵ_T to the sampled points. It adds Laplace noise $Lap(0, \frac{K\Delta}{\epsilon_T})$ to each sampled point and outputs the corresponding a posterior estimate, while for non-sampled points it outputs prior estimates. A similar sampling procedure can be added to our proposed method where we set $w_t = 0$ for non-sampled points.

[7] *Event-level* privacy only guarantees the privacy at one time step; it requires sequence pairs $z_{1:T}, \widehat{z}_{1:T}$ defined in differential privacy (Definition 2.2) to be different in at most one data point. In contrast, our work considers *user-level* privacy where privacy at all times is ensured.

- *Ride-sharing counts* [Fanaee-T and Gama 2013]: this is generated using historical log from Capital Bikeshare system in 2011. It includes the counts of rented bikes aggregated on both an hourly and daily basis. Because each data point is a count over a dataset, query sensivity $\Delta = 1$.
- *NY traffic volume counts in 2011* [DOT 2011]: this is collected by the Department of Transportation (DOT). It contains the counts of traffic in various roadways from 12AM to 1PM on an hourly basis each day. We aggregate the counts from all roadways and concatenate sequences from different days in chronological order. Because each data point is a count over a dataset, query sensivity $\Delta = 1$.
- *Federal Test Procedure (FTP) drive cycle* [EPA 2008]: this dataset includes a speed profile for vehicles and simulates urban driving patterns. It can be used for emission certification and fuel economy testing of vehicles in the United States. Because each data point is the speed of one vehicle, query sensitivity Δ is the range of the vehicle's speed. In this setting, D_t only includes one data point and the definition of differential privacy (Definition 2.2) is reduced to that of local differential privacy [Kasiviswanathan et al. 2011].

Accuracy metric: We use *relative error (RE)* defined as the normalized MSE to measure the accuracy of $x_{1:T}$:

$$RE(z_{1:T}, x_{1:T}) = \frac{1}{T} \frac{||z_{1:T} - x_{1:T}||_2}{\max_{1 \le t \le T} |z_t|} .$$

8.1 Comparison with other methods

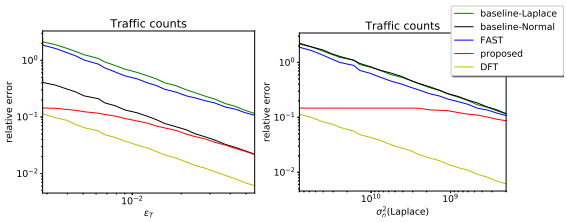


Fig. 4. Comparison of different methods

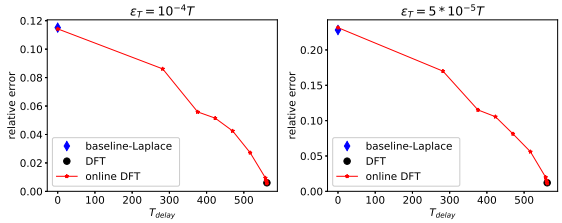


Fig. 5. Comparison with Online DFT

The comparison results are shown in Figure 4, where we use $\delta_T = 10^{-7}$ in baseline-Normal and the proposed method; $\Delta = 1$ as each data point z_t is a count over a dataset. The left plot compares the relative error achieved by different methods under the same ϵ_T. However, the baseline-Laplace, FAST and DFT methods satisfy $(\epsilon_T, 0)$-differential privacy while the baseline-Normal and proposed methods satisfy $(\epsilon_T, 10^{-7})$-differential privacy. Although $\delta_T = 10^{-7}$ appears small, the total privacy loss ϵ_T under these methods are calculated using different composition methods. Comparing different methods solely based on ϵ_T may not be appropriate as the improvement in ϵ_T may come from the composition strategy rather than the algorithm itself.

To address this issue, we add the right plot in Figure 4, where the noises in baseline-Laplace and baseline-Normal are chosen such that the error achieved by baseline-Normal is no less than that under baseline-Laplace, i.e., the black curve is slightly over the green curve in the plot. This would guarantee that baseline-Normal provides stronger privacy than baseline-Laplace. By further controlling the proposed method to have the same privacy as baseline-Normal (noise variances in two methods satisfy Eqn. (2)), and FAST and DFT to have the same privacy as baseline-Laplace, we can guarantee that the proposed method is at least as private as FAST and DFT. In the plot, the x-axis denotes the variance of added noise in baseline-Laplace and the noise parameters of the other methods are selected accordingly. It shows that the proposed method outperforms FAST; the improvement is more significant when the privacy requirement is

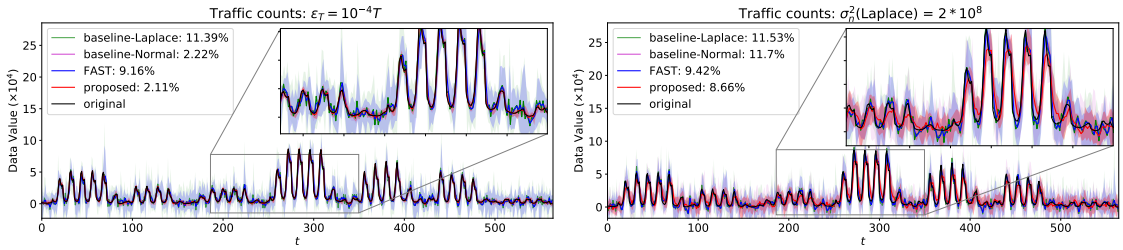


Fig. 6. Sequences aggregated from 10 runs of experiments using different methods under the same ϵ_T (upper plot). In the lower plot, noise variance is selected in each method such that the proposed method and baseline-Normal are at least as private as FAST and baseline-Laplace.

high. While generally DFT performs better than the proposed method, it is an offline method which requires the entire sequence to be known a priori. However, as perturbation increases (more private), the proposed method can achieve similar performance as DFT.

The DFT method can also be adapted online. One way to do this is to perform DFT over a subsequence of length $T_{delay} \ll T$ (data released with delay T_{delay}). We examine the performance of such a method on the Traffic dataset by comparing it with DFT and baseline-Laplace. Figure 5 shows that when $T_{delay} = 0$ (data released in real-time, DFT applied to one data point each time and on one coefficient), the performance is similar to baseline-Laplace; as T_{delay} increases, its accuracy increases at the expense of increased delay. Because the performance of the proposed algorithm falls between baseline-Laplace and offline DFT (Figure 4), there exists a delay $0 < T_{delay} < T$ such that the sequence released using DFT with delay T_{delay} and the proposed method have similar performance.

Figure 6 shows the private traffic counts generated using various methods. For each method, we repeat the experiment 10 times and obtain 10 sample paths $\{x_{1:T}\}_{k=1}^{10}$. The curves in the plot show the average $\frac{1}{10}\sum_{k=1}^{10} x_{1:T}^k$ while the shaded area indicates their variance whose upper and lower bound at each t are $\max_k x_t^k$ and $\min_k x_t^k$, respectively.

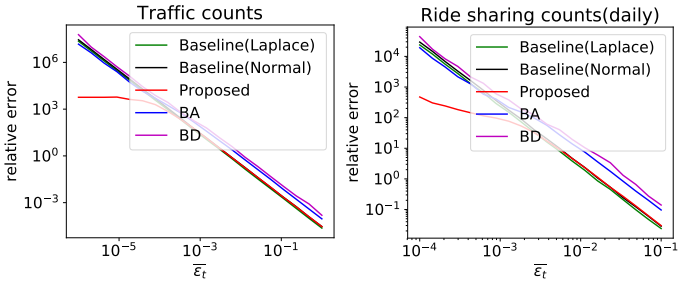


Fig. 7. Comparison with BA and BD [Kellaris et al. 2014].

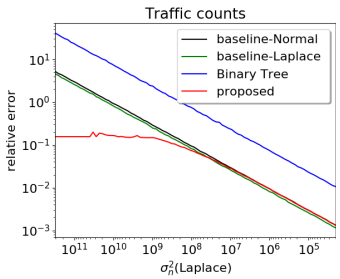


Fig. 8. Comparison with Binary Tree Mechanism

We also compare our proposed method with BA and BD proposed in [Kellaris et al. 2014]. Unlike our model, where a single query is released at every time step, BA and BD are designed to release a vector of length d each time. Moreover, BA and BD adopt $(\epsilon, 0)$-differential privacy. In order to compare with our method, we set $d = 1$ and use baseline-Laplace and baseline-Normal as two baselines. Specifically, we choose noises for different methods such that: (1) our proposed method and baseline-Normal have the same privacy guarantee; (2) BA, BD, and baseline-Laplace have the same privacy guarantee; and (3) baseline-Normal is at least as private as baseline-Laplace. The results are shown in Figure 7, where

the y-axis indicates the averaged relative error of 10 independent runs of experiment and x-axis is the privacy loss per time step under baseline-Laplace. As illustrated, our method outperforms others. It is worth noting that BA and BD may not even outperform baseline-Laplace. This is because in both BA and BD, half of the privacy budget is assigned to measure the dissimilarity between previously released data and new data; thus only half of the privacy budget is left for releasing the sequence. Moreover, as mentioned, BA and BD are meant for releasing a vector, especially when d is large; the error of the released sequence can be large when d is small (Theorems 6 and 7 in [Kellaris et al. 2014]). It further suggests that in settings where only a single query is released ($d = 1$), BA and BD may not be suitable.

