Constant Matters: Fine-grained Error Bound on Differentially Private Continual Observation

Hendrik Fichtenberger [1] **Monika Henzinger** [2] **Jalaj Upadhyay** [3]

Abstract

We study fine-grained error bounds for differentially private algorithms for counting under continual observation. Our main insight is that the matrix mechanism when using lower-triangular matrices, can be used in the continual observation model. More specifically, we give an explicit factorization for the counting matrix M_{count} and upper bound the error explicitly. We also give a fine-grained analysis, specifying the exact constant in the upper bound. Our analysis is based on upper and lower bounds of the *completely bounded norm* (cb-norm) of M_{count}. Along the way, we improve the best-known bound of 28 years by Mathias (SIAM Journal on Matrix Analysis and Applications, 1993) on the cb-norm of M_{count} for a large range of the dimension of M_{count}. Furthermore, we are the first to give concrete error bounds for various problems under continual observation such as binary counting, maintaining a histogram, releasing an approximately cut-preserving synthetic graph, many graph-based statistics, and substring and episode counting. Finally, we note that our result can be used to get a fine-grained error bound for non-interactive local learning and the first lower bounds on the additive error for (ϵ, δ)-differentially-private counting under continual observation.

1. Introduction

In recent times, many large-scale applications of data analysis involved repeated computations because of the incidence of infectious diseases (Apple, 2021; CDC, 2020), typically with the goal of preparing some appropriate response. However, the privacy of the user data (such as a positive or negative test result) is equally important. In such an application, the system is required to continually produce outputs while preserving a robust privacy guarantee such as *differential privacy*. This setting was already used as a motivating example by Dwork et al. (2010) in the first work on differential privacy under continual release, where they write:

> "Consider a website for H1N1 self-assessment. Individuals can interact with the site to learn whether symptoms they are experiencing may be indicative of the H1N1 flu. The user fills in some demographic data (age, zipcode, sex), and responds to queries about his symptoms (fever over $100.4°$ F?, sore throat?, duration of symptoms?). We would like to continually analyze aggregate information of consenting users in order to monitor regional health conditions, with the goal, for example, of organizing improved flu response. Can we do this in a differentially private fashion with reasonable accuracy (despite the fact that the system is continually producing outputs)?"

In the *continual release* (or *observation*) *model* the input data arrives as a stream of items $x_1, x_2, \ldots, x_T$, with data x_i arriving in *round* i and the mechanism has to be able to output an answer after the arrival of each item. The study of the continual release model was initiated concurrently by Dwork et al. (2010) and Chan et al. (2011) through the study of the *(continual) binary counting* problem: Given a stream of bits, i.e., zeros and ones, after each bit, output the sum of the bits so far, i.e., the number of ones in the input so far. Dwork et al. (2010) showed that there exists a differentially private mechanism, called the *binary (tree) mechanism*, for this problem with an additive error of $O(\log^{5/2} T)$, where the additive error is the maximum additive error over all rounds i. This was further improved to $O(\log^{5/2}(t))$ at time $t \leqslant T$ by Chan et al. (2011) (see their Corollary 5.3). These algorithms use Laplace noise for differential privacy. Jain et al. (2021) showed that using Gaussian noise instead an additive error of $O(\log(t)\sqrt{\log(T)})$ can be achieved. However, the constant has never been

[*]Equal contribution [1]Google Zurich, Switzerland [2]Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria [3]Rutgers University, USA. Correspondence to: Hendrik Fichtenberger <fichtenberger@google.com>, Monika Henzinger <monika.henzinger@ist.ac.at>, Jalaj Upadhyay <jalaj.upadhyay@rutgers.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

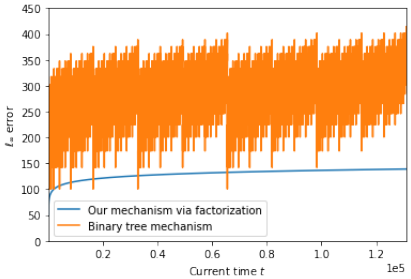
explicitly stated. Given the wide applications of binary counting in many downstream tasks, such as counting in the sliding window model (Bolot et al., 2013), frequency estimation (Cardoso & Rogers, 2021), graph problems (Fichtenberger et al., 2021), frequency estimation in the sliding window model (Chan et al., 2012; Epasto et al., 2023; Huang et al., 2021; Upadhyay, 2019), counting under adaptive adversary (Jain et al., 2021), median estimation (Alabi et al., 2022), optimization (Choquette-Choo et al., 2022; Denisov et al., 2022; Henzinger et al., 2023; Kairouz et al., 2021; Smith et al., 2017), graph spectrum (Upadhyay et al., 2021), and matrix analysis (Upadhyay & Upadhyay, 2021), constants can define whether the output is useful or not in practice. In fact, from the practitioner's point of view, it is important to know the constant hidden in the $O(\cdot)$ notation. With this in mind, we ask the following central question:

> *Can we get fine-grained bounds on the constants in the additive error of differentially private algorithms for binary counting under a continual release?*

The problem of reducing the additive error for binary counting under continual release has been pursued before (see Wang et al. (2021) and references therein). Most of them use some "smoothening" technique (Wang et al., 2021), assume some structure in the data (Rastogi & Nath, 2010), or measure error in mean squared loss (Wang et al., 2021)[1]. There is a practical reason to smoothen the output of the binary mechanism as its additive error is highly non-smooth (see Figure 1) due to the way the binary mechanism works: its additive error at any time t depends on how many dyadic intervals are summed up in the output for t. This forces the error to have non-uniformity of order $\log_2(T)$, which makes it hard to *interpret*[2]. For example in *exposure-notification systems* that have to operate when the stream length is in order of 10^8, it is desirable that the algorithm is scalable and its output fulfills properties such as monotonicity and smoothness to make the output interpretable. Thus, *the focus of this paper is to design a scalable mechanism for binary counting in the continual release model with a smooth additive error and to show a (small) fine-grained error bound.*

Our contributions. We prove concrete bounds on the additive error for counting under continual release that is tight up to a small additive gap. Since our bounds are closed-form expressions, it is straightforward to evaluate them for any data analysis task. Furthermore, our algorithms only perform few matrix-vector multiplications and additions, which makes them easy to implement and tailor to operations natively implemented in modern hardware. Finally, our algorithms are also efficient and the additive error of the output is smooth. As counting is a versatile building block, we get concrete bounds on a wide class of problems under continual release such as maintaining histograms, generating synthetic graphs that preserve cut sizes, computing various graph functions, and substring and counting episode in string sequences (see Section 1.2 for more details). Furthermore, this also leads to an improvement in the additive error for non-interactive local learning (Smith et al., 2017), private learning on non-Euclidean geometry (Asi et al., 2021), and private online prediction (Asi et al., 2022a): These algorithms use the binary mechanism as a subroutine and using our mechanism instead would give constant-factor improvements in the additive error. This, in turn, allows to *decrease* the privacy parameter in private learning, which is highly desirable and listed as motivation in several recent works (Asi et al., 2022b; Denisov et al., 2022; Henzinger et al., 2023)[3].

Figure 1: Additive ℓ_∞ error with $T = 2^{16}, \epsilon = 0.8, \delta = 10^{-10}$.



Our bounds bridge the gap between theoretical work on differential privacy, which mostly concentrates on asymptotic analysis to reveal the *capabilities of* differential private algorithms, and practical applications, which need to obtain useful *information from* differential private algorithms for their specific use cases.

1.1. The Formal Problem

We study the problem of continual release of an input vector x over T time epochs, i.e., the vector x is given incrementally to the mechanism in *rounds* or *time steps*. In time step t, $x[t]$ is revealed to the mechanism and it has to output $\sum_{i=1}^{t} x[i]$ under differential privacy. Equivalently, let x_t be a t-dimensional vector formed by the first t coordinates of x and M_{count} be a lower diagonal binary matrix defined below

$$M_{\mathsf{count}}[i,j] = \begin{cases} 1 & i \geqslant j \\ 0 & i < j \end{cases} \tag{1}$$

[1]While mean squared error is useful in some applications like learning (Denisov et al., 2022; Henzinger et al., 2023; Kairouz et al., 2021), in many applications we prefer a worst-case additive error, the metric of choice in this paper.

[2]For example, consider a use case of ECG monitoring on a smartwatch of a heart patient. Depending on whether $t = 2^i - 1$ and $t = 2^i$ for some $i \in \mathbb{N}$, the error of the output of binary mechanism might send an SOS signal or not.

[3]Note that Asi et al. (2022b) and Henzinger et al. (2023) appeared subsequent to this work on arXiv.

where for any matrix A, $A[i,j]$ denotes its $(i,j)^{\text{th}}$ coordinate. Then continual observation model requires us to output $M_{\mathsf{count}}(t)x_t$, where $M_{\mathsf{count}}(t)$ is the $t \times t$ submatrix of M_{count}. Similar to previous work (Chan et al., 2011; Dwork et al., 2010), we consider *event-level privacy*, that is, two streams are neighboring if they differ in exactly one time epoch by at most 1.

There has been a large body of work on designing differentially private algorithms for general linear query matrices in the static setting, i.e., not under continual release. One of the scalable techniques that provably reduce the error on linear queries is a query matrix optimization technique known as *workload optimizer* (see McKenna et al. (2021) and references therein). There have been various algorithms developed for this, one of them being the *matrix mechanism* (Li et al., 2015), which first determines two matrices R and L such that $M = LR$ and then outputs $L(Rx + z)$, where $z \sim N(0, \sigma^2\mathbb{I})$ is a vector of Gaussian values for a suitable choice of σ^2 and $\mathbb{I}$ is the identity matrix. For a privacy budget (ϵ, δ), it can be shown that the additive error, denoted as ℓ_∞ error of the answer vector (see Theorem 2.3), of the matrix mechanism for $|\mathcal{Q}|$ linear queries with ℓ_2-sensitivity $\Delta_{\mathcal{Q}}$ (eq. (7)) represented by a workload matrix M using the Gaussian mechanism is as follows: with probability $2/3$ over the random coins of the algorithm, the additive error is at most

$$\underbrace{\frac{2}{\epsilon}\sqrt{\frac{4}{9} + \ln\left(\frac{1}{\delta}\sqrt{\frac{2}{\pi}}\right)}}_{C_{\epsilon,\delta}} \Delta_{\mathcal{Q}} \left\|L\right\|_{2\to\infty} \left\|R\right\|_{1\to 2} \sqrt{\ln(6|\mathcal{Q}|)}, \tag{2}$$

where $\|A\|_{2\to\infty}$ (resp., $\|A\|_{1\to 2}$) is the maximum ℓ_2 norm of rows (resp. columns) of A. The function $C_{\epsilon,\delta}$ is a function arising as in the proof of the privacy guarantee of the Gaussian mechanism when $\epsilon < 1$ (see Theorem A.1 in Dwork & Roth (2014). If $\epsilon \geqslant 1$, one can analytically compute $C_{\epsilon,\delta}$ using Algorithm 1 in Balle & Wang (2018). For the rest of this paper, we use $C_{\epsilon,\delta}$ to denote this function. Therefore, we need to find a factorization $M = LR$ that minimizes $\|L\|_{2\to\infty}\|R\|_{1\to 2}$. Note that the quantity

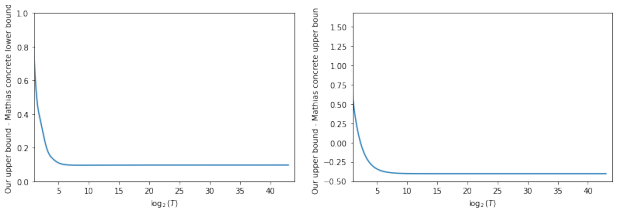
$$\|M\|_{\mathsf{cb}} = \min_{M=LR}\{\|L\|_{2\to\infty}\|R\|_{1\to 2}\} = \max_{W}\frac{\|W \bullet M\|}{\|W\|}.$$

is the *completely bounded norm*[4] (abbreviated as cb-norm). Here $W \bullet M$ denotes the Schur product (Schur, 1911).

The factor $C_{\epsilon,\delta}\Delta_{\mathcal{Q}}\sqrt{\ln(6|\mathcal{Q}|}$ in equation (2) is due to the error bound of the Gaussian mechanism followed by the union bound and is the same for all factorizations. Thus to get a concrete additive error, we need to find a factorization $M = LR$ such that the quantity $\|M\|_{\mathsf{cb}}$ is not just small, but also can be computed concretely. *Furthermore, we observe that if both L and R are lower-triangular matrices, then the resulting mechanism works not only in the static setting but also in the continual release model*. Therefore, for the rest of the paper, we only focus on finding such a factorization of the workload matrix M_{count}, which is a fundamental query in the continual release model.

[4] Paulsen (1986) in Section 7.7 attributes the second equality to Haagerup (1980).

Figure 2: (left) Gap between our upper bound and explicitly computed Mathias lower bound. (right) The gap between our upper bound and explicitly computed Mathias upper bound



1.2. Our Results

1. Bounding $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$. The question of finding the optimal value of $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$ was also raised in the conference version of Matoušek et al. (2020), who showed asymptotically tight bound. In the past, there has been considerable effort to get a tight bound on $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$ (Davidson, 1984; Mathias, 1993) with the best-known result prior to this work is the following due to Mathias (1993) (Corollary 3.5):

$$\left(\frac{1}{2} + \frac{1}{2T}\right)\widehat{\gamma}(T) \leqslant \|M_{\mathsf{count}}\|_{\mathsf{cb}} \leqslant \frac{\widehat{\gamma}(T)}{2} + \frac{1}{2}, \tag{3}$$

where $\widehat{\gamma}(T) = \frac{1}{T}\sum_{j=1}^{T}\left|\csc\left(\frac{(2j-1)\pi}{2T}\right)\right|$.

