Improved Accuracy for Private Continual Cardinality Estimation in Fully Dynamic Streams via Matrix Factorization

JOEL DANIEL ANDERSSON, Institute of Science and Technology Austria, Austria
PALAK JAIN, Boston University, USA
SATCHIT SIVAKUMAR, Boston University, USA

We study differentially-private statistics in the *fully dynamic continual observation* model, where many updates can arrive at each time step and updates to a stream can involve both insertions and deletions of an item. Earlier work (e.g., Jain et al., NeurIPS 2023 for counting distinct elements; Raskhodnikova & Steiner, PODS 2025 for triangle counting with edge updates) reduced the respective cardinality estimation problem to continual counting on the *difference stream* associated with the true function values on the input stream. In such reductions, a change in the original stream can induce many changes in the difference stream. This poses a challenge for applying private continual counting algorithms to obtain optimal error bounds. We improve the accuracy of several such reductions by studying the associated ℓ_p-sensitivity vectors of the resulting difference streams and isolating their properties.

We demonstrate that our framework gives improved bounds for counting distinct elements, estimating degree histograms, and estimating triangle counts (under a slightly relaxed privacy model), thus offering a general approach to private continual cardinality estimation in streaming settings. Our improved accuracy stems from tight analysis of known factorization mechanisms for the counting matrix in this setting; the key technical challenge is arguing that one can use state-of-the-art factorizations for sensitivity vector sets with the properties we isolate. In particular, we show that for approximate DP, the celebrated square-root factorization of the counting matrix (Henzinger et al., SODA 2023; Fichtenberger et al., ICML 2023) can be employed to give concrete improvements in accuracy over past work based on the binary tree mechanism. We also give a tight analysis of b-ary tree mechanisms with subtraction under these sensitivity patterns, including a polytime routine for computing the ℓ_p sensitivity, yielding time and space efficient algorithms for both pure and approximate DP. Empirically and analytically, we demonstrate that our improved error bounds offer a substantial improvement in accuracy for cardinality estimation problems over a large range of parameters.[1]

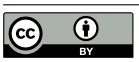
CCS Concepts: • **Security and privacy** → *Information-theoretic techniques*; • **Theory of computation** → **Online algorithms**; **Theory of database privacy and security**.

Additional Key Words and Phrases: differential privacy, continual observation, matrix factorization, counting

ACM Reference Format:
Joel Daniel Andersson, Palak Jain, and Satchit Sivakumar. 2026. Improved Accuracy for Private Continual Cardinality Estimation in Fully Dynamic Streams via Matrix Factorization. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 106 (May 2026), 26 pages. https://doi.org/10.1145/3801902

[1]Formal statements and proofs omitted from this version of the paper can be found in the full version.

Authors' Contact Information: Joel Daniel Andersson, joel.andersson@ist.ac.at, Institute of Science and Technology Austria, Klosterneuburg, Austria; Palak Jain, the.palakj@gmail.com, Boston University, Boston, USA; Satchit Sivakumar, satchit@bu.edu, Boston University, Boston, USA.



ACM 2836-6573/2026/5-ART106
https://doi.org/10.1145/3801902

1 Introduction

Privacy is a central challenge for systems that process sensitive data, especially in settings where data evolves continuously and must be monitored or used over time. The continual observation model[2] of differential privacy [11, 19] provides rigorous guarantees for individual privacy in exactly such settings. In this model, algorithms process data streams that evolve over time, aiming to output accurate statistics at each time step while preserving differential privacy across all outputs. The central challenge in this setting is that privacy loss accumulates over time, and naive composition of privacy guarantees at each time step leads to an error blowup that is polynomial in the time-horizon T, compared to estimating the statistic only once for the entire stream. In this work, we revisit the problem of differentially private cardinality estimation under continual observation in the challenging *fully dynamic* setting where updates can involve both insertions and deletions.

Our main contribution is to show that new analyses of Toeplitz factorization mechanisms for continual counting[3] can yield concrete accuracy improvements for several cardinality estimation problems in the continual observation model of differential privacy. To enable this, we (1) define and analyze a new structured class of sensitivity vectors that arise in reductions from cardinality estimation problems to continual counting. The problems we consider include counting distinct elements in a stream (CountDistinct), tracking the degree histogram of a dynamically evolving graph (DegreeCount), and counting the number of triangles in such a graph (TriangleCount). (2) prove error bounds for factorization mechanisms based on Toeplitz factorizations for continual counting with respect to this new class of sensitivity vectors; and (3) show that even tree-aggregation-based factorization mechanisms admit improved error guarantees when adapted to sensitivity vectors with the structure we identify.

Improved Mechanisms for Continual Counting. The continual release of bit sums, also called 'continual counting', is a fundamental building block in various data analysis tasks. The binary tree mechanism [19, 11] for this problem has asymptotic error that is within logarithmic factors of optimal in the stream length T [19, 35]. However, while the binary tree mechanism is frequently used to design continual release mechanisms for other tasks [39, 38, 50, 41, 40], the constants involved in its error guarantee are suboptimal [25, 35], which can have a big impact in practical applications [41, 17].

Therefore, a long line of work has focused on optimizing the concrete accuracy of differentially private algorithms for continual counting. All of these approaches, whether presented explicitly in this way or not, can be expressed within the general framework of factorization mechanisms [45].[4] In this framework, the target workload for the problem of continual counting (prefix sum queries over a time horizon T) is represented by the lower triangular all-ones matrix $A \in \mathbb{R}^{T \times T}$. Hence, the prefix sums of stream $\vec{\mathbf{x}}$ correspond to the matrix-vector product $A\vec{\mathbf{x}}$. Factorization mechanisms consider factorizations of the form $A = LR$, and compute the intermediate quantities $R\vec{\mathbf{x}} + \vec{\mathbf{z}}$ where $\vec{\mathbf{z}}$ is an i.i.d. Gaussian noise vector with variance scaled proportionally to the *sensitivity* $\mathrm{sens}_2(R) = \max_{\vec{\mathbf{y}} \sim \vec{\mathbf{y}}'} \|R(\vec{\mathbf{y}} - \vec{\mathbf{y}}')\|_2$, where $\sim$ represents the neighboring relation. These can then be post-processed using the linear transformation L to get answers to the original queries.

This line of work has two flavors: The first considers hierarchical methods that correspond to sparse factorizations of A. That is, approaches similar to the binary tree mechanism, typically involving a combination of more clever aggregation functions and modifications to the tree structures [28, 49, 36, 2, 4]. The second set of works considers dense factorizations $A = LR$ [17, 25, 35, 18, 3, 33, 30], where R lacks a simple combinatorial structure, but instead is structured in some other

[2] Also called the continual release model.

[3] i.e., summation of bits.

[4] Sometimes also referred to as matrix mechanisms.

way (for example, R could be a Toeplitz matrix). These methods can achieve near-optimal concrete accuracy for continual counting. However, we cannot directly apply these improvements in the context of cardinality estimation, as discussed below.

Inability to Directly Use Improved Continual Counting Mechanisms for Cardinality Estimation. In cardinality estimation problems such as counting distinct elements or releasing the degree list of a graph [5], a frequent strategy is to construct the *difference stream* (for a function f and stream $\vec{\mathbf{x}}$, the difference stream is $d_f(\vec{\mathbf{x}})[t] = f(\vec{\mathbf{x}})[t] - f(\vec{\mathbf{x}})[t-1]$) and to run the continual counting algorithms on this stream [51, 24, 38, 50].