We then compare our method with the binary tree (BT) mechanism proposed in [Chan et al. 2011]. Figure 8 compares the performance of different algorithms on Traffic volume counts data, similar results are observed for other datasets. Since the BT mechanism adopts $(\epsilon, 0)$-differential privacy, we use the same strategy above and take baseline-Laplace and baseline-Normal as two baselines, and choose noises such that the proposed method is at least as private as the BT mechanism. It shows that ours is significantly better than BT, and the performance of BT is even worse than baseline-Laplace. This is mainly because the BT mechanism focuses on a different setting: (1) it is proposed for releasing the sum of data points seen so far, i.e., $\sum_{i=1}^{t} z_i$, which means that the data point z_t needs to be repeatedly queried $\forall t' \geq t$, and (2) the BT mechanism aims at controlling event-level privacy (which our comparison is on user-level privacy), which generally requires more perturbation depending on how events are defined.

8.2 Impact of parameters ρ and w_t

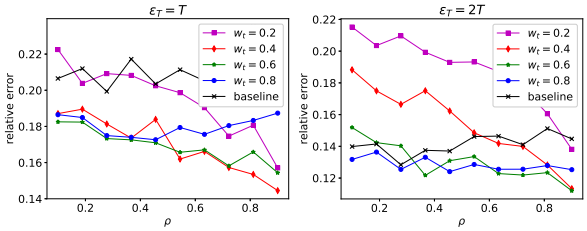


Fig. 9. Impact of correlation on performance

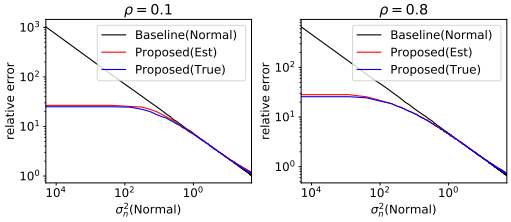
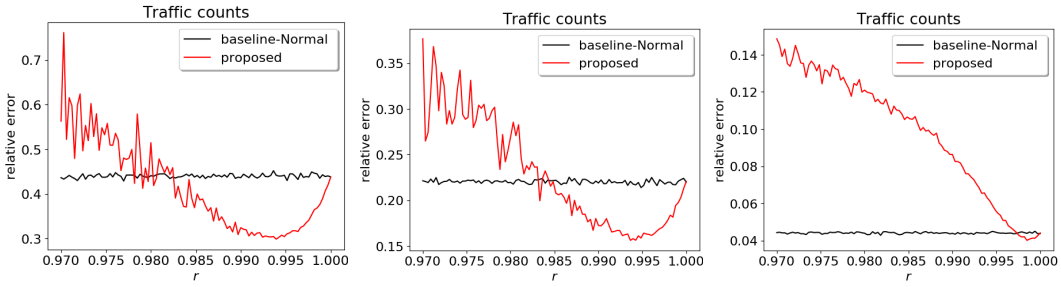


Fig. 10. Impact of estimation from noisy sequence: $Z_{1:T}$ satisfies $Z_{t+1} = \rho Z_t + U_t$ with $U_t \sim \mathcal{N}(0, 10)$, $Z_0 = 0$ and weak ($\rho = 0.1$) or strong ($\rho = 0.8$) autocorrelation.

As mentioned earlier, the baseline is a special case ($w_t = 1, \forall t$) of our method, which can always outperform the former with better tuned weights. The achievable improvement depends on the correlation of the sequence. We show this in Figure 9, the error of various synthetic sequences using different weights under the same privacy ϵ_T. Each sequence follows Gaussian AR(1) with $Z_t \sim \mathcal{N}(0, 1)$ but the correlation ρ varies from 0.1 to 0.9. It shows that (i) in all cases, one can find weights for our method to outperform the baseline; sequences with high ρ have the highest accuracy under the same ϵ_T; (ii) with weak (resp. strong) privacy as shown on the right (resp. left), the smallest weights that can give improvement are close to 1 (resp. 0) and the achievable improvement is small (resp. large) as compared to the baseline. As released data depends less (resp. more) on estimates when weights are large (resp. small), the correlation within the sequence does not (resp. does) affect performance significantly. In the right (resp. left) plot with weak (resp. strong) privacy, curves with lowest error are similar under different ρ (resp. decreases in ρ).

We also examine the impact of estimating parameters from a noisy sequence; the result is shown in Figure 10, where Gaussian AR(1) sequences are generated. Red curves represent the relative error achieved using the proposed method



(a) $\epsilon_T = 0.5 * 10^{-5} \times T$ (b) $\epsilon_T = 10^{-5} \times T$ (c) $\epsilon_T = 0.5 * 10^{-4} \times T$

Fig. 11. Performance of the proposed method under a geometric sequence of weights $w_t^2 = r^{t-1}, r \in (0.97, 1)$. It shows that for any privacy loss ϵ_T, always there exists a lower bound $r_o \in (0, 1)$ such that $\forall r \geq r_o$, the proposed algorithm with weights $w_t^2 = r^{t-1}$ outperforms the baseline method.

where $\hat{\mu}_t$, $\hat{\sigma}_t^2$ and $\hat{\rho}_t$ at each time are estimated from the previous released sequence; blue curves represent the case where we use true parameters μ, σ^2, ρ to estimate z_t using x_{t-1}. As expected, estimating parameters from a noisy sequence degrades the performance. However, even with this impact the proposed method continues to outperform the baseline significantly.

As mentioned in Section 5, the proposed method can attain a bounded total privacy loss ($\lim_{T\to\infty} \epsilon_T < \infty$) by taking weights $w_t^2 = ar^{t-1}, r \in (0, 1)$ as a decreasing geometric sequence. In Figure 11, we examine the performance of our algorithm on traffic count dataset when weights $w_t^2 = r^{t-1}, r \in (0, 1)$. Specifically, each figure shows the relative error of the proposed method (red curves) and baseline (black curves) as functions of $r \in (0.97, 1)$ under a certain privacy loss ϵ_T. Because r doesn't impact the baseline method, the error of baseline remains the same (the oscillation in the plot comes from the randomness). The results show that for any privacy loss ϵ_T, always there exists a lower bound $r_o \in (0, 1)$ such that $\forall r \geq r_o$, the proposed algorithm with weights $w_t^2 = r^{t-1}$ outperforms the baseline method. Moreover, as privacy guarantee gets weaker (i.e., ϵ_T increases), the lower bound r_o that leads to the improvement also increases. This is consistent with Theorem 6.2.

8.3 Queries outside of count/stationary queries

As discussed in Section 7, the proposed method is not limited to AR(1) processes but is applicable to any (weakly) stationary random process. In fact, the empirical results further show that the proposed method works well even for *non-stationary* sequences. In particular, real-world datasets we considered such as ride sharing counts (daily), NY traffic volume counts, FTP drive cycle are non-stationary, i.e., the mean/variance of data changes over time, the correlation between values at any two time steps depend not only on their time difference, but also on the particular time step. Next, we further demonstrate this on synthetic non-stationary data sequences (Figure 12).

Specifically, the synthetic data is generated based on the following:

$$Z_t = \rho Z_{t-1} + U_t + \eta \cdot t, \ \forall t \geq 1 \ . \tag{4}$$

We adopt Gaussian distributed $U_t \sim \mathcal{N}(0, \sigma_u^2) = \mathcal{N}(0, 10)$ and $\rho = 0.8$. When $\eta = 0$, Eqn. (4) reduces to Gaussian AR(1) process in (1). Term $\eta \cdot t$ in Eqn. (4) is added for interrupting stationarity: as $\eta \geq 0$ increases, the interruption is more severe. Figure 12 illustrates the sample paths of $Z_{1:T}$ under different values of η, where each Z_t is Gaussian distributed

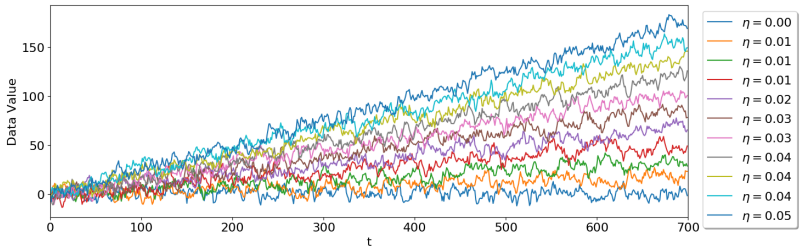
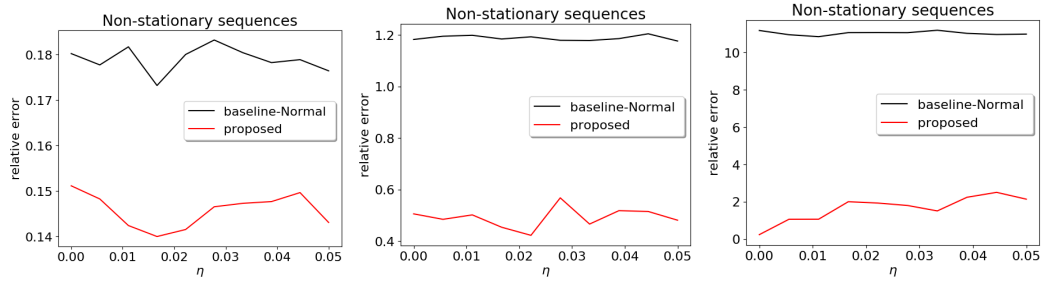