The key point to note is that the proof of Mathias (1993) relies on the dual characterization of cb-norm, and, thus, does not give an explicit factorization. In contrast, we give an explicit factorization into lower triangular matrices that achieve a bound in terms of the function, $\Psi : \mathbb{N} \to \mathbb{R}$ defined over natural numbers as follows:

$$\Psi(T) := 1 - \frac{1-\gamma}{\pi} + \frac{\ln(T)}{\pi} + \frac{2}{T} \tag{4}$$

where γ is the Euler-Mascheroni constant ($\gamma \leqslant 0.578$).

Theorem 1.1 (Upper bound on $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$)**.** *Let $M_{\mathsf{count}} \in \{0,1\}^{T\times T}$ be the matrix defined in eq. (1). Then, there is an explicit factorization $M_{\mathsf{count}} = LR$ into lower triangular matrices such that*

$$\|L\|_{2\to\infty}\|R\|_{1\to 2} \leqslant \Psi(T). \tag{5}$$

The bounds in eq. (3) do not have a closed form formula; however, we can show that it converges in the limit as $T \to$

∞ (Theorem C.4). However, our focus in this paper is on concrete bounds. Therefore, we compare our bound to the non-constructive bounds in eq. (3) by computing concrete values of $\widehat{\gamma}(T)$ for $1 \leqslant T \leqslant 2^{44}$. We show that for $8 \leqslant T \leqslant 2^{44}$, we improve on the upper bounds given by Mathias (1993), with an improvement of ≈ 0.35 for $50 \leqslant T \leqslant 2^{44}$. This is the first improvement on the cb-norm of M_{count} in about three decades. Furthermore our (theoretical) upper bound and the (analytically computed) *lower* bound of Mathias (1993) are less than 0.1 apart for all $2^5 \leqslant T \leqslant 2^{44}$ (Figure 2).

Additionally, our result has the advantage that we achieve the bound with an explicit factorization of $M_{\mathsf{count}} = LR$ such that both L and R are *lower-triangular* matrices. As discussed above, this allows us to use it for various applications. Using this fact and carefully choosing the "noise vector" for every time epoch, the following result is a consequence of Theorem 1.1 and eq. (2):

Theorem 1.2 (Upper bound on differentially private continual counting)**.** *Let $\epsilon, \delta \in (0,1)$ be the privacy parameters. There is an (ϵ, δ)-differentially private algorithm for binary counting in the continual release model with output a_t in round t such that in every execution, with probability at least $1-\beta$ over the coin tosses of the algorithm, simultaneously, for all rounds t with $1 \leq t \leq T$, it holds that*

$$\left| a_t - \sum_{i=1}^{t} x_i \right| \leqslant C_{\epsilon,\delta} \Psi(t) \sqrt{\ln(6T/\beta)}, \tag{6}$$

where $\Psi(t)$ is as in eq. (4). The time to output a_t in round t is $O(t)$.

As mentioned above, the binary mechanism of Chan et al. (2011) and Dwork et al. (2010) and its improvement (when the error metric is expected mean squared) by Honaker (2015) can be seen as factorization mechanism. This has been independently noticed by Denisov et al. (2022). While Chan et al. (2011) and Dwork et al. (2010) do work in the continual release model, Honaker (2015)'s optimization does not because for a partial sum, $\sum_{i \leqslant t} x_i$, it also uses the information stored at the nodes formed after time t. Therefore, for the comparison with related work, we do not include Honaker's optimization (Honaker, 2015). Moreover, Honaker's optimization is for minimizing the expected mean squared error (i.e., in ℓ_2^2 norm), and *not* the ℓ_∞ error, which is the focus of our work. To the best of our knowledge, only Chan et al. (2011) and Dwork et al. (2010) consider additive error in terms of ℓ_∞ norm. All other approaches used for binary counting under continual observation (see (Wang et al., 2021) and references therein) use some form of smoothening of the output and consider the *expected mean squared error*. While useful in some applications, many applications require to bound the additive ℓ_∞ error.

The most relevant work to ours is the subsequent work by Henzinger et al. (2023) that show that our factorization gives almost tight concrete bounds on performing counting under continual observation with respect to the *expected mean squared error*. On the other hand, we bound the *maximum absolute additive error*, i.e., in ℓ_∞ norm of the error. Note that our explicit factorization for M_{count} has the nice property that there are exactly T distinct entries (instead of possibly T^2 entries in Denisov et al. (2022))'s factorization. This has a large impact on computation time.

Remark 1.3 (Suboptimality of the binary mechanism with respect to the constant). The binary mechanism computes a linear combination of the entries in the streamed vector x as all of the internal nodes of the binary tree can be seen as a linear combination of the entries of x. Now we can consider the binary mechanism as a matrix mechanism. The right factor R_{binary} is constructed as follows: $R_{\mathsf{binary}} = W_m$ where $W_1, \cdots, W_m$ are defined recursively as follows:

$$W_1 = \begin{pmatrix} 1 \end{pmatrix}, \quad W_k = \begin{pmatrix} W_{k-1} & 0 \\ 0 & W_{k-1} \\ 1_{2^{k-2}} & 1_{2^{k-2}} \end{pmatrix}, \quad k \leqslant m.$$

That is, $R_{\mathsf{binary}} = W_m \in \{0,1\}^{(2T-1)\times T}$, with each row corresponding to the partial sum computed for the corresponding node in the binary tree from leaf i to the root. The corresponding matrix L_{binary} is a matrix in $\{0,1\}^{T\times(2T-1)}$, where row i has a one in at most $\log_2(i)$ entries, corresponding exactly to the binary representation of i.

In particular, this factorization view tells us that $\|L\|_{2\to\infty} \|R\|_{1\to 2} = \log_2(T)(\log_2(T)+1)$, which implies that the ℓ_∞-error is suboptimal by a factor of $\pi \log_2(e)$. Our experiments (Figure 3) confirm this behavior experimentally. This was observed for the *mean-squared error* by several works (Denisov et al., 2022; Honaker, 2015) culminating in the work of Henzinger et al. (2023), who gave a theoretical proof for the suboptimality of binary mechanism for mean-squared error.

Applications. Our result for continual binary counting can be extended in various directions. We show how to use it to quantify the additive error for (1) outputting a synthetic graph on the same vertex set which approximately preserves the values of all (S,P)-cuts with S and P being disjoint vertex sets of the graph, (2) frequency estimation (aka histogram estimation), (3) various graph functions, (4) substring counting, and (5) episode counting. Our mechanism can also be adapted for the locally private non-interactive learners of Smith et al. (2017) by replacing the binary mechanism with our matrix mechanism, which requires major changes in the analysis. In Section 1.2, we tabulate these applications. Based on a lower bound construction of Jain et al. (2021), we show in Section C that for large enough T and

constant $|S|$ the additive error in (1) is tight up to polylogarithmic factors and the additive error in (4) is tight for large enough T up to a factor that is linear in $\log\log|\mathcal{U}|\ln T$, where $\mathcal{U}$ is the universe of letters (see Theorem B.4 and Section C for details). Finally, we can use our mechanism to estimate the running average of a sequence of T bits with absolute error $\Psi(t)C_{\epsilon,\delta}\sqrt{\ln(6T)}/t$. All the applications are presented in detail in Appendix B. We note that all our algorithms are differentially private in the setting considered in Dwork et al. (2010).

2. Lower Bounds. We now turn our attention to new lower bounds on continual counting under approximate differential privacy. Prior to this paper, the only known lower bound for differentially private continual observation was for counting under pure-differential privacy. There are a few other methods to prove lower bounds. For example, we can use the relation between hereditary discrepancy and private linear queries (Muthukrishnan & Nikolov, 2012) along with the generic lower bound on hereditary discrepancy (Matoušek et al., 2020) to get an $\Omega(1)$ lower bound. This can be improved by using the reduction of continual counting to threshold and get an $\Omega(\log^*(T))$ (Bun et al., 2015).

Our lower bound is for a class of mechanisms, called *data-independent mechanisms*, that add a random variable sampled from the distribution which is independent of the input. Most commonly used differentially private mechanisms, such as the Laplace mechanism and the Gaussian mechanism fall under this class.

For binary counting, recall from eq. (3) that there exists a lower bound on $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$ in terms of $\widehat{\gamma}(T)$. In Theorem C.4, we show that

$$\lim_{T\to\infty}\widehat{\gamma}(T)\to\frac{2\ln(T)}{\pi}$$

from above. Combined with the proof idea of Edmonds et al. (2020), this gives the following bound for non-adaptive input streams, i.e., where the input stream is generated before seeing any output of the mechanism.

Theorem 1.4 (Lower bound)**.** *Let ϵ,δ be a sufficiently small constant and $\mathfrak{M}$ be the set of data-independent (ϵ,δ)-differentially private algorithms for binary counting under non-adaptive continual observation. Then*

$$\max_{x\in\{0,1\}^T}\mathbb{E}_{\mathcal{M}\in\mathfrak{M}}\left[\|\mathcal{M}(x)-M_{\mathsf{count}}x\|_\infty^2\right]\in\Omega\left(\frac{\ln^2(T)}{\epsilon^2}\right).$$

That is, the variance of the ℓ_∞-error is $\Omega(\ln^2(T)/\epsilon^2)$. A formal proof of Theorem 1.4 is presented in Section C.

1.3. Our Technical Contribution

1. Using the matrix mechanism in the continual release model. Our idea to use the matrix mechanism $\mathcal{F}$ in the continual release model is as follows: Assume M is known to $\mathcal{F}$ before any items of the stream x arrive and there exists an explicit factorization of $M=LR$ into lower triangular matrices L and R that can be computed efficiently by $\mathcal{F}$ during preprocessing. As we show, this is the case for the matrix M_{count}. This property leads to the following mechanism: At the time t, the mechanism $\mathcal{F}$ creates x' with consists of the current x-vector with $T-t$ zeros appended, and then returns the tth entry of $L(Rx'+z)$, where z is a suitable "noise vector". As L and R are lower-triangular, the tth entry of $L(Rx'+z)$ is identical to the tth entry in $L(Rx^f+z)$, where x^f is the final input vector x, and, thus, it suffices to analyze the error of the static matrix mechanism. Note that our algorithm can be implemented so that it requires time $O(t)$ at time t. The advantage of this approach is that it allows us to use the analysis of the static matrix mechanism while getting an *explicit bound on the additive error* of the mechanism in the continual release model.

Factorization in terms of lower triangular matrices is not known to be necessary for the continual release model; however, Denisov et al. (2022) pointed out that an arbitrary factorization will not work. For example, they discuss that Honaker (2015)'s optimization of the binary mechanism can be seen as a factorization but it cannot be used for continual release as the output of his linear program at time t can give positive weights to values of x generated at a future time $t'>t$. Furthermore, as instead of computing $L(Rx'+z)$ we work with $t\times t$-dimensional submatrices of L and R, we can replace a $\log T$ factor in the upper bound on the additive error due to the binary mechanism by a $\ln(t)$, where $t\leqslant T$ denotes the current time step.

2. Bounding $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$. The upper bound can be derived in various ways. One direct approach would be to find appropriate Kraus operators of a linear map and then use the characterization by Haagerup & Pisier (1993) of the completely bounded norm. This approach yields an upper bound of $1+\frac{\ln(T)}{\pi}$; however, it does not directly give lower triangular factorization L and R.

Instead, we use the characterization given by Paulsen (1982), which gives us a factorization in terms of lower triangular matrices. More precisely, using three basic trigonometric identities, we show that the $(i,j)^{\text{th}}$ entry of R and L is an integral of every even power of the cosine function, $\frac{2}{\pi}\int_0^{\pi/2}\cos^{2(i-j)}(\theta)\mathrm{d}\theta$ for $i\geqslant j$. This choice of matrices leads to the upper bound in eq. (5).

3. Applications. While counting and averaging under continual release follows from bounds in Theorem 1.1, computing cut-functions requires some ingenuity. In particular, one can consider (S,P)-cuts for an n-vertex

Table 1: Applications of Theorem 1.1 ($\epsilon, \delta \in (0,1)$ are privacy parameter, $\eta \in (0,1)$ is the multiplicative approximation parameter, u is the dimension of each data item for histogram estimation and b a bound on its ℓ_0-sensitivity, and U is the set of letters, ℓ is the maximum length of the substrings that are counted, T is the length of the stream), n is the number of users in local-DP. Here, graph functions include subgraph counting, cost of minimum spanning tree, etc.