One work employing this strategy is that of Jain et al. [38], which uses this strategy for counting distinct elements. Unfortunately, for this task, a change to the input stream at one time step $\vec{\mathbf{x}}[t]$ could result in as many as T changes to the difference stream (in fully dynamic streams where elements can be both inserted and deleted). Thus, a naive ε-DP mechanism, one that uses an ε/T-DP mechanism for continual counting to release the prefix sums of this difference stream (giving an ε-DP guarantee via group privacy), would result in a factor T blowup in error compared to continual counting.[6] A similar phenomenon occurs for other cardinality estimation problems, including the estimation of triangle counts and degree lists for graph streams.

To get around this, Jain et al. [38] suggest parametrized accuracy guarantees, specifically for CountDistinct. They define the notion of *maximum flippancy* of a stream, and show that maximum flippancy bounded by k implies that a change to the input stream at one time step $\vec{\mathbf{x}}[t]$ could result in at most k changes to the difference stream.[7] In this case, since neighboring inputs induce changes in at most k coordinates of the difference stream, using an ε/k-DP continual counting mechanism yields an ε-DP algorithm with a factor of k error blowup. [8] Jain et al. also improve the dependence on maximum flippancy to a factor of $\sqrt{k}$ via a non-black-box use of the binary tree mechanism.

Unfortunately, their analysis is specific to the binary tree mechanism (which suffers from unfavorable constants in error bounds), and as raised in their paper, it is unclear if mechanisms that involve smaller noise addition (such as sophisticated Toeplitz factorization mechanisms) can be substituted in its place. Their analysis is also tailored to CountDistinct, and does not provide a roadmap to obtaining similar accuracy guarantees for other cardinality estimation problems like DegreeCount. This motivates the main question of this paper.

Can we improve the accuracy of differentially private cardinality estimation under continual observation by replacing the binary tree mechanism with other continual counting mechanisms?

We answer this question affirmatively, giving significant improvements in accuracy for several cardinality estimation problems.

We note that mechanisms for cardinality estimation problems are used as a building block in more complicated tasks, and so achieving better error bounds for them is critical for encouraging more practical deployment of differential privacy. For example, CountDistinct is used to detect abnormalities in networks [1], compute genetic differences between species [6] etc., TriangleCount is used to detect and monitor cohesiveness of communities in a social network [20], detect presence of spamming activity in Web graphs [7] etc., DegreeCount is used to track information about the spread of STDs [12], compute parameters of online social networks [16] etc. Indeed, for

[5] See Section 2.1 for formal definitions of the specific problems we consider.

[6] Note that such an algorithm would have worse error than the simple baseline of adding Gaussian noise to the function value at each time step and paying for composition over T time steps.

[7] For further discussion of parametrized guarantees, see Section 3.

[8] Note that this would give privacy and accuracy for streams with maximum flippancy bounded by k; extending to privacy for all streams can be done via a pre-processing step.

standard continual counting, the constant factor improvements in accuracy via novel factorization mechanisms [41, 17, 25, 35, 18, 47] have encouraged their adoption in private machine learning systems [52, 27].

2 Short Preliminaries

We introduce a minimal set of preliminaries serving as necessary context to the paper.

Notation. Let $0 \le i \le j < T$ and $\vec{\mathbf{x}} \in \mathbb{R}^T$. Vectors are zero-indexed, with $\vec{\mathbf{x}}[i : j] \in \mathbb{R}^{j-i+1}$ representing the vector with entries $\vec{\mathbf{x}}[i], \ldots, \vec{\mathbf{x}}[j]$. We let $[i, j] = \{i, i+1, \ldots, j\}$, and use $[T] = [0, T-1]$. For $b > 0$, $\log_b(\cdot)$ denotes the logarithm in base b, with $\log(\cdot) \equiv \log_2(\cdot)$ and $\ln(\cdot) \equiv \log_e(\cdot)$.

Differential Privacy and Continual Observation. Most of our results are stated for *ρ-zero-concentrated differential privacy.*

Definition 2.1 (ρ-Zero-Concentrated Differential Privacy [9]). *A randomized algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{Y}$ satisfies ρ-zCDP if for all neighboring inputs $x, y \in \mathcal{D}$, denoted $x \sim y$, we have that*

$$\forall \alpha > 1 : D_\alpha(\mathcal{M}(x) \| \mathcal{M}(y)) \le \rho\alpha\,,$$

where $D_\alpha(\mathcal{M}(x) \| \mathcal{M}(y))$ is the α-Rényi divergence between $\mathcal{M}(x)$ and $\mathcal{M}(y)$.

We refer to "$\sim$" as the *neighboring relation* for a given problem.

Our focus is on designing differentially private *online* algorithms. Let f be a function taking as input a streamed sequence $\vec{\mathbf{x}}$ of length T, and outputting $f(\vec{\mathbf{x}}) \in \mathbb{R}^T$ where $f(\vec{\mathbf{x}})[t]$ only depends on $\vec{\mathbf{x}}[0 : t]$. Consider a neighboring relation $\sim$ on inputs to f. We say that $\mathcal{M}$ *privately releases f under continual observation* (with respect to a fixed neighboring relation $\sim$), if

(1) For any valid input stream $\vec{\mathbf{x}}$, $\mathcal{M}(\vec{\mathbf{x}}) \approx f(\vec{\mathbf{x}})$, and $\mathcal{M}(\vec{\mathbf{x}})[t]$ is produced on receiving $\vec{\mathbf{x}}[t]$.
(2) The full transcript $\mathcal{M}(\vec{\mathbf{x}})$ satisfies differential privacy with neighboring relation $\sim$.

Error Notions. Our results focus on two notions of error for real-valued functions f.

$$\mathsf{MeanSE}(\mathcal{M}, \vec{\mathbf{x}}) = \sqrt{\mathbb{E}\left[\frac{1}{T}\sum_{t\in[T]}(f(\vec{\mathbf{x}})[t] - \mathcal{M}(\vec{\mathbf{x}})[t])^2\right]} \qquad \textit{(root mean squared error)}$$

$$\mathsf{MaxSE}(\mathcal{M}, \vec{\mathbf{x}}) = \sqrt{\mathbb{E}\left[\max_{t\in[T]}(f(\vec{\mathbf{x}})[t] - \mathcal{M}(\vec{\mathbf{x}})[t])^2\right]} \qquad \textit{(root maximum squared error)}$$

The definition here is for one-dimensional functions f. If f is vector-valued (for example, DegreeCount in the next section), root mean squared error and root max squared error are defined as the maximum over coordinates of the root (mean or maximum) squared error for the output of the mechanism restricted to each coordinate. Whenever $\mathcal{M}$ is an unbiased estimator of f, these errors are independent of $\vec{\mathbf{x}}$, in which case we drop the second argument and write e.g., $\mathsf{MaxSE}(\mathcal{M})$ for short.

2.1 Three Fully Dynamic Cardinality Estimation Problems

We begin by introducing notation shared across the problems. An input stream $\vec{\mathbf{x}}$ corresponds to a sequence of T updates (insertions and deletions, possibly multiple per time step) made to items from a finite universe $\mathcal{U}$. Formally, for all $t \in [T]$, we consider updates $\vec{\mathbf{x}}[t] \in \mathcal{U}_\pm$ where $\mathcal{U}_\pm := (\{+, -\} \times \mathcal{U}) \cup \{\perp\})^*$. Here $(+, u)$ and $(-, u)$ denotes adding or deleting an item $u \in \mathcal{U}$, and $\perp$ represents a "no op". We will use $\vec{\mathbf{x}}[t][i]$ to represent the i^{th} update at time step t, and informally write $+u$, $-u$ as shorthands for adding or deleting u. We note that some papers (including those studying the problems we consider [38, 15, 50]) restrict the setting further to allow only one update per timestep. Motivated by practical settings, we allow for multiple items to arrive per time step. We discuss some differences between these settings in Appendix B.