Fig. 12. Sample paths of synthetic non-stationary data sequence: $Z_{1:T}$ satisfies $Z_{t+1} = \rho Z_t + U_t + \eta \cdot t$ with $U_t \sim \mathcal{N}(0, 10)$, $Z_0 = 0$ and $\rho = 0.8$. $\eta \in [0, 0.05]$ controls the degree of non-stationarity

with mean $\mu \cdot t$ and variance $\sigma_z^2 = \frac{\sigma_u^2}{1-\rho^2} \approx 27.8$. Note that for these sequences, the sensitivity Δ_t is same over time and we set $\Delta_t = 1$ in the experiment. To examine the impact of non-stationarity, we adjust the accuracy metric and define relative error as

$$RE(z_{1:T}, x_{1:T}) = \frac{1}{T} \frac{||z_{1:T} - x_{1:T}||_2}{3\sigma_z} .$$

Figures 13 compares the accuracy of our method with baseline-Normal under the same privacy guarantee. We observe that our method always outperforms the baseline significantly, and the impact of non-stationarity is minor; it is more significant when the privacy guarantee is more restrictive: when $\epsilon_T = 10^{-3} \times T$, the performance of our method deteriorates as the data becomes more non-stationary.



(a) $\epsilon_T = 10^{-1} \times T$ (b) $\epsilon_T = 10^{-2} \times T$ (c) $\epsilon_T = 10^{-3} \times T$

Fig. 13. Impact of non-stationarity: $Z_{1:T}$ satisfies $Z_{t+1} = \rho Z_t + U_t + \eta \cdot t$ with $U_t \sim \mathcal{N}(0, 10)$, $Z_0 = 0$ and $\rho = 0.8$. $\eta \in [0, 0.05]$ controls the degree of non-stationarity. In each plot, the accuracy of the proposed method and baseline-Normal are compared under the same privacy guarantee. Results show that the impact of non-stationarity is minor; it is more significant when the privacy guarantee is more restrictive: when $\epsilon_T = 10^{-3} \times T$, the performance of our method deteriorates as the data becomes more non-stationary.

The proposed method is not limited to count queries and is more broadly applicable. For example, this method can be used in intelligent transportation systems to enable *private* vehicle-to-vehicle communication. In our studies [Huang et al. 2020; Zhang et al. 2019], a predictive cruise controller is designed for a follower vehicle using a private speed profile transmitted from its leader vehicle. Specifically, instead of broadcasting the real speed profile (FTP drive cycle), the leader vehicle generates a differentially private speed profile using the proposed method. A follower vehicle then designs an optimal speed planner based on the received information. Within this application context, query $Q(v)$ represents the vehicle's speed information, and sensitivity Δ is the range of the vehicle's speed. Figure 14 shows the

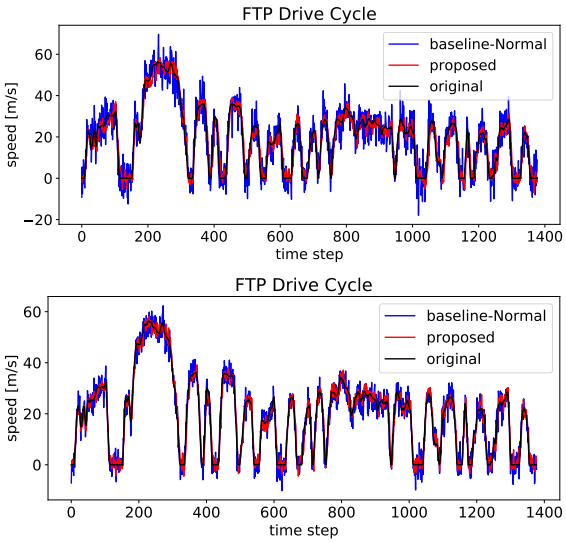


Fig. 14. Drive cycles under different levels of privacy: the privacy guarantee in the upper plot is stronger than that of the lower plot.

private speed profiles generated using the proposed method and baseline-Normal. A private optimal speed planner is designed in [Huang et al. 2020; Zhang et al. 2019] using these private profiles. The results show that the controller performance deteriorates significantly under the baseline method. In contrast, the controller designed with the proposed method can attain an accuracy that is sufficient for predictive control purposes. We refer an interested reader to [Huang et al. 2020; Zhang et al. 2019] for more details and the performance of the private controller.

9 CONCLUSION

This paper presented a method for releasing a differentially private data sequence in real time. It estimates the unreleased data from those already released based on their correlation, which is learned on the fly during the release process. This estimate along with the actual data is then released as a convex combination with added perturbation. This is shown to achieve higher accuracy with lower privacy loss compared to various existing approaches.

ACKNOWLEDGMENTS

This work was supported by the NSF under grants CNS-1646019, CNS-1739517, IIS-2040800 and IIS-2112471.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

Yang Cao and Masatoshi Yoshikawa. 2015. Differentially private real-time data release over infinite trajectory streams. In *2015 16th IEEE International Conference on Mobile Data Management*, Vol. 2. IEEE, 68–73.

Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. 2019. Quantifying Differential Privacy in Continuous Data Release Under Temporal Correlations. *IEEE Transactions on Knowledge and Data Engineering* 31, 7 (2019), 1281–1295. https://doi.org/10.1109/TKDE.2018.2824328

T-H Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)* 14, 3 (2011), 1–24.

Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 638–649.

Rui Chen, Benjamin Fung, and Bipin C Desai. 2011. Differentially private trajectory data publication. *arXiv preprint arXiv:1112.2020* (2011).

DOT. 2011. traffic volume counts, NYC OpenData. https://data.cityofnewyork.us/Transportation/Traffic-Volume-Counts-2011-2012-/wng2-85mv.

Cynthia Dwork. 2006. Differential Privacy, In 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006). 4052, 1–12. https://www.microsoft.com/en-us/research/publication/differential-privacy/

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 486–503.

Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 715–724.

Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

EPA. 2008. EPA Federal Test Procedure (FTP). https://www.epa.gov/emission-standards-reference-guide/epa-federal-test-procedure-ftp.

Liyue Fan and Li Xiong. 2014. An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Transactions on Knowledge and Data Engineering* 26, 9 (2014), 2094–2106.

Hadi Fanaee-T and Joao Gama. 2013. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence* (2013), 1–15. https://doi.org/10.1007/s13748-013-0040-3

Ferdinando Fioretto and Pascal Van Hentenryck. 2019. OptStream: Releasing Time Series Privately. *Journal of Artificial Intelligence Research* 65 (2019), 423–456.

G.D. Forney. 1973. The viterbi algorithm. *Proc. IEEE* 61, 3 (1973), 268–278. https://doi.org/10.1109/PROC.1973.9030

Arik Friedman and Assaf Schuster. 2010. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 493–502.

Soheila Ghane, Lars Kulik, and Kotagiri Ramamohanarao. 2019. TGM: A generative mechanism for publishing trajectories with differential privacy. *IEEE Internet of Things Journal* 7, 4 (2019), 2611–2621.

Mehmet Emre Gursoy, Ling Liu, Stacey Truex, and Lei Yu. 2018. Differentially private and utility preserving publication of trajectory data. *IEEE Transactions on Mobile Computing* 18, 10 (2018), 2315–2329.

Jingyu Hua, Yue Gao, and Sheng Zhong. 2015. Differentially private publication of general time-serial trajectory data. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 549–557. https://doi.org/10.1109/INFOCOM.2015.7218422

Chunan Huang, Xueru Zhang, Rasoul Salehi, Tulga Ersal, and Anna G. Stefanopoulou. 2020. A Robust Energy and Emissions Conscious Cruise Controller for Connected Vehicles with Privacy Considerations. In *2020 American Control Conference (ACC)*. 4881–4886. https://doi.org/10.23919/ACC45564.2020.9147406

Bradley E Huitema and Joseph W McKean. 1991. Autocorrelation estimation and inference with small samples. *Psychological Bulletin* 110, 2 (1991), 291.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2017. The composition theorem for differential privacy. *IEEE Transactions on Information Theory* 63, 6 (2017), 4037–4049.

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.

Georgios Kellaris and Stavros Papadopoulos. 2013. Practical differential privacy via grouping and smoothing. In *Proceedings of the VLDB Endowment*, Vol. 6. VLDB Endowment, 301–312.

Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. 2014. Differentially private event sequences over infinite streams. *Proceedings of the VLDB Endowment* 7, 12 (2014), 1155–1166.

Mohammad Mahdi Khalili, Xueru Zhang, and Mahed Abroshan. 2021b. Fair Sequential Selection Using Supervised Learning Models. *Advances in Neural Information Processing Systems* 34 (2021).

Mohammad Mahdi Khalili, Xueru Zhang, Mahed Abroshan, and Somayeh Sojoudi. 2021a. Improving fairness and privacy in selection problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 8092–8100.

Mohammad Mahdi Khalili, Xueru Zhang, and Mingyan Liu. 2019. Contract design for purchasing private data using a biased differentially private algorithm. In *Proceedings of the 14th Workshop on the Economics of Networks, Systems and Computation*. 1–6.