Problem	Additive error	Reference
(S,P)-cuts (with $\lvert S\rvert \leqslant \lvert T\rvert$)	$3C_{\epsilon,\delta}\lvert S\rvert\left(1+\frac{\ln(T)}{\pi}\right)\sqrt{(\lvert S\rvert+\lvert P\rvert)\ln(\lvert S\rvert+\lvert P\rvert)\ln(6T)}$	Theorem 5.2
Histogram estimation	$C_{\epsilon,\delta}\left(1+\frac{\ln(T)}{\pi}\right)\sqrt{b\ln(6\lvert U\rvert T)}$	Theorem B.1
Graph functions	$C_{\epsilon,\delta}(1+\frac{\ln(T)}{\pi})\sqrt{\ln(6T)}$	Theorem B.3
Counting all length $\leq \ell$ substrings	$C_{\epsilon,\delta}\left(1+\frac{\ln(T)}{\pi}\right)\ell\sqrt{\ln(6T\lvert U\rvert^{\ell})}$	Theorem B.4
Counting all length $\leq \ell$ episodes	$2C_{\epsilon,\delta}\left(1+\frac{\ln(T)}{\pi}\right)\ell\sqrt{\lvert U\rvert^{\ell-1}\ell\ln(6T\lvert U\rvert^{\ell})}$	Theorem B.6
1-dimensional local convex risk min.	$C_{\frac{\epsilon}{2},\frac{\delta}{2}}\sqrt{\frac{\ln(6(\epsilon\sqrt{n}+1))}{2n}}\left(1+\frac{\ln(\epsilon\sqrt{n}+1)}{\pi}\right)+\frac{2}{\epsilon\sqrt{n}}$	Theorem B.8

graph $\mathcal{G} = (V, E, w)$ as linear queries by constructing a matrix M whose rows correspond to each possible cut query $(S, P) \in V \times V$ and whose columns corresponds to all possible edges in $\mathcal{G}$. The entry $((S, P), j)$ of M equals to 1 if the edge j crosses the boundary of the cut (S, P). However, it is not clear how to use it in the matrix mechanism efficiently because known algorithms for finding a factorization as well as the resulting factorization depend polynomial on the dimension of the matrix and the number of rows in M is $O(2^n)$. Instead, we show how to exploit the algebraic structure of cut functions so that at each time step t the mechanism only has to compute $L_t R(t) x(t)$, where L_t is a $\binom{n}{2} \times t\binom{n}{2}$-dimensional matrix, $R(t)$ is $t\binom{n}{2} \times t\binom{n}{2}$-dimensional matrix and $x(t)$ is $t\binom{n}{2}$-dimensional. This gives a mechanism that has an error $O(\min(|S|, |P|)\ln(t)\sqrt{(|S| + |P|)\ln(|S| + |P|)\ln(6T)}$ (see Theorem 5.2 for exact constant) and can be implemented to run in time $O(tn^4)$ per time step.

Binary counting can also be extended to histogram estimation. Cardoso & Rogers (2021) gave a differentially private mechanism for histogram estimation where exactly one item either arrives or is removed in each time epoch. We extend our approach to work in the setting that they call *known-domain, restricted ℓ_0-sensitivity* and improve the constant factor in the additive error by the same amount as for binary counting.

We also show in Section B.5 an application of our mechanism to non-interactive local learning. The non-interactive algorithm for local convex risk minimization is an adaption of the algorithm of Smith et al. (2017), which uses the binary tree mechanism for binary counting as a subroutine. Replacing it by our mechanism for binary counting (Theorem 1.2) leads to various technical challenges: From the algorithmic design perspective, Smith et al. (2017) used the binary mechanism with a randomization routine from Duchi et al. (2013), which expects as input a binary vector, while we apply randomization to Rx, where R has real-valued entries. We overcome this difficulty by using two instead of one binary counter. From an analysis point of view, the error analysis in Smith et al. (2017) is based on the error analysis in Bassily & Smith (2015) that uses various techniques, such as the randomizer of Duchi et al. (2013), error-correcting codes, and the Johnson-Lindenstrauss lemma. However, one can show that we can give the same *uniform approximation* (see Definition 5 in Smith et al. (2017) for the formal definition) as in Smith et al. (2017) by using the Gaussian mechanism and two binary counters so that the rest of their analysis applies.

2. Notations and Preliminaries

We use $v[i]$ to denote the ith coordinate of a vector v. For a matrix A, we use $A[i, j]$ to denote its $(i, j)^{\text{th}}$ coordinate, $A[:, i]$ to denote its ith column, $A[i, :]$ to denote its ith row, $\|A\|_{\mathsf{tr}}$ to denote its trace norm of square matrix, $\|A\|_F$ to denote its Frobenius norm, $\|A\|$ to denote its operator norm, and $A^\top$ to denote transpose of A. We use $\mathbb{I}_d$ to denote the identity matrix of dimension d. If all the eigenvalues of a symmetric matrix $S \in \mathbb{R}^{d\times d}$ are non-negative, then the matrix is known as *positive semidefinite* (PSD for short) and is denoted by $S \succeq 0$. For symmetric matrices $A, B \in \mathbb{R}^{d\times d}$, the notations $A \preceq B$ implies that $B - A$ is PSD. For an $a_1 \times a_2$ matrix A, its *tensor product* (or *Kronecker product*) with another matrix B is

$$\begin{pmatrix} A[1,1]B & A[1,2]B & \cdots & A[1,a_2]B \\ A[2,1]B & A[2,2]B & \cdots & A[2,a_2]B \\ \vdots & \ddots & \vdots & \\ A[a_1,1]B & A[a_1,2]B & \cdots & A[a_1,a_2]B \end{pmatrix}.$$

We use $A \otimes B$ to denote the tensor product of A and B. In our case, the matrix B would always be the identity matrix of the appropriate dimension.

One main application of our results is in *differential privacy* formally defined below:

Definition 2.1. A randomized mechanism $\mathcal{M}$ gives (ϵ, δ)-differential privacy if for all *neighboring* data sets D and D' in the domain of $\mathcal{M}$ differing in at most one row, and all measurable subset S in the range of $\mathcal{M}$, $\mathsf{Pr}\left[\mathcal{M}(D) \in S\right] \leqslant e^{-\varepsilon}\mathsf{Pr}\left[\mathcal{M}(D') \in S\right] + \delta$, where the probability is over the private coins of $\mathcal{M}$.

This definition requires, however, to define *neighboring* data sets in the continual release model. In this model, the data is given as a *stream* of individual data items, each belonging to a unique user, each arriving one after the other, one per time step. In the privacy literature, there are two well-studied notions of neighboring streams (Chan et al., 2011; Dwork et al., 2010): (i) *user-level privacy*, where two streams are neighboring if they differ in potentially all data items of a single user; and (ii) *event-level privacy*, where two streams are neighboring if they differ in a single data item in the stream. We here study event-level privacy.

Our algorithm uses the Gaussian mechanism. To define the Gaussian mechanism, we need to first define ℓ_2-*sensitivity*. For a function $f : \mathcal{X}^n \to \mathbb{R}^d$ its ℓ_2-sensitivity is defined as

$$\Delta f := \max_{\text{neighboring } X, X' \in \mathcal{X}^n} \|f(X) - f(X')\|_2 . \quad (7)$$

Definition 2.2 (Gaussian mechanism)**.** Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function with ℓ_2-sensitivity Δf. For a given $\epsilon, \delta \in (0, 1)$ the Gaussian mechanism $\mathcal{M}$, which given $X \in \mathcal{X}^n$ returns $\mathcal{M}(X) = f(X) + e$, where $e \sim \mathcal{N}(0, C^2_{\epsilon,\delta}(\Delta f)^2 \mathbb{I}_d)$, satisfies (ϵ, δ)-differential privacy.

Definition 2.3 (Accuracy)**.** A mechanism $\mathcal{M}$ is (α, T)-*accurate* for a function f if, for all finite input streams x of length T, the maximum absolute error satisfies $||f(x) - \mathcal{M}(x)||_\infty \leqslant \alpha$ with probability at least $2/3$ over the coins of the mechanism.

3. Proof of Theorem 1.1

We prove the following in Appendix A.

Lemma 3.1. *Let M_{count} be the matrix defined in eq. (1) and L and R be as in Algorithm 1. Then $M_{\mathsf{count}} = LR$.*

The proof of Theorem 1.1 relies on the following lemmas.

Lemma 3.2 (Chen & Qi (2005))**.** *For integer $m > 0$, define $\mathcal{S}_m := \left(\frac{1}{2}\right)\left(\frac{3}{4}\right)\cdots\left(\frac{2m-1}{2m}\right)$. Then, $\mathcal{S}_m \leqslant \sqrt{\frac{1}{\pi(m+1/4)}}$.*

Proof of Theorem 1.1. Let L and R be $T \times T$ matrices defined in Algorithm 1. Then we show the following:

Lemma 3.3. $\|L\|_{1\to 2} = \|L\|_{2\to\infty} = \|R\|_{1\to 2} = \|R\|_{2\to\infty}$. *Further,*

$$\|L\|^2_{2\to\infty} \leqslant 1 + \sum_{i=2}^{T} \frac{1}{\pi i} \leqslant 1 - \frac{1-\gamma}{\pi} + \frac{\ln(T)}{\pi} + \frac{2}{T}.$$

Algorithm 1 FACTORIZATION(M_{count})

1: Define a function $f : \mathbb{Z}_+ \to \mathbb{R}$, recursively as follows

$$f(0) = 1 \text{ and } f(k) = \left(\frac{2k-1}{2k}\right) f(k-1);\ k \geqslant 1.$$

2: Let the (i, j)-th entry of L and R be defined as $L[i, j] = R[i, j] = f(i - j)$.

3: **Output** L, R as the factorization

Proof. Using Lemma 3.2, we have

$$\|L\|^2_{2\to\infty} \leqslant 1 + \sum_{i=2}^{T} \frac{1}{\pi i} \leqslant 1 - \frac{1-\gamma}{\pi} + \frac{\ln(T)}{\pi} + \frac{2}{T} - \alpha_n,$$

where γ is the Euler-Mascheroni constant ($\gamma \leqslant 0.578$) and $0 \leqslant \alpha_n \leqslant 1/8T^2$. Since α_n is positive and $\|L\|_{2\to\infty} = \|L\|_{1\to 2}$ and $R = L$. □

Theorem 1.1 follows from Lemma 3.1 and 3.3. □

4. Proof of Theorem 1.2

Fix a time $t \leqslant T$. Let L_t denote the $t \times t$ principal submatrix of L and R_t be the $t \times t$ principal submatrix of R. Let the vector formed by the streamed bits be $x_t = \begin{pmatrix} x[1] & \cdots & x[t] \end{pmatrix} \in \{0, 1\}^t$. Let $z_t = \begin{pmatrix} z[1] & \cdots & z[t] \end{pmatrix}$ be a freshly sampled Gaussian vector such that $z[i] \sim \mathcal{N}(0, C^2_{\epsilon,\delta} \|R_t\|^2_{1\to 2})$. Let $M_{\mathsf{count}}(t)$ denote the $t \times t$ principal submatrix of M_{count}. The algorithm computes

$$\widetilde{x}_t = L_t(R_t x_t + z_t) = L_t R_t x_t + L_t z_t = M_{\mathsf{count}}(t) x_t + L_t z_t$$

and outputs the tth coordinate of $\widetilde{x}_t$ (denoted by $\tilde{x}_t[t]$). So far, the data structure maintained by the mechanism when it enters round t is simply x_{t-1}. Now, note that the tth coordinate of $M_{\mathsf{count}}(t)x_t$ equals the sum of the first t bits and can be computed in constant time when the sum of the bits of the previous round $t - 1$ is known (i.e., maintained). Then, sampling a fresh Gaussian vector z_t and computing the tth coordinate of $L_t z_t$ takes time $O(t)$. For privacy, note that the ℓ_2-sensitivity of $R_t x_t$ is $\|R_t\|_{1\to 2}$; therefore, adding Gaussian noise with variance $\sigma^2_t = C^2_{\epsilon,\delta} \|R_t\|^2_{1\to 2}$ preserves (ϵ, δ)-differential privacy. Now for the accuracy guarantee,

$$\left|\widetilde{x}_t[t] - \sum_{i=1}^{t} x[i]\right| = \left|\sum_{i=1}^{t} L_t[t, i] z_t[i]\right|.$$

because $\widetilde{x}_t[t] = \sum_{i=1}^{t} x[i] + \sum_{i=1}^{t} L_t[t, i] z_t[i]$.

Recall that $z[i] \sim \mathcal{N}(0, \sigma^2_t)$. The Cauchy-Schwarz inequality shows that the function $\ell(z_t) := \sum_{i=1}^{t} L_t[t, i] z[i]$ has Lipschitz constant $\|L_t\|_{2\to\infty}$, i.e., the maximum row norm.

Now define $z'[i] := z[i]/\sigma_t$ and note that $z'[i] \sim \mathcal{N}(0,1)$ and $\mathbb{E}[\ell(z'_t)] = \mathbb{E}[\ell(z_t)] = 0$. Using the concentration inequality for Gaussian random variables with unit variance and a function f with Lipschitz constant $\|L_t\|_{2\to\infty}$ (e.g., Proposition 4 in Zeitouni (2016)) implies that

$$\begin{aligned}&\mathsf{Pr}_{z_t}\left[|\ell(z_t) - \mathbb{E}[\ell(z_t)]| > a\right]\\ &= \mathsf{Pr}_{z_t}\left[|\ell(z'_t) - \mathbb{E}[\ell(z'_t)]| > a/\sigma_t\right] \leqslant 2e^{-a^2/(2\sigma_t^2\|L_t\|_{2\to\infty}^2)}.\end{aligned}$$

The result follows using Theorem 1.1, which implies that $\|L_t\|_{2\to\infty} = \|R_t\|_{1\to 2} \leq \sqrt{\Psi(t)}$.

5. Application I: Continually Releasing a Synthetic Graph for Answering Cut Queries

For a weighted graph $\mathcal{G} = (V, E, w)$, we let n denote the size of the vertex set V and m denote the size of the edge set E, and $w : V \times V \to \mathbb{R}_{\geqslant 0}$ is the weight function denoting the weight of the edge $(u, v) \in V \times V$. When the graph is uniformly weighted (i.e., all existing edges have the same weight, all non-existing have weight 0), then the graph is denoted $\mathcal{G} = (V, E)$.

One important matrix representation of a graph is its *Laplacian* (or *Kirchhoff matrix*). For a graph $\mathcal{G}$, its Laplacian $K_{\mathcal{G}}$ is the matrix form of the negative discrete Laplace operator on a graph that approximates the negative continuous Laplacian obtained by the finite difference method.

Definition 5.1 ((S, P)-cut)**.** For two disjoint subsets S and P, the size of the cut (S, P)-cut, $\Phi_{S,P}(\mathcal{G})$, is

$$\Phi_{S,P}(\mathcal{G}) := \sum_{u\in S, v\in P} w(u, v).$$

When $P = V\backslash S$, we denote $\Phi_{S,P}(\mathcal{G})$ by $\Phi_S(\mathcal{G})$.