The neighboring relation we use for all of our problems will correspond to item-level privacy.[9] Formally, we say that two streams $\vec{\mathbf{x}}$ and $\vec{\mathbf{x}}' \in \mathcal{U}_{\pm}^T$ are neighboring if there exists at most one element $u \in \mathcal{U}$ such that replacing all updates corresponding to u (i.e. $-u$ or $+u$) with $\perp$ in one of the streams results in the other. Next, we define our problems of interest.

Counting Distinct Elements under Continual Observation. CountDistinct takes as input a stream $\vec{\mathbf{x}}$ (a sequence of T updates) and outputs a stream of T non-negative integers corresponding to the number of distinct elements at each time step. For $t \in [T]$,

$$\mathsf{CountDistinct}(\vec{\mathbf{x}})[t] = \sum_{u \in \mathcal{U}} \mathbb{1}(u \text{ has been inserted more often than deleted in } \vec{\mathbf{x}}[0:t]) \,.$$

Graph Functions under Continual Observation. For graph problems, we consider a fixed vertex set $[n]$, and consider updates on the universe $\mathcal{U} = [n] \times [n]$, corresponding to insertions and deletions of possible edges. Hence, a stream $\vec{\mathbf{x}} \in \mathcal{U}_{\pm}^T$ in this setting can be thought of as inducing a sequence of graphs $G_0, \ldots, G_{T-1}$ all on the same n nodes, but with evolving edge sets. At time step t, an edge belongs to G_t if and only if its cumulative number of insertions up to and including time step t exceeds its cumulative number of deletions. We will abuse notation to interchangeably refer to the stream as a sequence of edge updates, or as this sequence of graphs. We note that our privacy notion then corresponds to edge privacy.

In this paper, we study the *degree histogram* and *triangle count* problems. DegreeCount is a function that takes as input a stream $\vec{\mathbf{x}}$ (a sequence of T graphs $G_0, \ldots, G_{T-1}$) and outputs a stream of T non-negative integer vectors each of size n, corresponding to the vertex degrees at each time step. TriangleCount is a function that takes as input a stream $\vec{\mathbf{x}}$ (a sequence of T graphs $G_0, \ldots, G_{T-1}$) and outputs a stream of T non-negative integers, corresponding to the number of triangles at each time step. A triangle in a graph corresponds to the presence of a set of edges $\{(a,b),(b,c),(a,c)\}$ for three nodes a, b, c in the graph. For $t \in [T]$, $i \in [n]$:

$$\mathsf{DegreeCount}(\vec{\mathbf{x}})[t][i] = \sum_{a \in [n]} \mathbb{1}[\text{edge } (i,a) \in G_t]$$

$$\mathsf{TriangleCount}(\vec{\mathbf{x}})[t] = \sum_{a,b,c \in [n]} \mathbb{1}[a,b,c \text{ form a triangle in } G_t] \,.$$

Difference Operator. We will find it useful to reason about the *difference sequence* (also called the difference stream) of $f : \mathcal{U}_{\pm}^T \to \{\mathbb{R}^n\}^T$ denoted d_f. For a given $\vec{\mathbf{x}}$, we define $d_f(\vec{\mathbf{x}}) \in \{\mathbb{R}^n\}^T$ via $d_f(\vec{\mathbf{x}})[t] = f(\vec{\mathbf{x}})[t] - f(\vec{\mathbf{x}})[t-1]$, where $f(\vec{\mathbf{x}})[-1] \equiv 0^n$.

2.2 Continual Counting and Factorization Mechanisms.

Private (integer) continual counting [19, 11] denotes the continual release problem for releasing all prefix sums on $\vec{\mathbf{x}} \in \mathbb{Z}^T$ with neighboring relation $\vec{\mathbf{x}} \sim \vec{\mathbf{y}}$ if $\|\vec{\mathbf{x}} - \vec{\mathbf{y}}\|_1 \le 1$ and $\|\vec{\mathbf{x}} - \vec{\mathbf{y}}\|_0 \le 1$. The corresponding target function can be expressed as the linear map $f(\vec{\mathbf{x}}) = A\vec{\mathbf{x}}$ where $A \in \{0,1\}^{T \times T}$ is the lower-triangular matrix of all-ones.

State-of-the-art algorithms for this problem are based on *factorization mechanisms* [45]. Before introducing this class, we need some additional concepts. We define the *sensitivity vector set* $\mathcal{S}$, induced by a neighboring relation $\sim$, as

$$\mathcal{S} = \left\{ \vec{\mathbf{x}} - \vec{\mathbf{y}} \,|\, \vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathbb{Z}^T \text{ and } \vec{\mathbf{x}} \sim \vec{\mathbf{y}} \right\} \,.$$

[9] Sometimes also referred to as *user-level* privacy [19]; "item-level privacy" is the terminology used in [38].

In particular, we will extensively reason about $\mathcal{S}_{D,k}$, defined as

$$\mathcal{S}_{D,k} = \left\{ \vec{\Delta} \in \mathbb{Z}^T \ : \ \max_{i \le j \in [T]} \left| \sum_{t=i}^{j} \vec{\Delta}[t] \right| \le D \quad \text{and} \quad \|\vec{\Delta}\|_1 \le k \right\}.$$

This constitutes the set of sensitivity vectors (which we will also call sensitivity streams) with k-bounded ℓ_1 norm, and D-bounded interval sums. We note that the interesting range of k is $[D, TD]$. If the ℓ_1 norm of a stream exceeds TD, then there must exist a value in the stream exceeding D, violating the interval sum property. On the other hand, if the ℓ_1 norm of a stream is smaller than D, then every interval sum automatically has magnitude smaller than D, rendering the interval sum condition vacuous.

For standard continual counting, where $\vec{\mathbf{x}} \sim \vec{\mathbf{y}}$ if $\|\vec{\mathbf{x}} - \vec{\mathbf{y}}\|_1 \le 1$ and $\|\vec{\mathbf{x}} - \vec{\mathbf{y}}\|_0 \le 1$, the sensitivity vector set equals $\mathcal{S}_{1,1}$. More generally, *continual counting on $\mathcal{S}$-streams* denotes the corresponding problem with sensitivity vector set $\mathcal{S}$. We define the associated ℓ_2 *sensitivity* of a matrix $R \in \mathbb{R}^{m \times T}$ by

$$\mathsf{sens}_2(R, \mathcal{S}) = \max_{\vec{\Delta} \in \mathcal{S}} \|R\vec{\Delta}\|_2 .$$

For $\mathcal{S}_{1,1}$, note that $\mathsf{sens}_2(R, \mathcal{S}_{1,1}) = \|R\|_{1 \to 2}$: the largest ℓ_2 norm over all columns of R.