Mohammad Mahdi Khalili, Xueru Zhang, and Mingyan Liu. 2021c. Designing Contracts for Trading Private and Heterogeneous Data Using a Biased Differentially Private Algorithm. *IEEE Access* 9 (2021), 70732–70745.

Chao Li, Daniel Yang Li, Gerome Miklau, and Dan Suciu. 2014. A theory of pricing private data. *ACM Transactions on Database Systems (TODS)* 39, 4 (2014), 1–28.

Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. 2016. Dependence Makes You Vulnberable: Differential Privacy Under Dependent Tuples.. In *NDSS*, Vol. 16. 21–24.

Ed McKenzie. 1985. Some simple models for discrete variate time series. *JAWRA Journal of the American Water Resources Association* 21, 4 (1985), 645–650.

Victor Perrier, Hassan Jameel Asghar, and Dali Kaafar. 2018. Private continual release of real-valued data streams. *arXiv preprint arXiv:1811.03197* (2018).

Vibhor Rastogi and Suman Nath. 2010. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 735–746.

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

Christine Task and Chris Clifton. 2012. A guide to differential privacy theory in social network analysis. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 411–417.

Hao Wang, Zhengquan Xu, Lizhi Xiong, and Tao Wang. 2017. Conducting Correlated Laplace Mechanism for Differential Privacy. In *Cloud Computing and Security*, Xingming Sun, Han-Chieh Chao, Xingang You, and Elisa Bertino (Eds.). Springer International Publishing, Cham, 72–85.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469.

William WS Wei. 2006. Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*.

Christian H Weiß. 2009. Monitoring correlated processes with binomial marginals. *Journal of Applied Statistics* 36, 4 (2009), 399–414.

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*. ACM, 1298–1309.

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

Xueru Zhang, Chunan Huang, Mingyan Liu, Anna Stefanopoulou, and Tulga Ersal. 2019. Predictive Cruise Control with Private Vehicle-to-Vehicle Communication for Improving Fuel Consumption and Emissions. *IEEE Communications Magazine* 57, 10 (2019), 91–97. https://doi.org/10.1109/MCOM.001.1900146

Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. 2018a. Improving the privacy and accuracy of ADMM-based distributed algorithms. In *International Conference on Machine Learning*. PMLR, 5796–5805.

Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. 2018b. Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 959–965.

Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. 2020. Recycled ADMM: Improving the Privacy and Accuracy of Distributed Algorithms. *IEEE Transactions on Information Forensics and Security* 15 (2020), 1723–1734. https://doi.org/10.1109/TIFS.2019.2947867

Xiao Zheng. 2020. Data trading with differential privacy in data market. In *Proceedings of 2020 the 6th International Conference on Computing and Data Engineering*. 112–115.

Tianqing Zhu, Ping Xiong, Gang Li, and Wanlei Zhou. 2015. Correlated Differential Privacy: Hiding Information in Non-IID Data Set. *IEEE Transactions on Information Forensics and Security* 10, 2 (2015), 229–242. https://doi.org/10.1109/TIFS.2014.2368363

APPENDIX

A RESULTS OF BINOMIAL AR(1) PROCESS

A.1 Definition

Definition A.1. (**Binomial AR(1) process**) Let $\pi \in (0,1)$ and $\rho \in [\max(-\frac{\pi}{1-\pi}, -\frac{1-\pi}{\pi}), 1]$. Define $\beta = \pi(1-\rho)$, $\alpha = \beta + \rho$, and fix $n \in \mathbb{N}$. Then $Z_{1:T}$ is a binomial AR(1) process [McKenzie 1985] if:

$$Z_t = \alpha \circ Z_{t-1} + \beta \circ (n - Z_{t-1}),\ t \geq 1 \tag{5}$$

where $Z_0 \sim Binomial(n, \pi)$ and "$\circ$" is called the thinning operator defined as $a \circ Z_{t-1} = \sum_{i=1}^{Z_{t-1}} Y_{i,t-1}$, where $Y_{i,t-1}, i = 1, \cdots, Z_{t-1}$ are i.i.d Bernoulli random variables with $\Pr(Y_{i,t-1} = 1) = a$, and all thinnings are independent of each other. Binomial AR(1) is also a stationary Markov process with the following properties: (1) $Z_t \sim Binomial(n, \pi)$; (2) its autocorrelation is $\text{Corr}(Z_t Z_{t-\tau}) = \text{Corr}(\tau) = \rho^{|\tau|}$.

Binomial AR(1) is typically used for modeling integer-valued counts sequences. Consider n independent entities, each of which can be either in state "1" or state "0". Then Z_t can be interpreted as the number of entities in state "1" at time t. Eqn. (5) implies that this "1"-entity count (Z_t) can be given by the number of "1"-entities in the previous time instant that didn't change state ($\alpha \circ Z_{t-1}$) plus the number of "0"-entities in the previous time instant that changed to state "1" ($\beta \circ (n - Z_{t-1})$); here α, β can be interpreted as the respective transition probabilities. Binomial AR(1) has been used to model many real-world scenarios such as counts of computer log-ins and log-outs [Weiß 2009], daily counts of occupied rooms in a hotel, etc.

A.2 MMSE estimates

Proposition A.2. *Consider a Binomial AR(1) process $Z_{1:T}$ defined by (5), the MMSE estimate of Z_{t+1} given $Z_t = z_t$ is $\hat{Z}_{t+1}(z_t) = \rho z_t + n\pi(1-\rho)$.If we use a perturbed $X_i = Z_i + N_i,\ i \in \{1, \cdots, t\}$ with $Var(N_t) = \frac{m}{2}$ to estimate Z_{t+1}, then the LMMSE estimate of Z_{t+1} given $X_i = x_i$ is*

$$\hat{Z}_{t+1}(x_i) \quad = \quad n\pi(1-\rho^{t+1-i}\frac{n\pi(1-\pi)}{n\pi(1-\pi)+\frac{m}{2}}) + \rho^{t+1-i}\frac{n\pi(1-\pi)}{n\pi(1-\pi)+\frac{m}{2}}x_i$$

A.3 Binomial Mechanism

Definition A.3. (Binomial noise) We call random variable N the binomial noise if it is zero mean and follows the shifted binomial distribution:

$$N + m \sim Binomial\ (2m, \frac{1}{2}),$$

whose probability mass function (PMF) is

$$\Pr(N = k) = \binom{2m}{k+m}\frac{1}{2^{2m}},\ k \in \{-m, \cdots, m-1, m\},$$

with a variance $\frac{m}{2}$.

Lemma A.4. *(Binomial Mechanism) Consider a query $Q : \mathcal{D} \to \mathbb{Z}$ that takes data $d \in \mathcal{D}$ as input and outputs an integer. The Binomial mechanism $\mathcal{B}(d) = Q(d) + N$ adds binomial noise N with variance $\frac{m}{2}$ to the output. If $1 \le \Delta Q + \frac{2m+1}{\exp(\frac{\epsilon}{\Delta Q})+1} \le m+1$ for $\epsilon > 0$, then the following holds:*

(i) $\forall\ \epsilon > 0$, it satisfies (ϵ, δ)-differential privacy with:

$$\delta = \exp(-\frac{1}{m}(m - \Delta Q + 1 - \frac{2m+1}{\exp(\frac{\epsilon}{\Delta Q})+1})^2).$$

(ii) $\forall\ \delta \in (0,1)$, it satisfies (ϵ, δ)-differential privacy with:

$$\epsilon = \Delta Q \log(\frac{2m+1}{m - \Delta Q + 1 - \sqrt{m \log\frac{1}{\delta}}} - 1).$$

Note that this is a generalization (for arbitrary sensitivity ΔQ) to the version (for the case $\Delta Q = 1$) first proposed in [Dwork et al. 2006]. This is an approximation of the Gaussian mechanism; it has a much looser bound compared to the latter and more noise is needed to ensure a same level of privacy, which is consistent with the conclusion in [Dwork et al. 2006]. However, the Gaussian mechanism only works when $\epsilon < 1$, while our Binomial mechanism does not have this restriction and is more suitable for a discrete setting.