In this section, we study the following problem. Let $\mathcal{G} = (V, E, w)$ be a weighted graph and consider a sequence of T updates to the edges of $\mathcal{G}$, where each update consists of an (edge, weight) tuple with weight in $[0, 1]$ that adds the weight to the corresponding edge. For t, $1 \leqslant t \leqslant T$, let $\mathcal{G}_t$ denote the graph that is obtained by applying the first t updates to $\mathcal{G}$. We give a differentially private mechanism that returns after each update t a graph $\overline{\mathcal{G}}_t$ such that for every cut (S, P) with $S\cap P = \emptyset$, the number of edges crossing the cut in $\overline{\mathcal{G}}_t$ differs from the number of edges crossing the same cut in $\mathcal{G}_t$ by at most $O(\min\{|S|, |P|\}\sqrt{n \ln n} \ln^{3/2} T)$:

Corollary 5.2. *There is an (ϵ, δ)-differentially private algorithm that, for any stream of edge updates of length $T > 0$, outputs a synthetic graph $\overline{\mathcal{G}}_t$ in round t in time $O(tn^4)$, and with probability at least 2/3 over the coin tosses of the algorithm, simultaneously for all rounds t with $1 \leq t \leq T$, it holds that for any $S, P \subset V$ with $S \cap P = \emptyset$, $|\Phi_{S,P}(\overline{\mathcal{G}}_t) - \Phi_{S,P}(\mathcal{G}_t)|$ is at most*

$$3C_{\epsilon,\delta}|S|\psi(t)\sqrt{(|S| + |P|)\ln(|S| + |P|)\ln(6T)},$$

where $\mathcal{G}_t$ is the graph formed at time t through edge updates and $C_{\epsilon,\delta}$ is as defined in eq. (2).

Proof. Let us first analyze the case where $P = V \setminus S$. In this case, we encode the updates as an $\mathbb{R}^{\binom{n}{2}}$ vector and consider the following counting matrix:

$$M_{\mathsf{cut}} = M_{\mathsf{count}} \otimes \mathbb{I}_{\binom{n}{2}} \in \{0, 1\}^{T\binom{n}{2}\times T\binom{n}{2}}$$

For the rest of this subsection, we drop the subscript and denote $\mathbb{I}_{\binom{n}{2}}$ by $\mathbb{I}$. Recall the function f defined by **??**. Let $L_{\mathsf{count}}[i, j] = f(i-j)$. Using this function, we can compute the following factorization of M_{cut}: $L = L_{\mathsf{count}}\otimes\mathbb{I}$ and $R = L$. Let $R(t)$ and $L(t)$ denote the $t\binom{n}{2} \times t\binom{n}{2}$ principal submatrix of R and L, respectively. They both consist of t *block matrices*, each being formed by all columns and $\binom{n}{2}$ rows of R and L, respectively. Further, let R_t and L_t denote the tth block matrix of $\binom{n}{2}$ rows of R and L, respectively. Let $x(t)$ be the $t \times \binom{n}{2}$ vector formed by the first t updates, i.e., the edges of $\mathcal{G}_t$ which are given by the $\binom{n}{2}$ vector $E_{\mathcal{G}_t} := L_t R(t)x(t)$.

Let $C_{\epsilon,\delta}$ be the function of ϵ and δ stated in eq. (6) and $\sigma^2 = C^2_{\epsilon,\delta}\|R_t\|^2_{1\to 2}\sqrt{\ln(6T)}$. Then the edges of the weighted graph $\overline{\mathcal{G}}_t$ which is output at time t are given by the $\binom{n}{2}$ vector $L_t\left(R(t)x(t) + z\right)$, where $z \sim \mathcal{N}(0, \sigma^2)^{t\times\binom{n}{2}}$. Note that computing the output naively takes time $O(t^2n^4)$ to compute $R(t)x(t)$, time $O(tn^2)$ to generate and add z, and time $O(tn^4)$ to multiply the result with L_t. However, if we store the vector of $R(t-1)x(t-1)$ of the previous round and only compute $R_tx(t)$ in round t, then the vector $R(t)x(t)$ can be created by "appending" $R_tx(t)$ to the vector $R(t-1)x(t-1)$. Thus, $R(t)x(t)$ can be computed in time $O(tn^4)$. This reduces the computation time at time step t to $O(tn^4)$.

We next analyze the additive error of this mechanism. The output at time t of the algorithm is a vector indicating the edges as $E_{\overline{\mathcal{G}}_t} = L_t(R(t)x(t) + z) = E_{\mathcal{G}_t} + L_tz$. Let $f_t = \begin{pmatrix} f(t-1) & \cdots & f(0)\end{pmatrix}^\top \in \mathbb{R}^t$ be a row vector whose coordinates are the evaluations of the function $f(\cdot)$ on $\{0, 1, \cdots, t-1\}$. Then $L_t = f_t \otimes \mathbb{I}$.

As L_tz is the weighted sum of random variables from $N(0, \sigma^2)$, it holds that $L_tz \sim N(0, \|L_t\|^2_{2\to\infty}\sigma^2\mathbb{I}_{\binom{n}{2}})$. In other words, the error is due to a graph, $\mathcal{R}$, with weights sampled from a Gaussian distribution $N(0, C^2_{\epsilon,\delta}\|L_t\|^2_{2\to\infty}\|R(t)\|^2_{1\to 2}\mathbb{I})$.

For a subset $S \subseteq [n]$, let $\chi_S = \sum_{i\in S}\overline{e}_i$, where $\overline{e}_i$ is the ith standard basis. It is known that for any positive weighted graph $\mathcal{G}$, the $(S, V\backslash S)$-cut $\Phi_S(\mathcal{G}) = \chi_S^\top K_{\mathcal{G}}\chi_S$. So, for any subset $S \subseteq [n]$, $|\Phi_S(\overline{\mathcal{G}}_t) - \Phi_S(\mathcal{G}_t)| \leqslant \sqrt{\Psi(t)}\left|\chi_S^\top K_{\mathcal{R}}\chi_S\right|$.

The proof now follows on the same line as in Upadhyay et al. (2021). In more detail, if K_n denotes the Laplacian of the

complete graph with n vertices, then $K_{\mathcal{R}} \preceq 3\sigma\sqrt{\frac{\ln(n)}{n}}K_n$. Here, $A \preceq B$ means that $x^\top(B-A)x \geqslant 0$ for all x. Setting $\sigma = 3C_{\epsilon,\delta}\Psi(t)\sqrt{\ln(6T)}$, the union bound gives that with probability at least $2/3$, simultaneously for all rounds T,

$$\begin{aligned}\left|\chi_S^\top K_{\mathcal{R}}\chi_S\right| &\leqslant 3\sigma\sqrt{\frac{\ln(n)}{n}}\left|\chi_S^\top L_n\chi_S\right| \\ &= 3\sigma|S|\,(n-|S|)\sqrt{\frac{\ln(n)}{n}} \leqslant 3\sigma\sqrt{n\ln(n)}|S| \\ &= 3C_{\epsilon,\delta}|S|\Psi(t)\sqrt{n\ln(n)\ln(6T)}. \qquad (8)\end{aligned}$$

We next consider the case of (S, P) cuts, where $S \cup P \subseteq V$ and $S \cap P = \emptyset$. Without loss of generality, let $|S| \leqslant |P|$. Let us denote by $\mathcal{G}_A$ the graph induced by a vertex set $A \subseteq V$. Now, consider the subgraph, $\mathcal{G}_{S\cup P}$, formed by the vertex set $S \cup P$. By Fiedler (1973)'s result, $s_i(\mathcal{G}_{S\cup P}) \leqslant s_i(\mathcal{G}_V)$, where $s_i(\mathcal{H})$ denotes the ith singular value of the Laplacian of the graph $\mathcal{H}$. Considering this subgraph, we have reduced the analysis of (S, P) cut on $\mathcal{G}$ to the analysis of $(S, \overline{S})$-cut on $\mathcal{G}_{S\cup P}$. Therefore, using the previous analysis, we get the result. The privacy proof is in given in Appendix B. □

Remark 5.3. In the worst case when $|S| = cn$ for some constant $c > 0$, this results in an additive error of order $n^{3/2}\sqrt{\ln n}\ln^{3/2}(T)$. This result gives a mechanism for maintaining the minimum cut as well as a mechanism for maintaining the maximum cut, sparsest cuts, etc with such an additive error. Moreover, we can extend the result to receive updates with weights in $[-1, 1]$ as long as the underlying graph only has positive weights at all times.

For maintaining the minimum cut in the continual release model we show in Appendix C that our upper bound is tight up to polylogarithmic factors in n and T for large enough T and constant S using a reduction from a lower bound in Jain et al. (2021).

6. Conclusion

We study the problem of binary counting under continual release. The motivation for this work is that (1) only an asymptotic analysis is known for the additive error for the classic mechanism (known as the binary mechanism) for binary counting under continual release, and (2) in practice the additive error of the binary mechanism is very non-smooth, which hampers its practical usefulness.

We first observe that the matrix mechanism can actually be used for binary counting in the continual release model *if the factorization uses lower-triangular matrices.* Then we give an explicit factorization for M_{count} that fulfill the following properties:

(1) We improved a 28-year-old result on $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$ for a large value of T to give an analysis of the additive error that only has a small gap between the upper and lower bound for the counting problem. This means that the behavior of the additive error is very well understood. The additive error is also a monotonic smooth function of the number of updates performed so far. In contrast, the error of the binary mechanism changes non-smoothly over time, making it less interpretable and reliable. We show that all these properties are not just theoretical advantages, but also make a big difference in the implementation (see Figure 3).

(2) The factorization for the binary mechanism consists of two lower-triangular matrices with exactly T distinct non-zero entries that follow a simple pattern so that only $O(T)$ space is needed.

(3) Our algorithm is very simple to implement, consisting of a matrix-vector multiplication and the addition of two vectors. Simplicity is an important design principle in large-scale deployment as it helps to reduce the points of vulnerabilities in a system. This is also one main reason for us to pick the Gaussian mechanism: it is easy to implement with floating point arithmetic while maintaining the provable guarantee of privacy. Further, the privacy guarantee can be easily stated in terms of concentrated-DP or Renyi-DP.

Finally, we show that our bounds have diverse applications that range from binary counting to maintaining histograms, various graph functions, outputting a synthetic graph that maintains the value of all cuts, substring counting, and episode counting. We believe that there are more applications of our mechanism, and this work will bring more attention to leading constants in the analysis of differentially private algorithms.

Acknowledgements. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019564 "The Design of Modern Fully Dynamic Data Structures (MoDynStruct)" and from the Austrian Science Fund (FWF) project Z 422-N, and project "Fast Algorithms for a Reactive Network Layer (ReactNet)", P 33775-N, with additional funding from the *netidee SCIENCE Stiftung*, 2020–2024.

2020–2024. JU's research was funded by Decanal Research Grant. A part of this work was done when JU was visiting Indian Statistical Institute, Delhi. The authors would like to thank Rajat Bhatia, Aleksandar Nikolov, Shanta Laisharam, Vern Paulsen, Ryan Rogers, Abhradeep Thakurta, and Sarvagya Upadhyay for useful discussions.

References

Alabi, D., Ben-Eliezer, O., and Chaturvedi, A. Bounded space differentially private quantiles. *arXiv preprint arXiv:2201.03380*, 2022.

Apple. https://covid19.apple.com/contacttracing, 2021.

Asi, H., Feldman, V., Koren, T., and Talwar, K. Private stochastic convex optimization: Optimal rates in l1 geometry. In *International Conference on Machine Learning*, pp. 393–403. PMLR, 2021.

Asi, H., Feldman, V., Koren, T., and Talwar, K. Private online prediction from experts: Separations and faster rates. *arXiv preprint arXiv:2210.13537*, 2022a.

Asi, H., Feldman, V., and Talwar, K. Optimal algorithms for mean estimation under local differential privacy. *arXiv preprint arXiv:2205.02466*, 2022b.

Balle, B. and Wang, Y.-X. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pp. 394–403. PMLR, 2018.

Bassily, R. and Smith, A. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 127–135. ACM, 2015.

Bolot, J., Fawaz, N., Muthukrishnan, S., Nikolov, A., and Taft, N. Private decayed predicate sums on streams. In *Proceedings of the 16th International Conference on Database Theory*, pp. 284–295. ACM, 2013.

Böttcher, A. and Grudsky, S. M. *Toeplitz matrices, asymptotic linear algebra and functional analysis*, volume 67. Springer, 2000.

Bun, M., Nissim, K., Stemmer, U., and Vadhan, S. Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 634–649. IEEE, 2015.

Bun, M., Nelson, J., and Stemmer, U. Heavy hitters and the structure of local privacy. *ACM Transactions on Algorithms*, 15(4):51, 2019.

Cardoso, A. and Rogers, R. Differentially private histograms under continual observation: Streaming selection into the unknown. *arXiv preprint arXiv:2103.16787*, 2021.

CDC. https://www.cdc.gov/coronavirus/2019-ncov/daily-life-coping/contact-tracing.html, 2020.

Chan, T. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, 2011. doi: 10.1145/2043621.2043626. URL http://doi.acm.org/10.1145/2043621.2043626.

Chan, T.-H. H., Li, M., Shi, E., and Xu, W. Differentially private continual monitoring of heavy hitters from distributed streams. In *International Symposium on Privacy Enhancing Technologies Symposium*, pp. 140–159. Springer, 2012.

Chen, C.-P. and Qi, F. The best bounds in wallis' inequality. *Proceedings of the American Mathematical Society*, 133 (2):397–401, 2005.

Choquette-Choo, C. A., McMahan, H. B., Rush, K., and Thakurta, A. Multi-epoch matrix factorization mechanisms for private machine learning. *arXiv preprint arXiv:2211.06530*, 2022.