We are now ready to define factorization mechanisms. Given a matrix factorization $A = LR$ where $L \in \mathbb{R}^{T \times m}, R \in \mathbb{R}^{m \times T}$, we define the corresponding ρ-zCDP mechanism $\mathcal{M}_{LR}$ for continual counting on $\mathcal{S}$-streams as follows:

$$\mathcal{M}_{LR}(\vec{\mathbf{x}}) = A\vec{\mathbf{x}} + L\vec{\mathbf{z}} \quad \text{where} \quad \vec{\mathbf{z}} \sim \mathcal{N}\left(0, \ \frac{\mathsf{sens}_2(R,\mathcal{S})^2}{2\rho}\right)^m .$$

Intuitively, the mechanism adds appropriately scaled and correlated Gaussian noise, drawn independently of $\vec{\mathbf{x}}$, to the true output at each step. Because of its simple structure, the error admits exact expressions:

$$\mathsf{MaxSE}(\mathcal{M}_{LR}) = \frac{\|L\|_{2 \to \infty} \cdot \mathsf{sens}_2(R, \mathcal{S})}{\sqrt{2\rho}}, \quad \mathsf{MeanSE}(\mathcal{M}_{LR}) = \frac{\|L\|_F \cdot \mathsf{sens}_2(R, \mathcal{S})}{\sqrt{2T\rho}},$$

where $\|\cdot\|_F$ is the *Frobenius* norm, and $\|\cdot\|_{2 \to \infty}$ is the largest ℓ_2 norm over all rows of the matrix. To emphasize that for a fixed ρ and T, the error is only a function in L, R and $\mathcal{S}$, we will overload notation and write $\mathsf{MaxSE}(L, R, \mathcal{S})$ to denote the corresponding error of $\mathcal{M}_{LR}$. Lastly, we note that analogous mechanisms can be constructed for related notions of differential privacy; see the full version of the paper for more details.

Factorizations for Continual Counting on $\mathcal{S}_{1,1}$-Streams. The classic *binary tree mechanism* of [19, 11] corresponds to an explicit, combinatorial factorization $L_2 R_2 = A$ with error $\Theta(\rho^{-1} \log T)$ for privacy parameter ρ. The more recent explicit *square-root factorization* [35, 25], corresponding to $L = R = \sqrt{A}$, improves over the binary tree mechanism by approximately a factor of ≈ 4.5 and 3.2 for MaxSE and MeanSE respectively. For both MaxSE and MeanSE, the leading constant in the error of the square root factorization mechanism is in fact *exactly* optimal over all factorizations. The matrix $\sqrt{A} \in \mathbb{R}^{T \times T}$ is an example of a *Toeplitz matrix*, meaning that the values on its diagonals are constant. It can thus be characterized by the value of its diagonals $r_0, \dots r_{T-1}$ where $r_j = \binom{2j}{j}/4^j \approx 1/\sqrt{\pi j}$ and $r_0 = 1$. See Appendix A for a more thorough review of related work in continual counting.

Efficient Factorizations. Lastly, we remark that the resource usage of a factorization relates to the hardness of constructing the correlated noise $L\vec{\mathbf{z}}$, entry-by-entry. E.g., if L lacks structure, it may be necessary to store $\Omega(T)$ entries of $\vec{\mathbf{z}}$ to output $(L\vec{\mathbf{z}})[t]$ at each t. In particular, while the binary tree mechanism can be implemented in space $O(\log T)$, no $o(T)$ space implementation is known for $\sqrt{A}\vec{\mathbf{z}}$; in fact, $\Omega(T)$ space is known to be necessary for some models of computation [5].

3 Parametrized Accuracy Guarantees

For all the problems we study in this paper, prior work ([38] for CountDistinct and [50] for DegreeCount and TriangleCount) establishes strong polynomial lower bounds on the accuracy as a function of the time horizon T. Motivated by these results, we pursue a complementary direction, as suggested for counting distinct elements in [38]: proving error bounds in terms of intrinsic parameters of the stream. As we argue in this section, although these parameters may be large in the worst case, they are typically much smaller in realistic streams. Hence, by proving bounds in terms of these parameters, we obtain improved accuracy guarantees for practical streams. For CountDistinct, we prove guarantees in terms of the maximum flippancy parameter introduced in [38], and for DegreeCount and TriangleCount, we define new parameters.

Counting Distinct Elements - Maximum Flippancy. For counting distinct elements, the parameter for which we prove accuracy guarantees is the *maximum flippancy* parameter introduced in [38]. For an item i in the universe $\mathcal{U}$, its flippancy in a stream $\vec{\mathbf{x}}$ is defined as the number of time steps at which the item switches from present to absent, or vice versa. The maximum flippancy of $\vec{\mathbf{x}}$ is the maximum, taken over all items i, of the flippancy of i in $\vec{\mathbf{x}}$.

To motivate this parameter in practice, consider a streaming service that wishes to monitor the number of user accounts currently logged in. This can be modeled as a CountDistinct problem where the universe consists of user accounts: insertions correspond to logins to the account, and deletions correspond to logouts.

Suppose the privacy budget is refreshed annually, and the service computes this statistic every five minutes, yielding a time horizon T on the order of 10^6. Even under an extreme usage pattern, a user might switch between logged-in and logged-out states at most, say, 8 times per day. This implies a maximum flippancy of roughly $8 \times 365 \approx 3000$, which is several orders of magnitude smaller than the time horizon. Moreover, the number of accounts with flippancy exceeding, say, 100 is likely to be very small. Thus, enforcing such a flippancy bound by truncating the stream—e.g., capping the number of recorded state changes per account—would affect the distinct elements statistic by a very small amount.

Degree Histograms - Maximum Degree Contribution. For releasing degree histograms, the parameter for which we prove accuracy guarantees is a new quantity we call the *maximum degree contribution* of an edge. The degree contribution of an edge e in $\vec{\mathbf{x}}$ is defined as the number of time steps t at which e switches from present to absent or vice versa between graphs G_{t-1} and G_t. The maximum degree contribution of $\vec{\mathbf{x}}$ is the maximum, over all edges e, of the degree contribution of e in $\vec{\mathbf{x}}$.

To motivate this parameter in practice, consider a social media service that wishes to monitor the distribution of users' friend counts in order to identify influential accounts. This can be modeled as a DegreeCount problem in which the vertex set corresponds to users, edges correspond to friendships between pairs of users, insertions correspond to forming friendships, and deletions correspond to unfriending. Suppose the privacy budget is refreshed monthly, and the service computes this statistic every minute, yielding a time horizon T on the order of 4×10^5. It is reasonable to expect that even for extreme cases, a particular friendship switches between "friend" and "not friend" at most once a day ≈ 30 times per month (and such a bound could even be enforced by the rules of the platform). Thus, the maximum degree contribution would be several orders of magnitude smaller than the time horizon.

Triangle Counting - Maximum Triangle Contribution. For releasing triangle counts, the parameter for which we prove accuracy guarantees is a new quantity we call the *maximum triangle contribution* of an edge. The triangle contribution of an edge e in $\vec{\mathbf{x}}$ is defined as the sum, over all

Table 1. Comparison of our (MaxSE) accuracy bounds with that from prior work for 1/2-zCDP mechanisms over streams of length T. See Section 3 for definitions and discussion of all the parameters we use.