A.4 Privacy Analysis using Binomial Noise

Theorem A.5. *Let $Z_t = Q(D_t)$ and Δ be the sensitivity of Q $\forall t$, consider Algorithm 2 using Binomial noise with $Var(N_t) = \frac{m}{2}, \forall t$ that takes sequence $z_{1:T}$ as input and outputs $x_{1:T}$, $\forall\ \tilde{\delta} \in [0,1]$, if $1 \le w_t\Delta + \frac{2m+1}{\exp(\frac{\epsilon}{w_t\Delta})+1} \le m+1,\ \forall t$, then the algorithm is $(\tilde{\epsilon}_{\tilde{\delta}}, 1-(1-\tilde{\delta})\prod_{t=1}^{T}(1-\delta_t))$-differentially private for:*

$$\tilde{\epsilon}_{\tilde{\delta}} = \min\left\{\sum_{t=1}^{T}\epsilon_t, \sum_{t=1}^{T}\frac{(e^{\epsilon_t}-1)\epsilon_t}{e^{\epsilon_t}+1} + \sqrt{\sum_{t=1}^{T}2\epsilon_t^2\log(e + \frac{\sqrt{\sum_{t=1}^{T}\epsilon_t^2}}{\tilde{\delta}})}, \sum_{t=1}^{T}\frac{(e^{\epsilon_t}-1)\epsilon_t}{e^{\epsilon_t}+1} + \sqrt{\sum_{t=1}^{T}2\epsilon_t^2\log(\frac{1}{\tilde{\delta}})}\right\}$$

with any $\epsilon_t > 0$ *and corresponding*

$$\delta_t = \exp(-\frac{1}{m}(m - w_t\Delta + 1 - \frac{2m+1}{\exp(\frac{\epsilon_t}{w_t\Delta}) + 1})^2),$$

or with any $\delta_t \in (0,1)$ *and corresponding*

$$\epsilon_t = w_t\Delta \log(\frac{2m+1}{m - w_t\Delta + 1 - \sqrt{m \log \frac{1}{\delta_t}}} - 1).$$

A.5 Proofs for results of binomial AR(1) process

A.5.1 Proof of Proposition A.2. The MMSE estimate of Z_{t+1} given $Z_t = z_t$ is $\mathbb{E}(Z_{t+1}|Z_t = z_t)$. Since the thinning is performed independently, given $Z_t = z_t$, the probability generating function satisfies the following:

$$\begin{aligned}
G(s) &= \mathbb{E}_{Z_{t+1}|Z_t=z_t}(s^{Z_{t+1}}|Z_t = z_t) = \mathbb{E}(s^{\alpha\circ Z_t}|Z_t = z_t)\mathbb{E}(s^{\beta\circ(n-Z_t)}|Z_t = z_t) = (1-\beta+\beta s)^n(\frac{1-\alpha+\alpha s}{1-\beta+\beta s})^{z_t} \\
G'(s) &= n\beta(1-\beta+\beta s)^{n-1}(\frac{1-\alpha+\alpha s}{1-\beta+\beta s})^{z_t} + (1-\beta+\beta s)^n z_t(\frac{1-\alpha+\alpha s}{1-\beta+\beta s})^{z_t-1}\frac{\alpha}{\beta}\frac{\frac{1}{\beta}-\frac{1}{\alpha}}{(\frac{1}{\beta}-1+s)^2} \\
G''(s) &= n\beta^2(n-1)(1-\beta+\beta s)^{n-2}(\frac{1-\alpha+\alpha s}{1-\beta+\beta s})^{z_t} + 2n\beta(1-\beta+\beta s)^{n-1}z_t(\frac{1-\alpha+\alpha s}{1-\beta+\beta s})^{z_t-1}\frac{\alpha}{\beta}\frac{\frac{1}{\beta}-\frac{1}{\alpha}}{(\frac{1}{\beta}-1+s)^2} \\
&\quad +(1-\beta+\beta s)^n z_t((z_1-1)(\frac{1-\alpha+\alpha s}{1-\beta+\beta s})^{z_t-2}(\frac{\alpha}{\beta}\frac{\frac{1}{\beta}-\frac{1}{\alpha}}{(\frac{1}{\beta}-1+s)^2})^2 + (\frac{1-\alpha+\alpha s}{1-\beta+\beta s})^{z_t-1}\frac{\alpha}{\beta}2\frac{\frac{1}{\alpha}-\frac{1}{\beta}}{(\frac{1}{\beta}-1+s)^3})
\end{aligned}$$

Since $\beta = \pi(1-\rho)$, $\alpha = \beta + \rho$, $\mathbb{E}(Z_{t+1}|Z_t = z_t) = \lim_{s\to 1} G'(s)$ and $\mathrm{Var}(Z_{t+1}|Z_t = z_t) = \lim_{s\to 1} G''(s) + G'(s) - (G'(s))^2$ gives:

$$\begin{aligned}
\mathbb{E}(Z_{t+1}|Z_t = z_t) &= \rho z_t + n\pi(1-\rho)\ ; \\
\mathrm{Var}(Z_{t+1}|Z_t = z_t) &= \rho(1-\rho)(1-2\pi)z_t + n\beta(1-\beta).
\end{aligned}$$

The corresponding MSE is:

$$\mathbb{E}((Z_{t+1} - \mathbb{E}(Z_{t+1}|Z_t = z_t))^2|Z_t = z_t) = \mathrm{Var}(Z_{t+1}|Z_t = z_t)$$

A.5.2 Proof of Lemma A.4. Consider any $d, \hat{d} \in \mathcal{D}$, and with them the binomial mechanism outputs the same results b. Let $\bar{b} = b - Q(d)$

$$\frac{\Pr(b = Q(d) + noise)}{\Pr(b = Q(\hat{d}) + \hat{noise})} = \frac{\mathscr{F}_N(b - Q(d))}{\mathscr{F}_N(b - Q(\hat{d}))} = \frac{\binom{2m}{m+b-Q(d)}}{\binom{2m}{m+b-Q(\hat{d})}} = \frac{\binom{2m}{m+\bar{b}}}{\binom{2m}{m+\bar{b}+\Delta Q}} = \prod_{i=1}^{\Delta Q}\frac{m+\bar{b}+i}{m-\bar{b}+1-i}$$

A sufficient condition for (6) being bounded by $\exp(\epsilon)$ is:

$$\forall i \in \{1,2,\cdots,\Delta Q\},\ \frac{m+\bar{b}+i}{m-\bar{b}+1-i} \le \exp(\frac{\epsilon}{\Delta Q}) \overset{\tilde{b}=\bar{b}+i}{\Longleftrightarrow} \frac{m+\tilde{b}}{m-\tilde{b}+1} \le \exp(\frac{\epsilon}{\Delta Q}) \tag{6}$$

from (6), we have:

$$\bar{b} \le \min_{i\in[\Delta Q]} m + 1 - \frac{2m+1}{\exp(\frac{\epsilon}{\Delta Q})+1} - i = m + 1 - \frac{2m+1}{\exp(\frac{\epsilon}{\Delta Q})+1} - \Delta Q$$

Let $\bar{B}$ be the random variable of shifted $Binomial(2m, \frac{1}{2})$ with zero mean and realization $\bar{b}$. According to Chernoff bound, $\forall t \in [0, \sqrt{2m}]$, there is $\Pr(\bar{B} \geq t\frac{\sqrt{2m}}{2}) \leq e^{-t^2/2}$.

Then if $1 \leq \Delta Q + \frac{2m+1}{\exp(\frac{\epsilon}{\Delta Q})+1} \leq m+1$, there is:

$$\Pr(\bar{B} \geq m+1-\frac{2m+1}{\exp(\frac{\epsilon}{\Delta Q})+1} - \Delta Q) \leq \exp(-\frac{1}{m}(m-\Delta Q+1-\frac{2m+1}{\exp(\frac{\epsilon}{\Delta Q})+1})^2) = \delta$$

Similarly, given $\delta \in [0,1]$, the corresponding ϵ is:

$$\epsilon = \Delta Q \log\left(\frac{2m+1}{m-\Delta Q+1-\sqrt{m\log\frac{1}{\delta}}} - 1\right)$$

A.5.3 Proof of Theorem A.5. The data of each individual here spans over T time steps, the total privacy loss is the accumulation of privacy loss from T time steps:

$$\frac{\mathscr{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|z_{1:T})}{\mathscr{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|\hat{z}_{1:T})} = \frac{\mathscr{F}_{X_1|Z_1}(x_1|z_1)}{\mathscr{F}_{X_1|Z_1}(x_1|\hat{z}_1)} \cdot \prod_{t=2}^{T} \frac{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|z_t,x_{1:t-1})}{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|\hat{z}_t,x_{1:t-1})}$$

If x_t is released under (ϵ_t, δ_t)-differential privacy at time t, then the total privacy loss can be calculated using advanced composition theorem below:

Theorem A.6. *(Advanced composition theorem for differential privacy [Kairouz et al. 2017]) For any $\epsilon_k > 0$, $\delta_k \in [0,1]$ for $k \in \{1,2,\cdots,T\}$, and $\tilde{\delta} \in [0,1]$, the class of (ϵ_k, δ_k)-differentially private mechanisms satisfy $(\tilde{\epsilon}_{\tilde{\delta}}, 1-(1-\tilde{\delta})\prod_{k=1}^{T}(1-\delta_k))$-differential privacy under T-fold adaptive composition, for*

$$\tilde{\epsilon}_{\tilde{\delta}} = \min\left\{\sum_{k=1}^{T}\frac{(e^{\epsilon_k}-1)\epsilon_k}{e^{\epsilon_k}+1} + \sqrt{\sum_{k=1}^{T}2\epsilon_k^2\log(e+\frac{\sqrt{\sum_{k=1}^{T}\epsilon_k^2}}{\tilde{\delta}})}, \sum_{k=1}^{T}\epsilon_k, \sum_{k=1}^{T}\frac{(e^{\epsilon_k}-1)\epsilon_k}{e^{\epsilon_k}+1} + \sqrt{\sum_{k=1}^{T}2\epsilon_k^2\log(\frac{1}{\tilde{\delta}})}\right\}$$

First calculate the (ϵ_t, δ_t) at each stage by Lemma A.4. Since $N_t + m \sim Binomial(2m, \frac{1}{2})$, for $t \leq 2$, $X_t = Z_t + N_t$ so that the sensitivity $\Delta Q_t = \Delta$; for $t > 2$, $X_t = [(1-w_t)(\hat{\mu}_{t-1}(1-r_t) + r_t X_{t-1}) + w_t Z_t + N_t]$ and the sensitivity is $\Delta Q_t = w_t\Delta$. Let $w_t = 1$ for $t \leq 2$. Then $\forall \epsilon_t > 0$,

$$\delta_t = \exp(-\frac{1}{m}(m - w_t\Delta + 1 - \frac{2m+1}{\exp(\frac{\epsilon_t}{w_t\Delta})+1})^2)$$

or $\forall \delta_t \in (0,1)$,

$$\epsilon_t = w_t\Delta\log(\frac{2m+1}{m-w_t\Delta+1-\sqrt{m\log\frac{1}{\delta_t}}} - 1)$$

Apply Theorem A.6 directly, Theorem A.5 is proved.