Davidson, K. R. Similarity and compact perturbations of nest algebras. 1984.

Denisov, S., McMahan, B., Rush, K., Smith, A., and Thakurta, A. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. *arXiv preprint arXiv:2202.08312*, 2022.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 429–438. IEEE, 2013.

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

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pp. 715–724, 2010. doi: 10.1145/1806689.1806787. URL http://doi.acm.org/10.1145/1806689.1806787.

Dwork, C., Naor, M., Reingold, O., and Rothblum, G. N. Pure differential privacy for rectangle queries via private partitions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 735–751. Springer, 2015.

Edmonds, A., Nikolov, A., and Ullman, J. The power of factorization mechanisms in local and central differential privacy. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 425–438, 2020.

Epasto, A., Mao, J., Medina, A. M., Mirrokni, V., Vassilvitskii, S., and Zhong, P. Differentially private continual releases of streaming frequency moment estimations. *arXiv preprint arXiv:2301.05605*, 2023.

Fichtenberger, H., Henzinger, M., and Ost, W. Differentially private algorithms for graphs under continual observation. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, 2021.

Fiedler, M. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

Haagerup, U. Decomposition of completely bounded maps on operator algebras, 1980.

Haagerup, U. and Pisier, G. Bounded linear operators between c^*-algebras. *Duke Mathematical Journal*, 71(3): 889–925, 1993.

Henzinger, M., Upadhyay, J., and Upadhyay, S. Almost tight error bounds on differentially private continual counting. *SODA*, 2023.

Honaker, J. Efficient use of differentially private binary trees. *Theory and Practice of Differential Privacy (TPDP 2015), London, UK*, 2015.

Huang, Z., Qiu, Y., Yi, K., and Cormode, G. Frequency estimation under multiparty differential privacy: One-shot and streaming. *arXiv preprint arXiv:2104.01808*, 2021.

Jain, P., Raskhodnikova, S., Sivakumar, S., and Smith, A. The price of differential privacy under continual observation. *arXiv preprint arXiv:2112.00828*, 2021.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pp. 5213–5225. PMLR, 2021.

Li, C., Miklau, G., Hay, M., McGregor, A., and Rastogi, V. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB journal*, 24 (6):757–781, 2015.

Lin, S., Qiao, J., and Wang, Y. Frequent episode mining within the latest time windows over event streams. *Applied intelligence*, 40(1):13–28, 2014.

Mathias, R. The hadamard operator norm of a circulant and applications. *SIAM journal on matrix analysis and applications*, 14(4):1152–1167, 1993.

Matoušek, J., Nikolov, A., and Talwar, K. Factorization norms and hereditary discrepancy. *International Mathematics Research Notices*, 2020(3):751–780, 2020.

McKenna, R., Miklau, G., Hay, M., and Machanavajjhala, A. Hdmm: Optimizing error of high-dimensional statistical queries under differential privacy. *arXiv preprint arXiv:2106.12118*, 2021.

Muthukrishnan, S. and Nikolov, A. Optimal private halfspace counting via discrepancy. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 1285–1292, 2012.

Paulsen, V. I. Completely bounded maps on c^*-algebras and invariant operator ranges. *Proceedings of the American Mathematical Society*, 86(1):91–96, 1982.

Paulsen, V. I. *Completely bounded maps and dilations*. New York, 1986.

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

Schur, J. Remarks on the theory of restricted "a ned bilinear forms with infinitely many ä mutable ones. 1911.

Smith, A., Thakurta, A., and Upadhyay, J. Is interaction necessary for distributed private learning? In *IEEE Symposium on Security and Privacy*, 2017.

Thakurta, A. Differential privacy: From theory to deployment, https://www.youtube.com/watch?v=Nvy-TspgZMs&t=2320s&ab_channel=USENIX, 2017.

Upadhyay, J. Sublinear space private algorithms under the sliding window model. In *International Conference on Machine Learning*, pp. 6363–6372, 2019.

Upadhyay, J. and Upadhyay, S. A framework for private matrix analysis in sliding window model. In *International Conference on Machine Learning*, pp. 10465–10475. PMLR, 2021.

Upadhyay, J., Upadhyay, S., and Arora, R. Differentially private analysis on graph streams. In *International Conference on Artificial Intelligence and Statistics*, pp. 1171–1179. PMLR, 2021.

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

Zeitouni, O. Gaussian fields, 2016.

Zipf, G. K. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.

A. Proof of Lemma 3.1

One way to prove the lemma is by using trigonometric inequalities for $\cos(2\theta)$ and that (i) For any $\theta \in [-\pi, \pi]$, $\sin^2(\theta) + \cos^2(\theta) = 1$, (ii) For even m, $\frac{2}{\pi}\int_0^{\pi/2} \cos^m(\theta)\mathrm{d}\theta = \left(\frac{1}{2}\right)\left(\frac{3}{4}\right)\cdots\left(\frac{m-1}{m}\right)$, (iii) For all $\theta \in [-\pi, \pi]$, $\cos(2\theta) = 2\cos^2(\theta) - 1$. This proof would require a lot of algebraic manipulations. However, our proof relies on an observation that all the three matrices, L, R, and M_{count} are $T \times T$ principal submatrix of Toeplitz operators. The main idea is to represent the Toeplitz operator in its functional form: Let $a_1, a_2, \cdots, \in \mathbb{C}$ denote the entries of the Toeplitz operator, $\mathcal{T}$, with complex entries. Then its unique associated symbol is

$$f_{\mathcal{T}}(z) = \sum_{n=0}^{\infty} a_n z^n,$$

where $z \in \mathbb{C}$ such that $|z| = 1$. We can then write $z = e^{\iota\theta}$ for $0 \leqslant \theta \leqslant 2\pi$. Then for operator $\mathcal{M}$ whose principal $T \times T$ submatrix is the matrix M_{count}, we arrive that its associated symbol is

$$f_{\mathcal{M}}(\theta) = \sum_{n=0}^{\infty} e^{\iota\theta n} = \left(1 - e^{\iota\theta}\right)^{-1}.$$

Let $\mathcal{L}$ denote the Toeplitz operator whose $T \times T$ submatrix is the matrix L and $\mathcal{R}$ denote the Toeplitz operator whose $T \times T$ submatrix is the matrix R. Then the symbol associated with $\mathcal{L}$ and $\mathcal{R}$ is

$$f_{\mathcal{R}}(\theta) = f_{\mathcal{L}}(\theta) = \sum_{n=0}^{\infty} f(n) e^{\iota\theta n} = 1 + \frac{1}{2}e^{\iota\theta} + \frac{3}{8}e^{2\iota\theta} + \cdots \tag{9}$$

Now $(1-x)^{-1/2} = 1 + \frac{1}{2}x + \frac{3}{8}x^2 + \cdots$. Therefore, comparing the terms, we can rewrite eq. (9) as follows:

$$f_{\mathcal{L}}(\theta) = \left(1 - e^{\iota\theta}\right)^{-1/2} \quad \text{and} \quad f_{\mathcal{R}}(\theta) = \left(1 - e^{\iota\theta}\right)^{-1/2}.$$

Since, for any two Toeplitz operators, $\mathcal{A}$ and $\mathcal{B}$ with associated symbols $f_{\mathcal{A}}(\theta)$ and $f_{\mathcal{B}}(\theta)$, respectively, $\mathcal{AB}$ has the associated symbol $f_{\mathcal{A}}(\theta) f_{\mathcal{B}}(\theta)$ (Böttcher & Grudsky, 2000), we have the lemma.

B. Applications in Continual Release

Privacy proof of Theorem 5.2. We now give the privacy proof. At the time t, we only need to prove privacy for $R(t)x(t) + z$ as multiplication by L_t can be seen as post-processing. Now consider two neighboring graphs formed by the stream of (edge, weight) tuple, where at time τ, they differ in weight. If $t < \tau$, the output distribution is the same because the input is the same. So, consider $t \geqslant \tau$. At this point, $x(t)$ and $x'(t)$ differ in exactly $t - \tau$ positions by at most 1. Breaking $x(t) - x'(t)$ in blocks of $\binom{n}{2}$ coordinates, the position where $x(t) - x'(t)$ is 1 is exactly corresponding to the edge they differ. Now multiplying with $R(t)$ results in a vector whose non-zero entries are $\{f(\tau), \cdots, f(t)\}$. Using Theorem 3.3, $\|R(t)(x(t) - x'(t))\|_2^2 = \|R_t\|_2^2 - \|R_\tau\|_2^2 \leqslant \|R_t\|_2^2$. Therefore, we have (ϵ, δ)-differential privacy from the choice of σ and Theorem 2.2.

B.1. Continual Histogram

Modifying the analysis for cut functions, we can use our algorithm to compute the histogram of each column for a database of u-dimensional binary vectors in the continual release model in a very straightforward manner. Said differently, assume $\mathcal{U}$ is a universe of size u and the input at a time step consists of the indicator vector of a subset of $\mathcal{U}$, which is a u-dimensional binary vector. Let b with $1 \leq b \leq u$ be the maximum number of 1s in the vector, i.e., the maximum size of a subset given at a time step.

Corollary B.1. *Let $\mathcal{U}$ be the universe of size u and let $1 \leq b \leq u$ be a given integer. Consider a stream of T vectors $x_t \in \{0,1\}^u$ such that $x_t[j] = 1$ if $j \in \mathcal{S}$ and $x_t[k] = 0$ for all $k \notin \mathcal{S}$ where at time t the subset $\mathcal{S} \subseteq \mathcal{U}$ with $|\mathcal{S}| \leq b$ is streamed. Then there is an efficient (ϵ, δ)-differentially private algorithm which outputs a vector $h_t \in \mathbb{R}^u$ in round t such that, with probability at least $1 - \beta$ over the coin tosses of the algorithm, simultaneously, for all rounds t with $1 \leq t \leq T$, it*

holds that

$$\left\| h_t - \sum_{i=1}^{t} x_i \right\|_\infty \leqslant C_{\epsilon,\delta} \Psi(t) \sqrt{b \ln(2uT/\beta)}.$$

The same bounds hold if items can also be removed, i.e., $x_t \in \{-1, 0, 1\}^u$ *as long as* $\sum_{i=1}^{t} x_i[j] \geq 0$ *for all* $1 \leqslant j \leqslant u$ *and all* $t \leqslant T$.

Proof. We consider the following matrix: $M_{\mathsf{hist}} = M_{\mathsf{count}} \otimes \mathbb{I}_u$ with every update being the indicator vector in $\{0,1\}^u$. We drop the subscript on $\mathbb{I}$ and denote $\mathbb{I}_u$ by $\mathbb{I}$ in the remainder of this subsection. Recall the function f defined by **??**. Let $L_{\mathsf{count}}[i,j] = f(i-j)$. Using this function, we can compute the following factorization of M_{hist}: $L = L_{\mathsf{count}} \otimes \mathbb{I}$ and $R = L$. Let $R(t)$, resp. $L(t)$, be the $tu \times tu$ principal submatrix of R, resp. L. Let R_t, resp. L_t, be the tth block matrix of $R(t)$, resp. $L(t)$, consisting of all columns and the last u rows. Then at any time epoch we output $h_t = L_t(R(t)x(t) + z_t)$, where $x(t) \in \{0,1\}^{tu}$ is the row-wise stacking of $x_1, \cdots, x_t$ and $z_t[i] \sim N(0, \sigma_t^2)$ for $\sigma_t^2 = C_{\epsilon,\delta}^2 b \left\| R_t \right\|_{1\to 2}^2$. For privacy, note that the ℓ_2-sensitivity of $R_t x_t$ is $\sqrt{b} \left\| R_t \right\|_{1\to 2}$; therefore, adding Gaussian noise with variance $\sigma_t^2 = C_{\epsilon,\delta}^2 b \left\| R_t \right\|_{1\to 2}^2$ preserves (ϵ, δ)-differential privacy.

Using the same proof as in the case of M_{count} we obtain

$$\left\| h_t - \sum_{i=1}^{t} x_i \right\|_\infty \leqslant C_{\epsilon,\delta} \left\| R_T \right\|_{1\to 2} \left\| L_T \right\|_{2\to\infty} \sqrt{b \ln(2uT/\beta)}.$$

Using Theorem 1.1, we have the corollary. □

B.2. Other Graph Functions

Our upper bounds can also be applied to continual release algorithms that use the binary mechanism to compute prefix sums. Let $f_1, f_2, \ldots, f_T$ be a sequence σ of T function values. The *difference sequence of* σ is $f_2 - f_1, f_3 - f_2, \ldots, f_T - f_{T-1}$. For a graph function under continual release, its sensitivity may depend on the allowed types of edge updates. Fichtenberger et al. (2021) show that the ℓ_1-sensitivity of the difference sequence of the cost of a minimum spanning tree, degree histograms, triangle count and k-star count does not depend on T for partially dynamic updates (either edge insertions or deletions) and is $\Omega(T)$ for fully dynamic updates (edge insertions and/or deletions). Using this result, they prove that one can privately compute these graph functions under continual observation by releasing noisy partial sums of the difference sequences of the respective functions. More generally, they show the following result for any graph function with bounded sensitivity of the difference sequence.

Lemma B.2 (Fichtenberger et al. (2021), cf Corollary 13)**.** *Let* f *be a graph function whose difference sequence has* ℓ_1*-sensitivity* Γ. *Let* $0 < p < 1$ *and* $\varepsilon > 0$. *For each* $T \in \mathbb{N}$, *the binary mechanism yields an* ϵ*-differentially private algorithm to estimate* f *on a graph sequence, which has additive error* $O(\frac{\Gamma}{\varepsilon} \cdot \ln^{3/2}(T) \cdot \ln(1/\beta))$ *with probability at least* $1 - \beta$.

We replace the summation by the binary mechanism in Theorem B.2 by summation using M_{count}, getting the following result.