Problem	Paper	Accuracy Parameter	Accuracy Upper Bounds	Privacy for all streams?
CountDistinct	[38]	Maximum flippancy k_{cd}	$\sqrt{k_{\text{cd}}} \log T$	Yes
	Our Work	Maximum flippancy k_{cd}	$\frac{1}{\pi \log e} \sqrt{k_{\text{cd}}} \log T$	Yes
DegreeCount	[50]	-	$\sqrt{T}$	Yes
	Our Work	Degree contribution k_{deg}	$\frac{\sqrt{2}}{\pi \log e} \sqrt{k_{\text{deg}}} \log T$	Yes
TriangleCount	[50]	Number of vertices n	$n\sqrt{T}$	Yes
	Our Work	Triangle contribution k_{tri}, degree D	$\frac{1}{\pi \log e} \sqrt{k_{\text{tri}} D} \log T$	No

time steps, of the change in the number of triangles involving e at that time step. The maximum triangle contribution of $\vec{\mathbf{x}}$ is the maximum, over all edges e, of the triangle contribution of e in $\vec{\mathbf{x}}$.

To motivate this parameter in practice, again consider a social media service that aims to monitor triangle counts as a measure of community cohesiveness. This can be modeled as a TriangleCount problem in the same graph-streaming framework as in the DegreeCount example above. Suppose the privacy budget is refreshed monthly and the statistic is computed every minute, yielding a time horizon T on the order of 4×10^5.

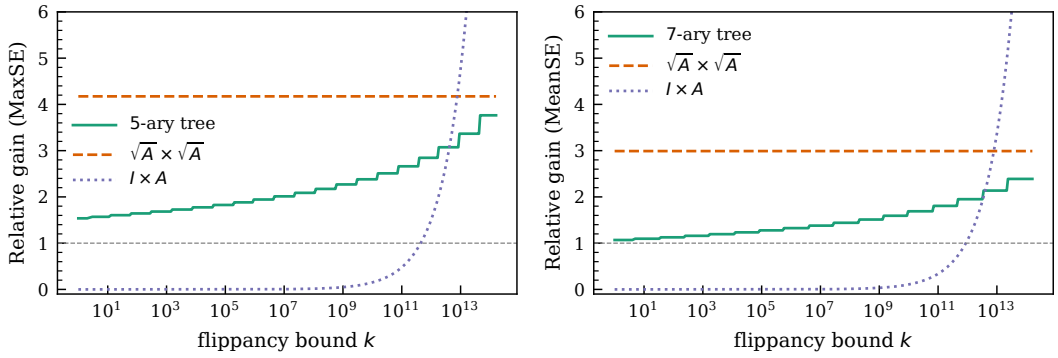
In addition, many social networks explicitly bound the number of friends a user can have (for example, Facebook limits users to 5000 friends), which corresponds to bounding the maximum degree of a vertex. For graph streams in which the graph at each time step has maximum degree at most D, known accuracy guarantees are expressed in terms of the parameter DT,[10] which in this setting can be on the order of 10^7. By contrast, the triangle contribution becomes large only if a single edge participates in many triangles and, moreover, if edges from many of those triangles are updated repeatedly over time. In realistic social networks, this scenario is unlikely: two users typically have only partially overlapping friend sets, and any particular friendship toggles between "friend" and "not friend" relatively infrequently (a property that can also be enforced by policy). Consequently, the maximum triangle contribution is expected to be substantially smaller than the worst-case bound of DT.

4 Our Results

In this summary of our results, we focus on MaxSE; the associated results are summarized in Table 1. The results for root mean squared error are similar and are discussed in the full version of the paper.

Counting Distinct Elements. For counting distinct elements under item-level privacy (neighboring streams differ in occurrences of a single item), prior work of [38] used the binary tree mechanism to obtain a 1/2-zCDP algorithm with root maximum squared error $\sqrt{k_{\text{cd}}} \log T$, where k_{cd} denotes the *maximum flippancy* of the stream (see Section 3 for discussion on this parameter).

[10]This approach only confers privacy for graph streams with degree bounded by D. For privacy for all streams, the best known accuracy is in terms of the quantity nT where n is the number of vertices in the graph.



(a) Relative improvement in MaxSE over [38].

(b) Relative improvement in MeanSE over [38].

Fig. 1. Root maximum and root mean expected squared error comparison between different 1/2-zCDP mechanisms for CountDistinct over $T = 2^{50}$ time steps for a fixed flippancy bound k. Errors are plotted relative to the exact error bounds proved in [38], which equal $\sqrt{k(1+\log T)\log T}$ and $\sqrt{0.5k(1+\log T)(2/T+\log T)}$ for MaxSE and MeanSE respectively. The comparison is done using upper bounds we obtain for b-ary trees with subtraction for the asymptotically optimal choice of b (5 and 7 respectively), the square-root factorization ($\sqrt{A} \times \sqrt{A}$), and the naive factorization ($I \times A$); the bounds we use to plot the graph are in the full version of the paper.

Our work gives bounds in terms of the same parameter, but improves the constant by roughly a factor of 4—see Figure 1 for an exact relative comparison.

Degree Histograms. For releasing degree histograms under item-level privacy (neighboring graph streams differ in all occurrences of a single edge), the best previously known algorithm relied on naive composition (recomputing the histogram at each time step), giving error that scaled polynomially with T. In addition, for notions of error closely related to MaxSE [11], prior work of [50] established strong polynomial accuracy lower bounds in terms of the time horizon T. Hence, our work instead achieves accuracy in terms of a new parameter we introduce: the *maximum degree contribution*, which as argued in Section 3 is much smaller than the number of time steps T in many natural graph streams.

Triangle Counting. For releasing triangle counts under item-level privacy (neighboring graph streams differ in all occurrences of a single edge), the best known algorithm prior to our work relied on composition (giving $n\sqrt{T}$ error for streams with n nodes), and prior work of [50] proved strong accuracy lower bounds in terms of the time horizon T for notions of error closely related to MaxSE. Our work instead obtains accuracy in terms of a new parameter we define: the *maximum triangle contribution*; as argued in Section 3, for typical graph streams with degree bound D over T updates, we expect this quantity to be much smaller than DT. A limitation, reflected in Table 1, is that our guarantee does not provide privacy for *all* graph streams: it holds only for graph streams where the maximum degree at each time step is bounded by a public parameter D and triangle contribution by a public parameter k_{tri}.

Low-Space Error Bounds. We note that one drawback of the above results (obtained using a specific Toeplitz factorization mechanisms) is that they require $\Omega(T)$ space to implement, in

[11] They work with the ℓ_∞ notion of error- $\mathbb{E}[\max_{t\in[T]} \|(f(\vec{x}) - \mathcal{M}(\vec{x}))[t]\|_\infty]$. Our upper bounds give improved error in this notion of error as well; we discuss more in Section 7.

Table 2. Comparison of our (MaxSE) accuracy bounds using 1/2-zCDP tree aggregation mechanisms over streams of length T. See Section 3 for definitions and discussion of all the parameters below.

Problem	Paper	Accuracy Parameter	Accuracy Upper Bounds
CountDistinct	[38]	Maximum flippancy k_{cd}	$\sqrt{k_{\text{cd}}} \log T$
	Our Work	Maximum flippancy k_{cd}	$0.609\sqrt{k_{\text{cd}} \log T \log(T/k_{\text{cd}})}$
DegreeCount	[50]	-	$\sqrt{T}$
	Our Work	Degree contribution k_{deg}	$0.609\sqrt{2k_{\text{deg}} \log T \log(T/k_{\text{deg}})}$
TriangleCount	[50]	Number of vertices n	$n\sqrt{T}$
	Our Work	Triangle contribution k_{tri}, degree D	$0.609\sqrt{k_{\text{tri}} D \log T \log(DT/k_{\text{tri}})}$

contrast with tree-based methods that can be implemented in $O(\log T)$ space.[12] Hence, we give results that leverage improved tree-based methods and novel analysis techniques to maintain $O(\log T)$ space while obtaining improved accuracy. The results are summarized in Table 2.