B PROOFS

B.1 Proof of Proposition 2.4

Finding the MMSE estimate of Z_{t+1} given $Z_t = z_t$ is equivalent to finding the mapping

$$f^* = \underset{f}{\operatorname{argmin}}\ \mathbb{E}((Z_{t+1} - f(Z_t))^2 | Z_t = z_t) = \underset{f}{\operatorname{argmin}} \int_{-\infty}^{\infty} p(z_{t+1}|z_t)(z_{t+1} - f(z_t))^2 dz_{t+1}$$

Differentiating with respect to f and equating the result to zero gives:

$$\int_{-\infty}^{\infty} p(z_{t+1}|z_t) f^*(z_t) dz_{t+1} = f^*(z_t) = \int_{-\infty}^{\infty} p(z_{t+1}|z_t) z_{t+1} dz_{t+1} = \mathbb{E}(Z_{t+1}|Z_t = z_t)$$

Therefore, the MMSE estimate of Z_{t+1} given $Z_t = z_t$ is $\mathbb{E}(Z_{t+1}|Z_t = z_t)$.

Since (Z_t, Z_{t+1}) is jointly Gaussian:

$$\begin{pmatrix} Z_t \\ Z_{t+1} \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \sigma_z^2 & \rho\sigma_z^2 \\ \rho\sigma_z^2 & \sigma_z^2 \end{bmatrix} \right),$$

it implies that $Z_{t+1}|Z_t \sim \mathcal{N}(\mu(1-\rho) + \rho Z_t, \sigma_z^2(1-\rho^2))$, combining with the above result, the MMSE estimate of Z_{t+1} given $Z_t = z_t$ is $\mathbb{E}(Z_{t+1}|Z_t = z_t) = \mu(1-\rho) + \rho z_t$. The corresponding MSE is:

$$\mathbb{E}((Z_{t+1} - \mathbb{E}(Z_{t+1}|Z_t = z_t))^2 | Z_t = z_t) = \mathrm{Var}(Z_{t+1}|Z_t = z_t) = \sigma_z^2(1-\rho^2)$$

Since $Z_i \sim \mathcal{N}(\mu, \sigma_z^2)$, $N_i \sim \mathcal{N}(0, \sigma_n^2)$, there is $X_i = Z_i + N_i \sim \mathcal{N}(\mu, \sigma_z^2 + \sigma_n^2)$ and $\mathrm{Corr}(X_i Z_{t+1}) = \rho^{t+1-i} \frac{\sigma_z}{\sqrt{\sigma_z^2+\sigma_n^2}}$.

(X_i, Z_{t+1}) is jointly Gaussian:

$$\begin{pmatrix} X_i \\ Z_{t+1} \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu \end{bmatrix}, \begin{bmatrix} \sigma_z^2 + \sigma_n^2 & \rho^{t+1-i}\sigma_z^2 \\ \rho^{t+1-i}\sigma_z^2 & \sigma_z^2 \end{bmatrix} \right),$$

it implies the MMSE estimate of Z_{t+1} given $X_i = x_i$:

$$\mathbb{E}(Z_{t+1}|X_i = x_i) = \mu(1 - \rho^{t+1-i}\frac{\sigma_z^2}{\sigma_z^2 + \sigma_n^2}) + \rho^{t+1-i}\frac{\sigma_z^2}{\sigma_z^2 + \sigma_n^2} x_i$$

The corresponding MSE is:

$$\sigma_z^2(1 - (\rho^{t+1-i})^2 \frac{\sigma_z^2}{\sigma_z^2 + \sigma_n^2})$$

B.2 Proof of Theorem 5.1

According to [Abadi et al. 2016], for a mechanism $\mathscr{M}$ outputs o, with inputs d and $\hat{d}$, let a random variable $c(o; \mathscr{M}, d, \hat{d}) = \log \frac{\Pr(\mathscr{M}(d)=o)}{\Pr(\mathscr{M}(\hat{d})=o)}$ denote the privacy loss at o, and

$$\alpha_{\mathscr{M}}(\lambda) = \max_{d,\hat{d}} \log \mathbb{E}_{o\sim\mathscr{M}(d)}\{\exp(\lambda c(o; \mathscr{M}, d, \hat{d}))\}$$

There is:

$$\begin{aligned} c(x_{1:T}; \mathscr{M}, z_{1:T}, \hat{z}_{1:T}) &= \log \frac{\mathscr{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|z_{1:T})}{\mathscr{F}_{X_{1:T}|Z_{1:T}}(x_{1:T}|z_{1:T})} = \sum_{t=2}^{T} \log \frac{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|z_t, x_{1:t-1})}{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|\hat{z}_t, x_{1:t-1})} + \log \frac{\mathscr{F}_{X_1|Z_1}(x_1|z_1)}{\mathscr{F}_{X_1|Z_1}(x_1|\hat{z}_1)} \\ &= \sum_{t=1}^{T} c(x_t; \mathscr{M}_t, z_t, \hat{z}_t) \end{aligned}$$

and for any pair of sequences $z_{1:T}, \hat{z}_{1:T}$, the following holds

$$\begin{aligned} \log \mathbb{E}_{X_{1:T}\sim\mathscr{M}(Z_{1:T})}\{\exp(\lambda c(x_{1:T}; \mathscr{M}, z_{1:T}, \hat{z}_{1:T}))\} &= \log \mathbb{E}_{X_{1:T}\sim\mathscr{M}(Z_{1:T})}\{\exp(\lambda \sum_{t=1}^{T} c(x_t; \mathscr{M}_t, z_t, \hat{z}_t)\} \\ &\leq \sum_{t=1}^{T} \log \mathbb{E}_{X_t\sim\mathscr{M}(Z_t)}\{\exp(\lambda c(x_t; \mathscr{M}_t, z_t, \hat{z}_t)\} \end{aligned} \quad (7)$$

Therefore, $\alpha_{\mathscr{M}}(\lambda) \leq \sum_{t=1}^{T} \alpha_{\mathscr{M}_t}(\lambda)$ also holds.

Consider $\alpha_{\mathscr{M}_t}(\lambda)$ first.

For $t \leq 2 - T_0$, $X_t = Z_t + N_t$ with $N_t \sim \mathcal{N}(0, \sigma_n^2)$

$$c(x_t; \mathscr{M}_t, z_t, \hat{z}_t) = \log \frac{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|z_t, x_{1:t-1})}{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|\hat{z}_t, x_{1:t-1})} = \log \frac{\mathscr{F}_{N_t}(n_t)}{\mathscr{F}_{N_t}(\hat{n}_t)} \leq \frac{1}{2\sigma_n^2}\Delta(2n_t + \Delta) .$$

$$\begin{aligned}
\alpha_{\mathscr{M}_t}(\lambda) &= \log \mathbb{E}_{N_t \sim \mathcal{N}(0,\sigma_n^2)}\{\exp(\lambda \frac{1}{2\sigma_n^2}\Delta(2n_t + \Delta))\} \\
&= \log \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{1}{2\sigma_n^2}(n_t - \lambda\Delta)^2) \cdot \exp(\frac{1}{2\sigma_n^2}(\lambda^2 + \lambda)\Delta^2) dn_t \\
&= \frac{\lambda(\lambda+1)\Delta^2}{2\sigma_n^2} .
\end{aligned}$$

For $t > 2$,

$$X_t = (1 - w_t)(\hat{\mu}_{t-1}(1 - r_t) + r_t X_{t-1}) + w_t Z_t + N_t$$

with $N_t \sim \mathcal{N}(0, \sigma_n^2)$.

$$c(x_t; \mathscr{M}_t, z_t, \hat{z}_t) = \log \frac{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|z_t, x_{1:t-1})}{\mathscr{F}_{X_t|Z_t,X_{1:t-1}}(x_t|\hat{z}_t, x_{1:t-1})} = \log \frac{\mathscr{F}_{N_t}(n_t)}{\mathscr{F}_{N_t}(\hat{n}_t)} \leq \frac{1}{2\sigma_n^2} w_t \Delta(2n_t + w_t\Delta) .$$

$$\begin{aligned}
\alpha_{\mathscr{M}_t}(\lambda) &= \log \mathbb{E}\{\exp(\lambda \frac{1}{2\sigma_n^2} w_t\Delta(2n_t + w_t\Delta))\} \\
&= \log \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{1}{2\sigma_n^2}(n_t - \lambda w_t\Delta)^2) \cdot \exp(\frac{1}{2\sigma_n^2}(\lambda^2 + \lambda) w_t^2\Delta^2) dn_t \\
&= \frac{\lambda(\lambda+1) w_t^2\Delta^2}{2\sigma_n^2} .
\end{aligned}$$