Corollary B.3. *Let* f *be a graph function whose difference sequence has* ℓ_2*-sensitivity* Γ. *There is an* (ϵ, δ)*-differentially private algorithm that, for any sequence of edge updates of length* $T > 0$ *to a graph* $\mathcal{G}$, *with probability at least 2/3 over the coin tosses of the algorithm, simultaneously for all rounds* t *with* $1 \leq t \leq T$, *outputs at time* t *an estimate of* $f(\mathcal{G}_t)$ *that has additive error at most* $C_{\epsilon,\delta} \Psi(t) \Gamma \sqrt{\ln(6T)}$.

B.3. Counting Substrings

We can also extend our mechanism for counting all substrings of length at most ℓ, where $\ell \geqslant 1$, in a sequence σ of letters. After each update i (i.e., a letter), we consider the binary vector $v_{\sigma,i}$ that is indexed by all substrings of length at most ℓ. The value of $v_{\sigma,i}[s]$, which corresponds to the substring s, indicates whether the suffix of length $|s|$ of the current sequence equals s. We can cast the problem of counting substrings as a binary sum problem on the sequence of vectors $v_{\sigma,\cdot}$ and apply $M_{\mathsf{ep}} = M_{\mathsf{count}} \otimes \mathbb{I}_u$ to the concatenated vectors, where $u = \sum_{i \leqslant \ell} |\mathcal{U}|^i$.

Corollary B.4. *Let $\mathcal{U}$ be a universe of letters, let $\ell \geqslant 1$. There exists an (ϵ, δ)-differentially private algorithm that, given a sequence of letters $s = s_1 \cdots s_T$ from $\mathcal{U}$, outputs, after each letter, the approximate numbers of substrings of length at most ℓ. With probability at least 2/3 over the coin tosses of the algorithm, simultaneously, for all rounds t with $1 \leq t \leq T$, the algorithm has at time t an additive error of at most $C_{\epsilon,\delta}\Psi(t)\ell\sqrt{\ln(2|U|^\ell)\ln(6T)}$, where $C_{\epsilon,\delta}$ is as defined in eq. (2).*

Proof. Let $\sigma = \sigma_1 \cdots \sigma_T$ and $\sigma' = \sigma'_1 \cdots \sigma'_T$ be two sequences of letters that differ in only one position p, i.e., $\sigma_i = \sigma'_i$ if and only if $i \neq p$. We observe that $v_{\sigma,i} = v_{\sigma',i}$ for any $i \notin \{p, \ldots, p+\ell-1\}$. Furthermore, for any i, $0 \leqslant i < \ell$ and j, $i+1 \leqslant j \leqslant \ell$, there exist only two substrings s of length j so that $v_{\sigma,p+i}[s] \neq v_{\sigma',p+i}[s]$. It follows that the ℓ_2-sensitivity is at most $\sqrt{\sum_{i=0}^{\ell-1}\sum_{j=i+1}^{\ell} 2} \leqslant \sqrt{\ell^2} = \ell$. Using $\sum_{i\leqslant\ell}|\mathcal{U}|^i \leqslant 2|U|^\ell$, the proof concludes analogously to the proof of Theorem B.1. □

B.4. Episodes

Given a universe of *events* (or *letters*) $\mathcal{U}$, an *episode* e of length ℓ is a word over the alphabet $\mathcal{U}$, i.e., $e = e_1 \cdots e_\ell$ so that for each i, $1 \leqslant i \leqslant \ell$, $e_i \in \mathcal{U}$. Given a string $s = s_1 \cdots s_n \in \mathcal{U}^*$, an occurence of e in s is a subsequence of s that equals e. A *minimal* occurrence of an epsiode e in s is a subsequence of s that equals e and whose corresponding substring of s does not contain another subsequence that equals e. In other words, $s_{i_1} \cdots s_{i_\ell}$ is a minimal occurence of e in s if and only if (1) for all j, $1 \leqslant j \leqslant \ell$, $s_{i_j} = e_j$ and (2) there does not exist $s_{j_1} \cdots s_{j_\ell}$ so that for all k, $1 \leqslant k \leqslant \ell$, $s_{j_k} = e_k$, and either $i_1 < j_1$ and $j_\ell \leqslant i_\ell$, or $i_1 \leqslant j_1$ and $j_\ell < i_\ell$. The support of an episode e on a string s is the number of characters from the string that are part of at least one minimal occurrence of e. Note that for an episode e, its minimal occurrences may overlap. For the non-differentially private setting, Lin *et al.* (Lin et al., 2014) provide an algorithm that dynamically maintains the number of minimal occurrences of episodes in a stream of events. For better performance, the counts may be restricted to those episodes that have some minimum support on the input (i.e., frequent episodes).

Lemma B.5 (Lin et al. (2014))**.** *Let $\mathcal{U}$ be a universe of events, let $\ell \geqslant 2$, and let $S \geq 1$. There exists a (non-private) algorithm that, given a sequence of events $s = s_1 \cdots s_T$ from $\mathcal{U}$, outputs, after each event, the number of minimal occurrences for each episode of length at most ℓ that has support at least S. The time complexity per update is $\tilde{O}(T/S + |\mathcal{U}|^2)$ and the space complexity of the algorithm is $\tilde{O}(|U| \cdot T/S + |\mathcal{U}|^2 \cdot T)$.*

There can be at most one minimal occurrence of e that ends at a fixed element $s_t \in s$. Therefore, we can view the output of the algorithm after event s_t as a binary vector $v_t \in \{0,1\}^{\sum_{i\leqslant\ell}|\mathcal{U}|^i}$ that is indexed by all episodes of length at most ℓ and that indicates, after each event s_t, if a minimal occurrences of epsiode e ends at s_t. Summing up the (binary) entries corresponding to e in $v_1, \ldots, v_t$ yields the number of minimal occurrences of e in $s_1 \cdots s_t$. Therefore, we can cast this problem of counting minimal occurrences of episodes as a binary sum problem and apply M_{ep}.

Corollary B.6. *Let $\mathcal{U}$ be a universe of events, let $\ell \geqslant 2$, and let $S \geq 1$. There exists an (ϵ, δ)-differentially private mechanism that, given a sequence of events $s = s_1 \cdots s_T$ from $\mathcal{U}$, outputs, after each event, the approximate number of minimal occurrences for each episode of length at most ℓ that has support at least S. With probability at least 2/3 over the coin tosses of the algorithm, simultaneously for all rounds t with $1 \leq t \leq T$ the algorithm has at time t an additive error of at most $C_{\epsilon,\delta}\Psi(t)\sqrt{|U|^{\ell-1}\ell\ln(2|U|^\ell)\ln(6T)}$.*

Proof. Let $\sigma = \sigma_1 \cdots \sigma_T$ and $\sigma' = \sigma'_1 \cdots \sigma'_T$ be two sequences of letters that differ in only one position p, i.e., $\sigma_i = \sigma'_i$ if and only if $i \neq p$. Recall that we are only interested in minimal occurences of episodes. Therefore, the number of query answers that are different for σ and σ' are trivially upper bounded by two times the maximum number of episodes that end on the same character (once for $\sigma[p]$ and once for $\sigma'[p]$), times the maximum length of an episode (as for every episode that ends at p, only the one with the latest start is a minimal occurrence). This is bounded by $2\sum_{i\leqslant\ell}|\mathcal{U}|^{i-1} \cdot \ell \leqslant 4|U|^{\ell-1}\ell$. It follows that the global ℓ_2-sensitivity is at most $2\sqrt{|U|^{\ell-1}\ell}$. Using $\sum_{i\leqslant\ell}|\mathcal{U}|^i \leqslant 2|U|^\ell$, the proof concludes analogously to the proof of Theorem B.1. □

On sparse streams. Until now, we have focused mainly on worst case analysis. We next consider the case when the stream is sparse, i.e., number of ones in the stream is upper bounded by a parameter s. Dwork et al. (Dwork et al., 2015) showed that under this assumption, one can asymtotically improve the error bound on continual release to $O\left(\frac{\log(T)+\log^2(s)}{\epsilon}\right)$ while preserving ϵ-differential privacy. Using their analysis combined with our bounds we directly get the error bound $C_{\epsilon,\delta}\left(5\log(T) + \Psi(n)\sqrt{\ln(6n)}\right)$.

B.5. Non-interactive Local Learning

In this section, we consider convex risk minimization in the non-interactive local differential privacy mode (LDP) using Theorem 1.1. That is, there are n participants (also known as *clients*) and one *server*. Every client has a private input d_i from a fixed universe $\mathcal{D}$. To retain the privacy of this input, each client applies a differentially-private mechanism to their data (*local* model) and then sends a single message to the server which allows the server to perform the desired computation (convex risk minimization in our case). After receiving all messages, the server outputs the result without further interaction with the clients (*non-interactive*).

In 1-dimensional convex risk minimization, a problem is specified by a convex, closed and bounded constraint set $\mathcal{C}$ in $\mathbb{R}$ and a function $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ which is convex in its first argument, that is, for all $D \in \mathcal{D}$, $\ell(\cdot; D)$ is convex. A data set $D = (d_1, \ldots, d_n) \in \mathcal{D}^n$ defines a loss (or *empirical risk*) function: $\mathcal{L}(\theta; D) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; d_i)$, where θ is a variable that is chosen as to minimize the loss function. The goal of the algorithm is to output a function f that assigns to each input D a value $\theta \in \mathcal{C}$ that minimizes the average loss over the data sample D. For example, finding the median of the 1-dimensional data set $D \in [0,1]^n$ consisting of n points in the interval $[0,1]$ corresponds to finding $\theta \in \mathcal{C}$ that minimizes the loss $\mathcal{L}(\theta, D) = \sum_i |\theta - d_i|$.

When the inputs are drawn i.i.d. from an underlying distribution $\mathcal{P}$ over the data universe $\mathcal{D}$, one can also seek to minimize the *population risk*: $\mathcal{L}_{\mathcal{P}}(\theta) = \mathsf{E}_{D\sim\mathcal{P}}[\ell(\theta; D)]$. We will use some notations in this section. Let $I_1, \cdots, I_w$ be w disjoint intervals of $[0,1]$ of size $s := \lfloor \frac{1}{\epsilon\sqrt{n}} \rfloor$. Let $\mathcal{B} = \{j \cdot s : 0 \leqslant j \leqslant w\}$. Given a vector $a \in \mathbb{R}^w$ let g be a "continuous intrapolation" of the vector a, namely $g : [0,1]^w \times [0,1] \to [0,1]$ such that $g(a, \theta) = a[k]$, where $k = \operatorname{argmin}_{z\in\mathcal{B}} |z - \theta|$, with ties broken for smaller values. Also, let $f : \mathbb{R}^w \times [0,1] \to [0,1]$ be defined as $f(a, x) = \int_0^x g(a, t)\mathrm{d}t$.

Smith et al. (2017) showed the following:

Theorem B.7 (Corollary 8 in Smith et al. (2017))**.** *For every 1-Lipschitz*[5] *loss function $\ell : [0,1] \times \mathcal{D} \to \mathbb{R}$, there is a randomized algorithm $Z : \mathcal{D} \to [0,1]$, such that for every distribution $\mathcal{P}$ on $\mathcal{D}$, the distribution $\mathcal{Q}$ on $[0,1]$ obtained by running Z on a single draw from $\mathcal{P}$ satisfies $\mathcal{L}_{\mathcal{P}}(\theta) = \mathsf{med}_{\mathcal{Q}}(\theta)$ for all $\theta \in [0,1]$, where $\mathsf{med}_{\mathcal{P}}(\theta) = \mathbb{E}_{d\sim\mathcal{Q}}[|\theta - d|]$.*

In other words, differentially private small error for 1-dimensional median is enough to solve differentially private loss minimization for general 1-Lipschitz functions. Prior work used a binary mechanism to determine the 1-dimensional median. We show how to replace this mechanism by the factorization mechanism of Theorem 1.2. As the reduction in Theorem B.7 preserves exactly the additive error, our analysis of the additive error in Theorem 1.2 carries through, giving a concrete upper bound on the additive error.

We first recall the algorithm of Smith et al. (2017). Median is non-differentiable at its minimizer θ^*, but in any open interval around θ^*, its gradient is either $+1$ or -1. STU first divides the interval $[0,1]$ into $w = \lceil \epsilon\sqrt{n} \rceil$ disjoint intervals $I_1, \cdots, I_w$ of $[0,1]$ of size $s := \lfloor \frac{1}{\epsilon\sqrt{n}} \rfloor$. Let $\mathcal{B} = \{j \cdot s : 0 \leqslant j \leqslant w\}$. Every client constructs a w-dimensional binary vector that has 1 only on the coordinate j if its data point $d_i \in I_j$. The client then executes the binary mechanism with the randomizer of Duchi *et al.* (Duchi et al., 2013) on its vector and sends the binary tree to the server. Based on this information the server computes a vector $x^{\mathsf{STU}} \in \mathbb{R}^w$, where $x^{\mathsf{STU}}[j]$ is the $1/n$ times the difference of the number of points in the interval $\cup_{l=1}^{j} I_l$ and the number of data points in the interval $\cup_{l=j+1}^{w} I_l$. The server outputs the function $f(x^{\mathsf{STU}}, \theta)$.

To replace the binary tree mechanism used in Smith et al. (2017) (dubbed as STU) into a factorization mechanism based algorithm is not straightforward because of two reasons: (i) Smith *et al.* used the binary mechanism with a randomization routine from Duchi et al. (2013), which expects as input a binary vector, while we apply randomization to Rx, where x is the binary vector, and (ii) the error analysis is based on the error analysis in Bassily & Smith (2015) which does not carry over to the factorization mechanism.