One important caveat to note is that obtaining privacy for all streams for CountDistinct and DegreeCount while maintaining superior accuracy for realistic streams is done with a pre-processing step that tracks the parameter of interest for the stream and modifies the stream accordingly to satisfy bounds on the parameter. It is not always clear how to track the parameters of interest while using space that is sub-polynomial in the time horizon T. Hence, using tree-based methods alone will not allow us to simultaneously achieve privacy for all streams and logarithmic space for these problems. Nonetheless, these results are significant since they reduce the problem to that of tracking parameters in low space. For example, for CountDistinct, recent work [15] gives a method to track a variant of maximum flippancy[13] with space $\tilde{O}(T^{1/3})$, and using the tree-based methods presented in this section would give improved accuracy while also preserving sublinear space.

Applicability to Other Problems. Before discussing the techniques to obtain all the above results, we briefly comment on their generality. We show how to reduce three different problems to instances of continual counting with a novel neighboring relation we define, and give algorithms for this task. Our techniques apply more widely; we discuss more in Section 8.

5 Technical Overview

We start this section by giving a high level description of our overall algorithm for a function f. This will showcase the general recipe that we use for all our problems. We refer to Section 2 for the notation we use in this section.

5.1 High-Level Overview of Algorithm

For a scalar function f, let d_f be a function that takes a stream $\vec{\mathbf{x}}$ and produces the corresponding difference stream $d_f(\vec{\mathbf{x}})[t] = f(\vec{\mathbf{x}})[t] - f(\vec{\mathbf{x}})[t-1]$. Our continual mechanism $\mathcal{M}$ for f at each time step t, creates an update for the difference stream by computing $f(\vec{\mathbf{x}})[t]$ and subtracting

[12]There is empirical work [47] indicating that for a wide range of values of T, Toeplitz factorizations with the properties we need can be implemented in polylogarithmic space. Accompanying theoretical results are also known in some settings [18], but the factorizations in this work do not satisfy the properties we need.

[13]Cummings et al. [15] call this quantity "occurrency", it corresponds to the maximum number of updates for a single item.

$f(\vec{\mathbf{x}})[t-1]$ from it. It then runs a continual counting mechanism to release partial sums of the difference stream d_f in an online fashion.[14] Hence, the root max (mean) squared error of $\mathcal{M}$ will be equal to the root max (mean) squared error of the continual counting mechanisms used.

We analyze multiple continual counting mechanisms to instantiate the above recipe. All of these are different factorization mechanisms. We point the reader to Section 2.2 for an overview of continual counting and factorization mechanisms. Recall that for a factorization mechanism, the error depends heavily on the sensitivity vector set $\mathcal{S}$ induced by the neighboring relation. In our setting, for a function f, the factorization mechanisms are applied to the difference stream d_f, and so we reason about the sensitivity vectors of the difference stream induced by the upstream neighboring relation. For a fixed function f, we will refer to this set of sensitivity vectors as the sensitivity vector set for our function f.

5.2 Sensitivity Vector Sets for Cardinality Estimation Functions

Fix a function f and two neighboring streams $\vec{\mathbf{x}}, \vec{\mathbf{y}}$; the corresponding sensitivity vector is $\vec{\Delta}_f(\vec{\mathbf{x}}, \vec{\mathbf{y}}) = d_f(\vec{\mathbf{x}}) - d_f(\vec{\mathbf{y}})$ (we will drop the parameters and simply say $\vec{\Delta}_f$ where clear). Our first main contribution is characterizing properties of sensitivity vectors $\vec{\Delta}_f$ for our functions that we can leverage to analyze the continual counting mechanisms of interest. Recall the definition of $\mathcal{S}_{D,k}$ from Section 2.2—it is the set of all real vectors with interval sums bounded between $-D$ and D and ℓ_1 norm bounded by k. Looking forward, we essentially show that the sensitivity vector sets for all our functions are subsets of $\mathcal{S}_{D,k}$ for appropriately chosen D and k.

- **All sensitivity vectors have bounded interval sums**. For CountDistinct and DegreeCount, we identify a crucial property of the sensitivity vectors that we leverage in our proofs of error bounds—they are *alternating*. That is, the non-zero entries alternate; if a non-zero entry of $\vec{\Delta}_{\mathsf{CountDistinct}}$ is 1, the next non-zero entry has to be -1 (and vice versa). Similarly, for DegreeCount, we observe that there are two indices a and b such that the sensitivity vector $\vec{\Delta}_{\mathsf{DegreeCount}}$ restricted to either of these two indices also has the same alternating property. For every other index c, $\vec{\Delta}_{\mathsf{DegreeCount}}[c] = 0^T$. We note that the 'alternating' property is also equivalent to all interval sums of the sensitivity vector being bounded between -1 and 1. For the function TriangleCount, we show that for graph streams with degree D, while the sensitivity vectors of the difference streams for this function are not "alternating", they satisfy the more general property that all the interval sums are bounded between $-D$ and D.
- **Realistic sensitivity vectors have bounded ℓ_1 norm.** A key goal for us is to obtain parametrized accuracy guarantees (See Section 3). To do this, we leverage the fact that the set of updates for a single item typically has a *bounded contribution* on the output statistic—which motivated the parameters we selected. We show that such bounded contribution is captured by the ℓ_1 norm of the sensitivity vector. We argue that for all the functions we consider, if the corresponding streams have their associated parameters bounded by k, then the ℓ_1 norms of every associated sensitivity vector is bounded by k (for DegreeCount, the ℓ_1 norm of the sensitivity vector restricted to each vertex is bounded by k).

Let $k_{\text{deg}}, k_{\text{cd}}, k_{\text{tri}}$ be the parameter bounds for DegreeCount, CountDistinct, and TriangleCount respectively (see Section 3 for definitions). By the above discussion, we have that the sensitivity vector sets for DegreeCount restricted to each vertex and CountDistinct are subsets of $\mathcal{S}_{1,k_{\text{deg}}}$ and $\mathcal{S}_{1,k_{\text{cd}}}$ respectively. Similarly, the sensitivity vector set for TriangleCount on graph streams with degree bounded by D is a subset of $\mathcal{S}_{D,k_{\text{tri}}}$. The next section will focus on continual counting on

[14]For vector valued functions f, a similar recipe applies, except multiple continual counting mechanisms are used to release the partial sums of each coordinate of the difference stream.

$\mathcal{S}_{D,k}$-streams, which we define as continual counting with differential privacy defined with respect to the $\mathcal{S}_{D,k}$-neighboring relation.

5.3 Continual Counting on $\mathcal{S}_{D,k}$-Streams

We now study continual counting on $\mathcal{S}_{D,k}$-streams via *factorization mechanisms*, thereby giving bounds for our cardinality estimation problems of interest. From the discussion on sensitivity vector sets above, and recalling facts from Section 2.2 on factorization mechanisms, we have that for a factorization of the all-ones matrix $A = LR$ the errors of the corresponding factorization mechanism are completely captured by the sensitivity $\mathsf{sens}_2(R, \mathcal{S}_{D,k}) = \max_{\vec{\Delta} \in \mathcal{S}_{D,k}} \|R\vec{\Delta}\|_2$, and matrix norms on L. Our technical arguments reason about the first quantity, and rely on past work for the second.

Leveraging Toeplitz Factorization Mechanisms. We prove the following main theorem bounding $\mathsf{sens}_2(R, \mathcal{S}_{D,k})$ under some restrictions on the right factorization matrix R.