If let $w_t = 1$ for $t \leq 2$, there is:

$$\alpha_{\mathscr{M}}(\lambda) \leq \lambda(\lambda+1)\frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^{T} w_t^2$$

Use the tail bound [Theorem 2, [Abadi et al. 2016]], for any $\epsilon_T \geq \frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^{T} w_t^2$, the algorithm is (ϵ_T, δ_T)-differentially private for

$$\delta_T = \min_{\lambda:\lambda\geq 0} h(\lambda) = \min_{\lambda:\lambda\geq 0} \exp(\lambda(\lambda+1)\frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^{T} w_t^2 - \lambda\epsilon_T)$$

To find $\lambda^* = \underset{\lambda:\lambda\geq 0}{\operatorname{argmin}}\, h(\lambda)$, take derivative of $h(\lambda)$ and assign 0 gives the solution $\bar{\lambda} = \frac{\epsilon_T}{\frac{\Delta^2}{\sigma_n^2}\sum_{t=1}^{T} w_t^2} - \frac{1}{2} \geq 0$, and $h''(\bar{\lambda}) > 0$, implies $\lambda^* = \bar{\lambda}$. Plug into (8) gives:

$$\delta_T = \exp\left((\frac{\frac{\Delta^2}{\sigma_n^2}\sum_{t=1}^{T} w_t^2}{4} - \frac{\epsilon_T}{2})(\frac{\epsilon_T}{\frac{\Delta^2}{\sigma_n^2}\sum_{t=1}^{T} w_t^2} - \frac{1}{2})\right) \tag{8}$$

Similarly, for any $\delta_T \in [0, 1]$, the algorithm is (ϵ_T, δ_T)-differentially private for

$$\epsilon_T = \min_{\lambda:\lambda\geq 0} h_1(\lambda) = \min_{\lambda:\lambda\geq 0} (\lambda+1)\frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^{T} w_t^2 + \frac{1}{\lambda}\log\left(\frac{1}{\delta_T}\right) \tag{9}$$

with $\lambda^* = \underset{\lambda:\lambda\geq 0}{\operatorname{argmin}}\, h_1(\lambda) = \sqrt{\frac{\log\frac{1}{\delta_T}}{\frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^{T} w_t^2}}$. Plug into (9) gives:

$$\epsilon_T = 2\sqrt{\frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^{T} w_t^2 \log(\frac{1}{\delta_T})} + \frac{\Delta^2}{2\sigma_n^2}\sum_{t=1}^{T} w_t^2 \tag{10}$$

B.3 Proof of Corollary 5.2

Let $\phi = \frac{\Delta^2\sum_{t=1}^{T} w_t^2}{\sigma_n^2}$, then according to Theorem 5.1,

$$\ln\delta_T = (\frac{\phi}{4} - \frac{\epsilon_T}{2})(\frac{\epsilon_T}{\phi} - \frac{1}{2})$$

reorganize gives:

$$\phi^2 + (8\ln\delta_T - 4\epsilon_T)\phi + 4\epsilon_T^2 = 0$$

$$\phi = 2\epsilon_T - 4\ln\delta_T \pm 4\sqrt{(\ln\delta_T)^2 - \epsilon_T\ln\delta_T}$$

Since $\epsilon_T \geq \frac{\phi}{2}$ must hold, only one case is possible.

$$\phi = \frac{\Delta^2\sum_{t=1}^{T} w_t^2}{\sigma_n^2} = 2\epsilon_T - 4\ln\delta_T - 4\sqrt{(\ln\delta_T)^2 - \epsilon_T\ln\delta_T}$$

Therefore,

$$\sigma_n^2 = \frac{\Delta^2\sum_{t=1}^{T} w_t^2}{2\epsilon_T + 4\ln\frac{1}{\delta_T} - 4\sqrt{(\ln\frac{1}{\delta_T})^2 + \epsilon_T\ln\frac{1}{\delta_T}}}$$

B.4 Proof of Theorem 6.1

$$\mathbb{E}_{X_{1:T}}(||x_{1:T} - z_{1:T}||^2) = \mathbb{E}_{X_{1:T}}(\sum_{t=1}^{T}(x_t - z_t)^2) = \mathbb{E}_{X_{1:T-1}}\{\sum_{t=1}^{T-1}(x_t - z_t)^2 + \underbrace{\mathbb{E}_{X_T|X_{1:T-1}}[(x_T - z_T)^2]}_{\textbf{term 1}}\} \tag{11}$$

Replacing $x_T = (1 - w_T)\hat{z}_T(x_{T-1}) + w_T z_T + n_T$ into **term 1** gives:

$$\textbf{term 1} \quad = \quad \mathbb{E}_{X_T|X_{1:T-1}}[((1 - w_t)(\hat{z}_T(x_{T-1}) - z_T) + n_T)^2] = (1 - w_T)^2(\hat{z}_T(x_{T-1}) - z_T)^2 + \sigma_n^2$$

Plug into Eqn. (11):

$$(11) \quad = \quad \mathbb{E}_{X_{1:T-1}}\{\sum_{t=1}^{T-1}(x_t - z_t)^2 + (1 - w_T)^2(\hat{z}_T(x_{T-1}) - z_T)^2 + \sigma_n^2\} = \mathbb{E}_{X_{1:T-2}}\{\sum_{t=1}^{T-2}(x_t - z_t)^2 + \sigma_n^2 + \textbf{term 2}\}$$

with

$$\begin{aligned}\textbf{term 2} \quad &= \quad \mathbb{E}_{X_{T-1}|X_{1:T-2}}\{(x_{T-1} - z_{T-1})^2 + (1 - w_T)^2(\hat{z}_T(x_{T-1}) - z_T)^2\} \\ &= \quad (1 - w_{T-1})^2(\hat{z}_{T-1}(x_{T-2}) - z_{T-1})^2 + \sigma_n^2 + (1 - w_T)^2\underbrace{\mathbb{E}_{X_{T-1}}\{(\hat{z}_T(x_{T-1}) - z_T)^2\}}_{\textbf{term 3}}\end{aligned}$$

Since $\hat{z}_T(x_{T-1})$ is the LMMSE estimator of Z_T given X_{T-1}, **term 3** is just the corresponding MSE. For a Gaussian AR(1) process $Z_{1:T}$ with $Z_t \sim \mathcal{N}(\mu, \sigma_z^2)$ and $\text{Corr}(Z_t Z_{t-1}) = \rho$. There is:

$$\textbf{term 3} = \sigma_z^2(1-\rho^2\frac{\sigma_z^2}{\sigma_z^2+\sigma_n^2})$$

Therefore,

$$\begin{aligned}
(11) &= \mathbb{E}_{X_{1:T-2}}\{\sum_{t=1}^{T-2}(x_t-z_t)^2 + (1-w_{T-1})^2(\hat{z}_{T-1}(x_{T-2})-z_{T-1})^2\} + (1-w_T)^2\sigma_z^2(1-\rho^2\frac{\sigma_z^2}{\sigma_z^2+\sigma_n^2}) + 2\sigma_n^2 \\
&= \cdots \\
&= \mathbb{E}_{X_1}\{(x_1-z_1)^2 + (1-w_2)^2(\hat{z}_2(x_1)-z_2)^2\} + \sigma_z^2(1-\rho^2\frac{\sigma_z^2}{\sigma_z^2+\sigma_n^2})\sum_{t=3}^{T}(1-w_t)^2 + (T-1)\sigma_n^2 \\
&= \sigma_z^2(1-\rho^2\frac{\sigma_z^2}{\sigma_z^2+\sigma_n^2})\sum_{t=2}^{T}(1-w_t)^2 + T\sigma_n^2
\end{aligned}$$

Since $w_1 = 1$,

$$\mathbb{E}_{X_{1:T}}(||x_{1:T}-z_{1:T}||^2) = \sigma_z^2(1-\rho^2\frac{\sigma_z^2}{\sigma_z^2+\sigma_n^2})\sum_{t=1}^{T}(1-w_t)^2 + T\sigma_n^2$$

B.5 Proof of Theorem 6.2

Since both satisfy (ϵ_T, δ_T)-differential privacy, according to Corollary 5.2, $(\sigma_n^2)^A$, $(\sigma_n^2)^B$ should satisfy:

$$\frac{T}{(\sigma_n^2)^B} = \frac{\sum_{t=1}^{T} w_t^2}{(\sigma_n^2)^A} = \frac{2\epsilon_T + 4\ln\frac{1}{\delta_T} - 4\sqrt{(\ln\frac{1}{\delta_T})^2 + \epsilon_T \ln\frac{1}{\delta_T}}}{\Delta^2}$$

By Theorem 6.1,

$$\mathbb{E}_{X_{1:T}^A}(||x_{1:T}-z_{1:T}||^2) = \sigma_z^2(1-\rho^2\frac{\sigma_z^2}{\sigma_z^2+(\sigma_n^2)^A})\sum_{t=1}^{T}(1-w_t)^2 + T(\sigma_n^2)^A$$

$$\mathbb{E}_{X_{1:T}^B}(||x_{1:T}-z_{1:T}||^2) = T(\sigma_n^2)^B = T(\sigma_n^2)^A\frac{T}{\sum_{t=1}^{T} w_t^2}$$