We now describe how we modify STU to give an LDP algorithm $\mathcal{A}$. Instead of forming a binary tree, every client i forms two binary vectors $u_i, v_i \in \{0,1\}^w$ with $u_i[j] = v_i[w - j] = 1$ if $d_i \in I_j$ and 0 otherwise. Note that in both vectors exactly 1 bit is set and that

$$\left(\sum_{i=1}^{n}\sum_{l=1}^{t} u_i[l]\right) - \left(\sum_{i=1}^{n}\sum_{l=t+1}^{w} v_i[l]\right)$$

[5] A function $\ell : \mathcal{C} \to \mathbb{R}$, defined over $\mathcal{C}$ endowed with ℓ_2 norm, is L-Lipschitz with respect if for all $\theta, \theta' \in \mathcal{C}$, $|\ell(\theta) - \ell(\theta')| \leqslant L\|\theta - \theta'\|_2$.

gives the number of bits in the interval $\cup_{l=1}^{t} I_l$ minus the number of data points in the interval $\cup_{l=t+1}^{w} I_l$. The user now sends two vectors $y_i, z_i \in \mathbb{R}^w$ to the server formed by running the binary counter mechanism defined in Theorem 1.2 on u_i and v_i with privacy parameters $(\frac{\epsilon}{2}, \frac{\delta}{2})$. Since the client's message is computed using a differentially private mechanism for each vector, the resulting distributed mechanism is (ϵ, δ)-LDP using the basic composition theorem.

On receiving these vectors, the server first computes the aggregate vector

$$\widehat{x}[t] = \frac{1}{n}\left(\sum_{i=1}^{n} y_i[t] - \sum_{i=1}^{n} z_i[w-t]\right). \tag{10}$$

The server then computes and outputs $f(\widehat{x}, \theta)$.

To analyze our mechanism let x^{STU} be the vector formed in STU and $\widetilde{x}$ be the vector that the server in STU would have formed if clients did not use any randomizer. Equation (3) in Smith et al. (2017) first showed that, for all

$$\forall \theta \in [0,1], \quad \left|g(x^{\mathsf{STU}}, \theta) - g(\widetilde{x}, \theta)\right| \leqslant \alpha \quad \text{for} \quad \alpha \in O\left(\frac{\log^2(\varepsilon^2 n)\sqrt{\log(\varepsilon^2 n)}}{\varepsilon\sqrt{n}}\right). \tag{11}$$

Smith et al. (2017) (see Theorem 6) then use the fact that $f(x,\theta) = \int_0^\theta g(x;s)\mathrm{d}s$ to show that $\left|f(x^{\mathsf{STU}}, \theta) - \mathsf{med}_{\mathcal{P}}(\theta)\right| \leqslant \left|g(x^{\mathsf{STU}}, \theta) - g(\widetilde{x}, \theta)\right| + \frac{2}{\epsilon\sqrt{n}}$ and use eq. (11) to get their final bound, which is $O\left(\frac{\log^2(\varepsilon^2 n)\sqrt{\log(\varepsilon^2 n)}}{\varepsilon\sqrt{n}}\right)$. We remark that we can replace x^{STU} by any $y \in \mathbb{R}^w$ as long as $|g(y,\theta) - g(\widetilde{x}, \theta)| \leqslant \alpha$ for all $\theta \in [0,1]$.

We now show an equivalent result to eq. (11). We argue that the vector $\widehat{x}$ serves the same purpose as x^{STU}. The key observation here is that $\sum_{i=1}^{n} y_i[t]$ contains the partial sum for the intervals $I_1, \ldots, I_t$ and $\sum_{i=1}^{n} z_i[w-t]$ contains the partial sum for $I_{j+1}, \ldots, I_w$. Let $\overline{x} = \frac{1}{n}(\sum_{i=1}^{n} u_i[t] - \sum_{i=1}^{n} v_i[w-t])$ be the vector corresponding to the estimates in eq. (10) if no privacy mechanism was used. Note that $\widetilde{x} = \overline{x}$. Since the randomness used by different clients is independent,

$$\mathsf{Var}[\widehat{x}[t]] = \frac{1}{n^2}\mathsf{Var}\left[\sum_{i=1}^{n}(y_i[t] - z_i[w-t])\right] = \frac{2}{n^2}\mathsf{Var}\left[\sum_{i=1}^{n} y_i[t]\right] = \frac{2}{n^2}\sum_{i=1}^{n}\mathsf{Var}\left[y_i[t]\right] = \frac{2}{n}\sigma_t,$$

where σ_t is the variance used in the binary counting mechanism of Theorem 1.2. Using the concentration bound as in the proof of Theorem 1.2, we have $\|\widehat{x} - \overline{x}\|_\infty \leqslant 2\beta$ with $\beta = C_{\frac{\epsilon}{2}, \frac{\delta}{2}}\sqrt{\frac{\ln(6(\epsilon\sqrt{n}+1))}{2n}}\left(1 + \frac{\ln(\epsilon\sqrt{n}+1)}{\pi}\right)$. By the definition of $g(\cdot,\cdot)$, we therefore have $\forall \theta \in [0,1]$, $|g(\widehat{x}, \theta) - g(\overline{x}, \theta)| \leqslant 2\beta$.

Now using the same line of argument as in Smith et al. (2017), we get the following bound:

Corollary B.8. *For every distribution $\mathcal{P}$ on $[0,1]$, with probability $2/3$ over $D \sim \mathcal{P}^n$ and $\mathcal{A}$, the output $\widehat{f} \leftarrow \mathcal{A}$ satisfies $|f(\widehat{x}, \theta) - \mathsf{med}_{\mathcal{P}}(\theta)| \leqslant 2\beta + \frac{2}{\epsilon\sqrt{n}}$, where $\mathsf{med}_{\mathcal{P}}(\theta) = \mathbb{E}_{d\sim\mathcal{Q}}[|\theta - d|]$. Further, $\mathcal{A}$ is (ϵ, δ)-LDP.*

Our algorithm $\mathcal{A}$ is non-interactive (ϵ, δ)-LDP algorithm and not ϵ-LDP as STU, but we can give $\mathcal{A}$ our algorithm as input to the $\mathsf{GenProt}$ transformation (Algorithm 3) in Bun et al. (2019) to turn it into a $(10\epsilon, 0)$-LDP algorithm (see Lemma 6.2 in Bun et al. (2019)) at the cost of increasing the population risk using Theorem 6.1 in Bun et al. (2019).

C. Lower Bounds

Definition C.1 (MAX-CUT)**.** Given a graph $\mathcal{G} = (V, E, w)$, the *maximum cut* of the graph is the optimization problem

$$\max_{S \subseteq V}\{\Phi_S(\mathcal{G})\} = \max_{S \subseteq V}\left\{\sum_{u \in S, v \in V \setminus S} w(u,v)\right\}.$$

Let $\mathsf{OPT}_{\mathsf{max}}(\mathcal{G})$ denote the maximum value.

In this section we use a reduction from the maximum sum problem. Let $\mathcal{X} = \{0,1\}^d$, let $x \in \mathcal{X}^T$, $d \in \mathbb{N}$, and for $1 \leqslant j \leqslant d$, let $x_t[j]$ denote the j-th coordinate of record x_t. A mechanism for the *d-dimensional maximum sum problem under continual observation* is to return for each $0 \leqslant t \leqslant T$, the value $\max_{1 \leqslant j \leqslant d} \sum_{s=1}^{t} x_s[j]$.

In (Jain et al., 2021) Jain et al. studied the problem of computing in the continual release model the maximum sum of a d-dimensional vector. Two vectors x and x' are neighboring if they differ in only one d-dimensional vectors x_t for some $1 \leqslant t \leqslant T$. They showed that for any (ϵ, δ)-differentially private and (α, T)-accurate mechanism for maximum sum problem under continual observation it holds that

1. $\alpha = \Omega\left(\min\{\frac{T^{1/3}}{\epsilon^{2/3}\log^{2/3}(\epsilon T)}, \frac{\sqrt{d}}{\epsilon \log d}, T\}\right)$ if $\delta > 0$ and $\delta = o(\epsilon/T)$;
2. $\alpha = \Omega\left(\min\{\sqrt{T/\epsilon}, d/\epsilon, T\}\right)$ if $\delta = 0$.

We use this fact to show a lower bound for maintaining a minimum cut under continual observation, where each update consists of a set of edges that are inserted or deleted.

Theorem C.2. *For all $\epsilon \in (0,1), \delta \in [0,1)$, sufficiently large $T \in \mathbb{N}$ and any mechanism $\mathcal{M}$ that returns the value of the minimum cut in a multi-graph with at least 3 nodes in the continual release model, is (ϵ, δ)-differentially private, and (α, T)-accurate it holds that*

1. $\alpha = \Omega\left(\min\{\frac{T^{1/3}}{\epsilon^{2/3}\log^{2/3}(\epsilon T)}, \frac{\sqrt{n}}{\epsilon \log n}, T\}\right)$ *if $\delta > 0$ and $\delta = o(\epsilon/T)$;*
2. $\alpha = \Omega\left(\min\{\sqrt{\frac{T}{\epsilon}}, \frac{n}{\epsilon}, T\}\right)$ *if $\delta = 0$.*

The same hold for any mechanism maintaining the minimum degree.

Proof. Using a mechanism $\mathcal{M}$ for the minimum cut problem under continual observation for a graph $\mathcal{G} = (V, E)$ with $d+1$ nodes we show how to solve the d-dimensional maximum sum problem under continual observation. During this reduction the input sequence of length T for the maximum sum problem is transformed into a input sequence of length T for the minimum cut problem. The lower bound then follows from this and the fact that $n = d+1$ in our reduction.

Let $\mathcal{G}$ be a clique with $d+1$ nodes such that one of the nodes is labeled v and all other nodes are numbered consecutively by $1, \ldots, d$. For every pair of nodes that does not contain v, give it T parallel edges, and give every node j with $1 \leqslant j \leqslant d$ $3T$ parallel edges to v. Note that v has initially degree $3Td$, every other node has initially degree $T(d+2)$ and the minimum degree corresponds to the minimum cut. Whenever a new vector x_t arrives, give to $\mathcal{M}$ a sequence update that removes one of the parallel edges (v, j) for every j with $x_t[j] = 1$. Let j^* be the index that maximizes $\sum_{s=1}^{t} x_s[j]$. Note that the corresponding node labeled j^* has degree $T(d+2) - \sum_{s=1}^{t} x_s[j^*]$, while v has degree at least $2Td \geq T(d+2)$ as $d+1 \geq 3$, and every other node has degree at least $T(d+2) - \sum_{s=1}^{t} x_s[j^*]$. Furthermore the minimum degree also gives the minimum cut in $\mathcal{G}$. Thus $\mathcal{M}$ can be used to solve the maximum sum problem and the lower bound follows from the above.

Note that the proof also shows the result for a mechanism maintaining the minimum degree. □

It follows that for $T \geqslant n^{3/2}/\log n$ the additive error for any (ϵ, δ)-differentially private mechanism is $\Omega(\sqrt{n}/(\epsilon \log n))$, which implies that our additive error is tight up to a factor of $\log n \log^{3/2} T$ if the minimum cut S has constant size.

We now show a lower bound for counting substrings:

Theorem C.3. *For all $\epsilon \in (0,1), \delta \in [0,1)$, sufficiently large $T \in \mathbb{N}$, universe $\mathcal{U}$, $\ell \geq 1$ and $S \geq 1$ and for any mechanism $\mathcal{M}$ that, given a sequence s of letters from U, outputs, after each letter the approximate number of substrings of length at most ℓ that has support at least S, is (ϵ, δ)-differentially private, and (α, T)-accurate it holds that*

1. $\alpha = \Omega\left(\min\{\frac{T^{1/3}}{\epsilon^{2/3}\log^{2/3}(\epsilon T)}, \frac{\sqrt{\log|U|}}{\epsilon \log\log|U|}, T\}\right)$ *if $\delta > 0$ and $\delta = o(\epsilon/T)$;*

2. $\alpha = \Omega\left(\min\{\sqrt{T/\epsilon}, \log|U|/\epsilon, T\}\right)$ *if* $\delta = 0$.

Proof. Using a mechanism for substring counting under continual observation up to length $\ell = 1$ and universe $\mathcal{U}$ of letters of size 2^d we show how to create a mechanism $\mathcal{M}$ for the d-dimensional maximum sum problem under continual observation. During this reduction the input sequence of length T for the maximum sum problem is transformed into a sequence of length T. The lower bound follows from this and the fact that $d = \log|U|$.

Let $\mathcal{U}$ consist of 2^d many letters s_p for $1 \leqslant p \leqslant 2^d$, one per possible record in $\mathcal{X} = \{0,1\}^d$. Given a d-dimensional bit-vector x_t at time step t we append to the input string s the corresponding letter $|U|$. Thus, two neighboring inputs $x, x' \in \mathcal{X}^T$ for the maximum sum problem lead to two neighboring sequences s and s' for the substring counting problem. The substring counting mechanism outputs at time step t an approximate count of all substrings of length 1 with maximum error α over all counts and all time steps. Our mechanism $\mathcal{M}$ determines the maximum count returned for any substring of length 1 and returns it. This gives an answer to the maximum sum problem with additive error at most α. □

This implies that for large enough T and constant ℓ the additive error of our mechanism is tight up to a factor of $\log\log|U|\log^{3/2} T$.