Theorem 5.1. *Let $T, k, D \in \mathbb{N}$. Let R be a real-valued lower-triangular $T \times T$ Toeplitz matrix with non-increasing and non-negative lower-diagonal values. Let $\vec{\Delta} \in \mathcal{S}_{D,k}$. Then,*[15]

$$\|R\vec{\Delta}\|_2 \leq \sqrt{kD} \cdot \|R\|_{1\to 2}\,.$$

where $\|R\|_{1\to 2}$ is the maximum ℓ_2 norm of any column in matrix R.

While we do not leverage it in this paper, we remark that this theorem also holds if $\mathcal{S}_{D,k}$ is defined over $\mathbb{R}^T$ (with the same bounds on interval sums and ℓ_1 norm) as the proof given in Section 6 does not rely on the integrality of vectors in $\mathcal{S}_{D,k}$. We also note that the case when $k = D = 1$, has been studied extensively in prior work [25, 35, 34, 18, 30]. It is known from this work that the square-root factorization where $L = R = \sqrt{A}$ satisfies the properties needed to apply Theorem 5.1, giving the upper bound in the following result.

Theorem 5.2 (Informal Error Bounds for Continual Counting on $\mathcal{S}_{D,k}$-streams). *Fix $T, k, D \in \mathbb{N}$. Let A be the lower-triangular all-ones matrix (of size $T \times T$). Then, the factorization mechanism with $L = R = \sqrt{A}$ achieves the following error bounds for $1/2$-zCDP continual counting on $\mathcal{S}_{D,k}$-streams of length T:*

$$\mathsf{MaxSE}(L, R, \mathcal{S}_{D,k}) \leq \left(\frac{1}{\pi \log(e)} + o(1)\right) \sqrt{kD} \log T\,,$$
$$\mathsf{MeanSE}(L, R, \mathcal{S}_{D,k}) \leq \left(\frac{1}{\pi \log(e)} + o(1)\right) \sqrt{kD} \log T\,.$$

Additionally, if $k = O(DT^{1/3})$ and $D \leq k \leq DT$, we have that

$$\mathsf{MaxSE}(L, R, \mathcal{S}_{D,k}) \geq \left(\frac{1}{\pi \log(e)} - o(1)\right) \sqrt{kD \log(DT/k) \log T}\,,$$
$$\mathsf{MeanSE}(L, R, \mathcal{S}_{D,k}) \geq \left(\frac{1}{\pi \log(e)} - o(1)\right) \sqrt{kD \log(DT/k) \log T}\,.$$

Note that the lower bound indicates that our analysis of the square-root factorization is tight up to lower order terms for a wide range of values of k and D. We conjecture that the dependence on T in the lower bound (specifically $\sqrt{\log(DT/k) \log T}$ vs $\log T$) is tight and that directly analyzing the square root matrix (instead of factoring through our general result for a broad class of Toeplitz matrices) might give this improved bound. We also note that when $k = D = 1$, our results recover

[15] If $k \leq D$, then the interval sum bound on $\vec{\Delta}$ is vacuous. In this case, for any matrix R, $\mathsf{sens}_2(R, \mathcal{S}_{D,k}) = k\|R\|_{1\to 2} \leq \sqrt{kD}\|R\|_{1\to 2}$ (see e.g., proof of [35, Fact 2.1] for the first step). Hence, the interesting range for the theorem is when $k > D$.

the tightest bounds on the square-root factorization for standard continual counting [30]. Carefully applying the above general result to our problems of interest gives the results in Table 1.

Leveraging Tree-Based Factorization Mechanisms. As mentioned in Section 4, to obtain improved error bounds while preserving low (logarithmic in the time horizon) space, we study tree aggregation factorization mechanisms. Building on work by [2, 3] on improved tree-aggregation mechanisms, as well as exploiting additional redundancy in the tree in our setting, we introduce a factorization $A = \hat{L}_b \hat{R}_b$, and study the quantity $\mathsf{sens}_2(\hat{R}_b, \mathcal{S}_{D,k})$, thereby characterizing the error notions of interest.

Theorem 5.3 (Informal error bounds for tree-based mechanisms). *Let $T, D, k, b \in \mathbb{N}$. The b-ary tree mechanism, with subtraction (with associated factors $\hat{L}_b, \hat{R}_b$) achieves the following error bounds for $1/2$-zCDP continual counting on $\mathcal{S}_{D,k}$-streams of length T:*

$$\mathsf{MaxSE}(\hat{L}_b, \hat{R}_b, \mathcal{S}_{D,k}) \leq \left(\frac{\sqrt{b-1}}{\sqrt{2}\log b} + o(1) \right) \sqrt{Dk \log(DT/k) \log(T)}\,,$$

$$\mathsf{MeanSE}(\hat{L}_b, \hat{R}_b, \mathcal{S}_{D,k}) \leq \left(\frac{\sqrt{b(1-1/b^2)}}{2\log b} + o(1) \right) \sqrt{Dk \log(DT/k) \log(T)}\,.$$

where the constant in the parenthesis is minimized for $b = 5$ and $b = 7$ respectively, with corresponding values 0.609 and 0.466.

We give closed form lower bounds for $\mathsf{sens}_p(\hat{R}_b, \mathcal{S}_{D,k})$ for $p \in \mathbb{N}$ that match our upper bounds up to lower order terms. We also give an exact polytime approach for computing $\mathsf{sens}_p(\hat{R}_b, \mathcal{S}_{D,k})$ via dynamic programming, which can be leveraged in practice to obtain exact bounds. Using the ℓ_1 sensitivity, we also derive state-of-the-art bounds, analogous to Theorem 5.3, for other notions of differential privacy. Detailed formal statements can be found in the full version of the paper.

5.4 Techniques Used in Proofs

A Reduction from $\mathcal{S}_{D,k}$ to $\mathcal{S}_{1,k}$. A key ingredient used throughout our proofs is a reduction from reasoning about streams in $\mathcal{S}_{D,k}$ to streams in $\mathcal{S}_{1,k}$. We show the following in the full version of the paper:

Theorem 5.4. *Let $T, D, k, m \in \mathbb{N}$ where $k \leq DT$, $\mathcal{S}_{D,k} \subset \mathbb{Z}^T$ and $R \in \mathbb{R}^{m \times T}$. Then,*

$$D \cdot \mathsf{sens}_2(R, \mathcal{S}_{1,\lfloor k/D \rfloor}) \leq \mathsf{sens}_2(R, \mathcal{S}_{D,k}) \leq \max_{k_1,\ldots,k_D : \sum_{\ell=1}^{D} k_\ell = k} \sum_{\ell=1}^{D} \mathsf{sens}_2(R, \mathcal{S}_{1,k_\ell})\,.$$

The lower bound in this result is straightforward and follows from simple facts about the scaling of norms. The upper bound is more intricate and involves breaking up a sensitivity stream $\vec{\Delta}$ from $\mathcal{S}_{D,k}$ into up to D different sensitivity streams $\vec{\Delta}^{(\ell)}$ from $\mathcal{S}_{1,k_\ell}$, where $\sum_{\ell=1}^{D} k_\ell \leq k$. Each stream corresponds to a distinct value that a prefix sum of $\vec{\Delta}$ can take, and the streams are carefully created such that $\sum_{\ell=1}^{D} \vec{\Delta}^{(\ell)} = \vec{\Delta}$. Combining structural properties of this decomposition with basic norm inequalities yields the upper bound.