If $\mathbb{E}_{X_{1:T}^A}(||x_{1:T}-z_{1:T}||^2) < \mathbb{E}_{X_{1:T}^B}(||x_{1:T}-z_{1:T}||^2)$, then $\exists t$ s.t. $w_t \neq 1$ and

$$\sigma_z^2(1-\rho^2\frac{\sigma_z^2}{\sigma_z^2+(\sigma_n^2)^A})\sum_{t=1}^{T}(1-w_t)^2 < T(\sigma_n^2)^A(\frac{T}{\sum_{t=1}^{T} w_t^2} - 1)$$

Reorganize it implies:

$$\frac{(\sigma_n^2)^A/\sigma_z^2}{1-\frac{\rho^2}{1+(\sigma_n^2)^A/\sigma_z^2}} > \frac{\sum_{t=1}^{T}(1-w_t)^2}{T(\frac{T}{\sum_{t=1}^{T} w_t^2} - 1)} \tag{12}$$

Therefore, if $\exists\ \{w_t\}_{t=1}^T, w_t \in (0,1)$ and $(\sigma_n^2)^A$ satisfy both (12) and (12), then $x_{1:T}^A$ will be more accurate than $x_{1:T}^B$. Consider the case when $w_t = w \in (0,1), \forall t$.

Then the right hand side of (12) is reduced to $h_1(w) = \frac{(1-w)^2}{\frac{1}{w^2}-1}$, since

$$\begin{aligned} \lim_{w\to 1} h_1(w) &= 0; \ \lim_{w\to 0} h_1(w) = 0 \\ h_1'(w) &= \frac{-2w(w^2+w-1)}{(1+w)^2} \end{aligned}$$

$\exists$ only one $\bar{w}$ over $(0,1)$ such that $\bar{w}^2 + \bar{w} - 1 = 0$. Therefore, $h_1(w)$ is strictly increasing from 0 to $h_1(\bar{w}) > 0$ over $(0, \bar{w})$ and strictly decreasing over from $h_1(\bar{w}) > 0$ to 0 over $(\bar{w}, 1)$.

Let $\xi = (\sigma_n^2)^A/\sigma_z^2 \geq 0$, then the left hand side of (12) can be re-written as $h_2(\xi) = \frac{\xi}{1-\frac{\rho^2}{1+\xi}}$, we have:

$$h_2'(\xi) = \frac{\xi^2 + 2\xi(1-\rho^2) + (1-\rho^2)}{(1+\xi-\rho^2)^2}$$

Since $h_2(0) = 0$ and $h_2'(\xi) > 0$ over $\xi \in [0, \infty)$, $h_2(\xi)$ is strictly increasing from 0 to $+\infty$ over $\xi \in [0, \infty)$. For all pairs of $(w, (\sigma_n^2)^A)$ satisfying (12), w and $(\sigma_n^2)^A$ is bijective and we can write $\xi = h_3(w)$ for some strictly increasing function h_3.

Since both h_2, h_3 are strictly increasing functions, $h_2(h_3(w))$ is strictly increasing from 0 over $w \in (0,1)$. Therefore, $\exists w \in (0,1)$, such that $h_2(h_3(w)) > h_1(w)$ and $x_{1:T}^A$ released by our method is more accurate than $x_{1:T}^B$.

Moreover, if $w > \frac{1-(\sigma_n^2)^B/\sigma_z^2}{1+(\sigma_n^2)^B/\sigma_z^2}$, then re-organize it implies

$$w^2 \frac{(\sigma_n^2)^B}{\sigma_z^2} > h_1(w).$$

Since $h_2(h_3(w)) = \frac{w^2\frac{(\sigma_n^2)^B}{\sigma_z^2}}{1-\frac{\rho^2}{1+w^2\frac{(\sigma_n^2)^B}{\sigma_z^2}}} > w^2\frac{(\sigma_n^2)^B}{\sigma_z^2}$, it further implies $h_2(h_3(w)) > h_1(w)$.

Therefore, if

$$w > \frac{1-(\sigma_n^2)^B/\sigma_z^2}{1+(\sigma_n^2)^B/\sigma_z^2},$$

then $x_{1:T}^A$ will be more accurate than $x_{1:T}^B$.

C ADDITIONAL EXPERIMENTS

Results on ride-sharing counts datasets are shown in Figures 15-19. Specifically, Figures 16-19 show the private sequences aggregated from 10 runs of experiments using different methods, Figure 15 shows the comparison of various methods under different privacy guarantee.

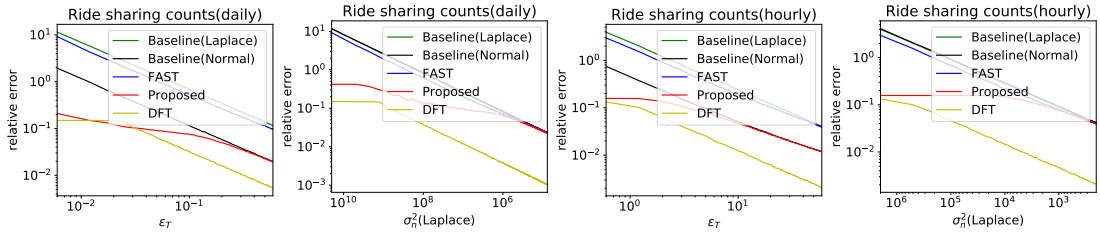


Fig. 15. Comparison of different methods

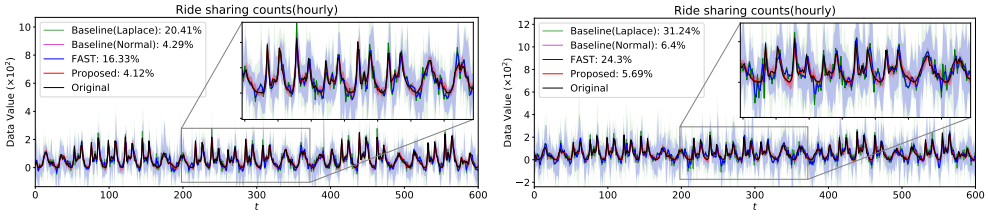


Fig. 16. Private sequences using different methods under the same $\epsilon_T = 2 * 10^{-2}T$(left) and $\epsilon_T = 1.3 * 10^{-2}T$(right)

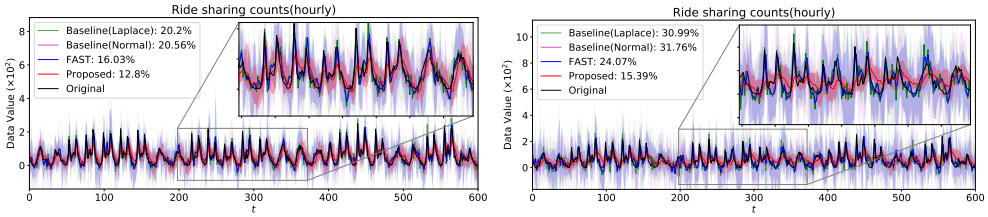


Fig. 17. Private sequences using different methods where parameters are selected such that error achieved by baseline-Normal is slightly larger than baseline-Laplace: σ_n^2(Laplace) $= 5 * 10^3$(left) and σ_n^2(Laplace) $= 1.18 * 10^4$(right)

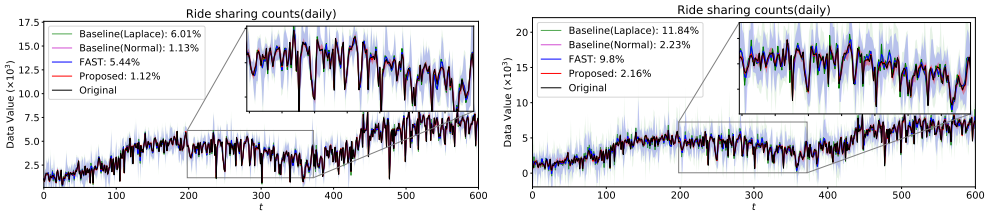


Fig. 18. Private sequences using different methods under the same $\epsilon_T = 2 * 10^{-3}T$(left) and $\epsilon_T = 10^{-4}T$(right)

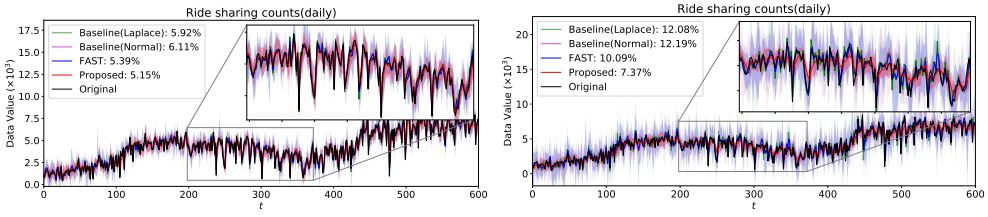


Fig. 19. Private sequences using different methods where parameters are selected such that error achieved by baseline-Normal is slightly larger than baseline-Laplace: σ_n^2(Laplace) $= 5 * 10^5$(left) and σ_n^2(Laplace) $= 2 * 10^8$(right)