Proof of Theorem 1.4. Define the function, $f(t) := \frac{2}{\pi}\log\left(\cot\left(\frac{\pi}{4t}\right)\right)$. It is easy to see that $f(t) = \frac{1}{t}\int_1^t \left|\csc\left(\frac{(2x-1)\pi}{2t}\right)\right| \mathrm{d}x$. From the basic approximation rule of Reimann integration, this implies that $\widehat{\gamma}_t \leqslant f(t)$. The following limit follows using L'Hospital rule:

$$\lim_{t\to\infty} \frac{2}{\pi}\frac{\ln\left(\cot\left(\frac{\pi}{4t}\right)\right)}{\ln(t)} = \lim_{t\to\infty} \frac{\csc^2\left(\frac{\pi}{4t}\right)}{2t\cot\left(\frac{\pi}{4t}\right)} = \frac{2}{\pi}.$$

That is, we have the following:

Lemma C.4. $\lim_{t\to\infty} \frac{1}{t}\sum_{j=1}^t \left|\{\csc\left(\frac{(2j-1)\pi}{2t}\right)\right| \to \frac{2}{\pi}$ *from above.*

Let us consider the case when we are using an additive, data-independent mechanism for non-adaptive continual observation. That is, $\mathfrak{M} = \{\mathcal{M} : \mathcal{M}(x) = M_{\mathsf{count}}x + z\}$, where z is a random variable over $\mathbb{R}^T$ whose distribution does not depend on x. The proof follows similarly as in the mean-squared case in Edmonds et al. (Edmonds et al., 2020). Note that,

$$\max_{x\in\{0,1\}^T} \mathbb{E}\left[\|\mathcal{M}(x) - M_{\mathsf{count}}x\|_\infty^2\right] = \mathbb{E}[\|z\|_\infty^2], \tag{12}$$

where the expectation is over the coin tosses of $\mathcal{M}$.

Let $\Sigma = \mathbb{E}[zz^\top]$ be the covariance matrix of z so that $\mathbb{E}[\|z\|_\infty^2] \geq \max_{1\leqslant i\leqslant T}\Sigma[i,i]$. Now let us define $K = M_{\mathsf{count}}B_1^T$ to be the so called *sensitivity polytope* and B_1 to be the T-dimensional ℓ_1 unit ball. As M_{count} has full rank, it follows that K is full dimensional. Now using Lemma 27 in Edmonds et al. (2020), we have that there exists an absolute constant C such that

$$\max_{y\in K}\left\|\Sigma^{-1/2}y\right\|_2 \leqslant C\epsilon.$$

Define $L = \Sigma^{1/2}$ and $R = \Sigma^{-1/2}M_{\mathsf{count}}$. Then

$$\|R\|_{1\to 2} = \max_{1\leqslant i\leqslant T}\left\|\Sigma^{-1/2}M_{\mathsf{count}}[:i]\right\|_2 \leqslant \max_{y\in K}\left\|\Sigma^{-1/2}y\right\|_2.$$

That is, $\|R\|_{1\to 2} \leqslant C\epsilon$. Further,

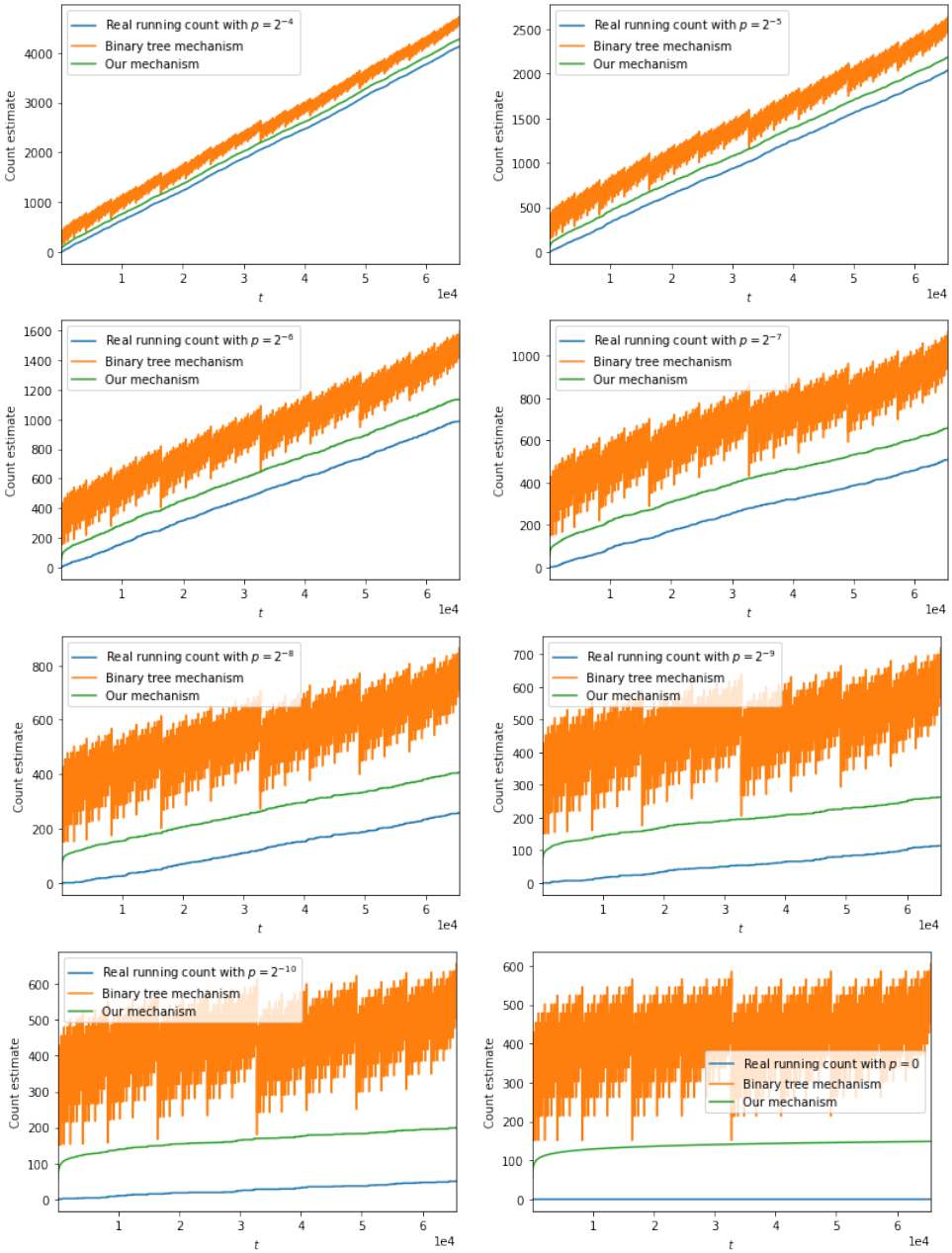
$$\|L\|_{2\to\infty}^2 = \max_{1\leqslant i\leqslant T}(L^\top L)[i,i] = \max_{1\leqslant i\leqslant T}\Sigma[i,i] \leq \mathbb{E}[\|z\|_\infty^2].$$

By the definition of $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$, we thus have

$$\|M_{\mathsf{count}}\|_{\mathsf{cb}}^2 \leqslant \|L\|_{2\to\infty}^2\|R\|_{1\to 2}^2 \leqslant C^2\epsilon^2\mathbb{E}[\|z\|_\infty^2].$$

Using the lower bound on $\|M_{\mathsf{count}}\|_{\mathsf{cb}}$, rearranging the last inequality, plugging them into eq. (12), and using Lemma 29 in Edmonds et al. (2020) completes the proof of Theorem 1.4. □

Figure 3: Comparison of our mechanism with binary mechanism for $T = 2^{16}, \epsilon = 0.5, \delta = 10^{-10}$ and various sparsity level. The x-axis is the current time epoch, and the y-axis gives the output of the algorithms at each time epoch.



D. Experiments

We empirically evaluated algorithms for two problems, namely continual counting and continual top-1 statistic, i.e. the frequency of the most-frequent element in the histogram. As the focus of this paper is on specifying the exact constants beyond the asymptotic error bound on differentially private counting, we do not make any assumption on the data or perform no post-processing on the output. The main goal of our experiments is to compare the additive error of (1) our mechanism and (2) the binary mechanism instantiated with Gaussian noise (i.e., the binary mechanism that achieves (ϵ, δ)-differential privacy). We do not compare our mechanism with the binary mechanism with Laplacian noise as it achieves a stronger

notion of differential privacy, namely ϵ-differential privacy, and has an asymptotically worse additive error.

We also implemented Honaker's variant of the binary mechanism (Honaker, 2015), but it was so slow that the largest T value for which it terminated before reaching a 5 hour time limit was 512. For these small values of T its ℓ_∞-error was worse than the ℓ_∞-error of the binary mechanism with Gaussian noise, which is not surprising as Honaker's variant is optimized to minimize the ℓ_2-error, not the ℓ_∞-error. Thus, we omit this algorithm in our discussion below.

Data sets for continual counting. For 8 different values of p, namely for every $p \in \{2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}, 2^{-9}, 2^{-10}, 0\}$, we generated a stream of $T = 2^{16}$ Bernoulli random variables $\mathsf{Ber}(p)$. Note that the eighth stream is an all-zero stream. Using Bernoulli random variables with $p \neq 0$ ensures that our data streams do not satisfy any smoothness properties, i.e., it makes it challenging for the mechanism to output smooth results.

We conjectured that using different p values would not affect the *magnitude of the additive ℓ_∞-error*, as the noise of both algorithms is independent of the data. Indeed our experiments confirmed this conjecture, i.e., the additive error in the output is not influenced by the different input streams. Note that the same argument also applies to real-world data, i.e., we would get the same results with real-world data. This was observed before and exploited in the empirical evaluation of differentially private algorithms for industrial deployment. For example, Apple used an all-zero stream to test its first differentially private algorithm, see the discussion on the accuracy analysis in the talk of Thakurta at Usenix 2017 (Thakurta, 2017). An all-zero stream was also used in the empirical evaluation of the differentially private continual binary counting mechanism in Denisov et al. (2022) (see Figure 1 in the cited paper).

We evaluated data streams with varying values of p to not only study the additive error, but also the *signal to noise ratio (SNR)* in data streams with different sparsity of ones.

Data sets for top-1 statistic. We generated a stream of 2048 elements from a universe of 20 items using Zipf's law (Zipf, 2016). Zipf's Law is a statistical distribution that models many data sources that occur in nature, for example, linguistic corpi, in which the frequencies of certain words are inversely proportional to their ranks. This is one of the standard distributions used in estimating the error of differentially private histogram estimation (Cardoso & Rogers, 2021).

Experimental setup. To ensure that the confidence in our estimates is as high as possible and reduce the fluctuation due to the stochasticity of the Gaussian samples, we ran both the binary tree mechanism and our matrix mechanism for 10^6 repetitions and took the average of the outputs of these executions.

Results. Figure 3 shows the output of the algorithms for continual counting. Figure 4 shows the private estimate of the frequency of the most frequent item for each algorithm. On the y-axis, we report the output of the algorithms and the non-private output, i.e., the real count. The x-axis is the current time epoch.

(1) The first main takeaway is that our additive error (i.e., difference between our estimate and real count) is consistently less than that of the binary mechanism. For $t = 2^i - 1$ for $i \in \mathbb{N}$ the improvement in the additive error is factor of roughly 4. This aligns with our Theorem 1.3. We note that a similar observation was made in the recent work (Henzinger et al., 2023) with respect to the ℓ_2-error.

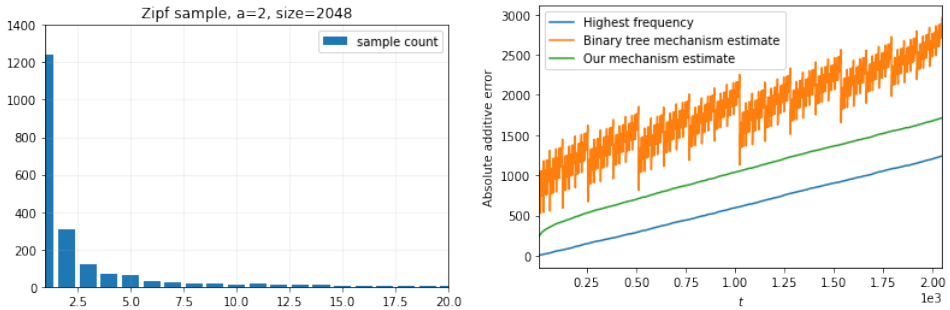
(2) The second main takeway of our experiments is that the error due to the binary mechanism is a non-smooth and a non-monotonic function of t while our mechanism distributes the error smoothly and it is monotonically increasing. This can be explained from the fact that the variance of the noise in our algorithm is a monotonic function, $\ln^2(t)$, while that of the binary mechanism is $\ln(b)$, where b is the number of ones in the bitwise representation of t, i.e., a non-smooth, non-monotonic function.

p	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
Binary	4.72	2.40	1.12	0.61	0.31	0.15	0.072
Our mech	14.50	7.37	3.46	1.86	0.96	0.47	0.22

Table 2: Average signal to noise ratio between private estimates and true count for various sparsity level.

(3) In Table 2, we present the average SNR over all time epochs between the private estimates and the true count for different sparsity values of the stream. We notice that our output is consistently better and is about three times better when $p = 2^{-10}$. We noticed that for $\epsilon = 0.5, \delta = 10^{-10}$ when the fraction of ones is about $1/80$, the average SNR for the binary mechanism drops below 1, i.e., the error is larger than the true count, while for our mechanism it only drops below 1 when the fraction

Figure 4: (Left) We give the Zipf's law distribution that items are sampled from at each round of an event stream. (Right) the running estimate of the most frequent item using the binary mechanism and our mechanism instantiated with $\epsilon = 0.1, \delta = 10^{-10}$.



of ones is $\leqslant 1/250$. That is, our mechanism can handle three times sparser streams than the binary mechanism for the same SNR. This observation continued to hold for different privacy parameters.

(4) For histogram estimation, our experiments reveal that our mechanism performs consistently better than the binary mechanism, both in terms of the absolute value of the additive error incurred as well as in terms of the smoothness of the error. This is consistent with our theoretical results. Further, on average over all time epochs, the SNR for our mechanism is 1.47 while that of binary mechanism is 0.52, i.e., it is a factor of about 3 better.