This reduction proves to be crucial in our analysis because streams in $\mathcal{S}_{1,k}$ enjoy a much more rigid structure than those in $\mathcal{S}_{D,k}$: they are *alternating*, meaning their non-zero entries have magnitude 1 and alternate in sign.

Toeplitz Factorizations (Upper Bound). We overview our techniques for Toeplitz factorizations next, beginning with upper bounds. Our paper has two proofs of Theorem 5.1, each of different flavor. One proof is combinatorial, the other more algebraic. Via the known link between sensitivity $\mathsf{sens}_2(R, \mathcal{S}_{D,k})$ and factorization error (recall Section 2.2), applying this to the square-root factorization yields the error upper bounds of Theorem 5.2.

Let $c(i,j) = (R^\top R)[i,j]$. For a sensitivity stream $\vec{\Delta} \in \mathcal{S}_{D,k}$, the starting point of both proofs is analyzing the quadratic form $\|R\vec{\Delta}\|_2^2$. As a precursor, we show that monotonicity of lower diagonal values of R induces useful monotonicity properties of $c(\cdot,\cdot)$ that we use in both proofs.

The combinatorial proof proceeds by first considering the alternating case - that is, considering a sensitivity stream $\vec{\Delta} \in \mathcal{S}_{1,k}$. We decompose the quadratic form into (i) a "squared term" of the form $c(i,i)\vec{\Delta}[i]^2$ and (ii) a "cross term" involving $c(i,j)\vec{\Delta}[i]\vec{\Delta}[j]$ for $i \neq j$. The key step is to show that the cross term is non-positive. Using the alternating structure, whenever $\vec{\Delta}[i]\vec{\Delta}[j] = 1$ for $i < j$, we can match this pair with another $i < j'$ where $\vec{\Delta}[i]\vec{\Delta}[j'] = -1$ and j' lies closer to i. Monotonicity properties of $c(\cdot,\cdot)$ then let us show that each matched pair contributes a non-positive amount to the cross term. This helps us prove Theorem 5.1 for $D = 1$, which we then extend to all D using the reduction from $\mathcal{S}_{D,k}$ streams to $\mathcal{S}_{1,k}$ streams in Theorem 5.4.

The second proof is linear algebraic and reasons about $\mathcal{S}_{D,k}$ directly; we find it more elegant, but much less intuitive. A key quantity that we work with is $\sum_{j=0}^{T-1} c(i,j)\vec{\Delta}[j]$. Writing $SUM_j = \sum_{i=0}^{j} \vec{\Delta}[i]$, we write $\vec{\Delta}[j]$ as a difference of adjacent prefix sums $SUM_{j+1} - SUM_j$, and then change indices of the sum- $\sum_{j=0}^{T-1} c(i,j)(SUM_{j+1} - SUM_j) = \sum_{j=0}^{T-1} [c(i,j) - c(i,j+1)] \cdot SUM_j$, a transformation commonly called *Abel summation* and *summation by parts*. It turns out that the latter quantity is easier to reason about via the monotonicity of $c(\cdot,\cdot)$, and we leverage this carefully to prove the theorem.

Square-Root Factorization (Lower Bound). For the lower bound on $\mathsf{sens}_2(\sqrt{A}, \mathcal{S}_{D,k})$, (and consequently error of $\sqrt{A} \times \sqrt{A}$), we use the probabilistic method. We first treat $\mathcal{S}_{1,k}$ streams: draw sensitivity vector $\vec{\Delta}$ uniformly at random from $\mathcal{S}_{1,k}$ (conditioned on $\|\vec{\Delta}\|_1 = k$) and analyze $\mathbb{E}\left[\|\sqrt{A} \cdot \vec{\Delta}\|_2^2\right]$. We split the expectation into a squared term involving $\mathbb{E}[\vec{\Delta}[i]^2]$ and a cross term involving $\mathbb{E}[\vec{\Delta}[i]\vec{\Delta}[j]]$ for $i \neq j$. The former is easy to reason about because each coordinate of $\vec{\Delta}$ is equally likely to be non-zero. Bounding the cross term ends up being the key challenge of the proof, which boils down to analyzing the probability that $\vec{\Delta}[i] \neq \vec{\Delta}[j]$, conditioned on them being non-zero. This corresponds to the number of non-zero values falling between i and j being even, which is exactly the parity of a hypergeometric random variable. We hence prove a result characterizing this probability (which may be of independent interest) and use it to get a handle on the cross term. Controlling the absolute value of the cross term ends up requiring intricate analytical manipulation. This gives the results for $\mathcal{S}_{1,k}$ streams; extending to $\mathcal{S}_{D,k}$ streams again uses Theorem 5.4.

Tree-Aggregation-Based Factorizations. For tree aggregation, we work directly with the combinatorial structure of the tree and again begin with streams from $\mathcal{S}_{1,k}$, exploiting their alternating property. We reduce the sensitivity analysis (for $\mathsf{sens}_2(R, \mathcal{S}_{1,k})$) to a new combinatorial problem we call *parity counting on trees*: given a complete b-ary tree with T leaves and k balls placed on leaves, how many internal nodes have an odd number of balls in their subtree? Due to the alternating structure of streams from $\mathcal{S}_{1,k}$, maximizing this quantity yields an exact characterization of the sensitivity $\mathsf{sens}_2(R, \mathcal{S}_{1,k})$ of the tree-aggregation matrix.

We use this characterization to derive upper and lower bounds that are tight up to lower order terms and to design a dynamic programming algorithm that computes the $\mathsf{sens}_2(R, \mathcal{S}_{1,k})$ exactly in

polynomial time. Our combinatorial view also motivates a sparsified tree factorization tailored to our setting: since standard tree factorizations use only a $(1 - 1/b)$ fraction of nodes, an equivalent fraction of nodes with an odd number of balls in their subtree can be ignored when computing the sensitivity. Combining these insights with the reduction from reasoning about $\mathcal{S}_{D,k}$ streams to $\mathcal{S}_{1,k}$ streams yields Theorem 5.3 and related results.

5.5 Challenges in Analyzing Factorization Mechanisms for $\mathcal{S}_{D,k}$-Streams

For continual counting on $\mathcal{S}_{D,k}$-streams, the case when $k = D = 1$ has been studied extensively in prior work. Here, the square-root factorization achieves near-optimal bounds [25, 35]. One might therefore wonder whether those techniques extend naturally to $k > 1$ (even when $D = 1$). In this section, we explain why our different analytical approaches appear necessary even for $\mathcal{S}_{1,k}$-streams.

The error of a factorization mechanism $\mathcal{M}_{LR}$ for $\mathcal{S}_{1,k}$-streams depends on $\mathrm{sens}_2(R, \mathcal{S}_{1,k}) = \max_{\vec{\Delta} \in \mathcal{S}_{1,k}} \|R\vec{\Delta}\|_2$, a natural optimization problem. When $k = 1$, its objective value exactly equals $\|R\|_{1\to 2}$: the largest ℓ_2 norm of any column in R, a value that can be computed efficiently. A natural first attempt to handle larger k might be to reduce to $k = 1$ by appealing to an operator norm bound: $\|R\vec{\Delta}\|_2 \le \|R\|_{1\to 2}\|\vec{\Delta}\|_1 \le k\|R\|_{1\to 2}$, but this gives a *linear* dependence on k (as opposed to the $\sqrt{k}$ dependence we target). Hence, we first opt for looking more closely at the optimization problem.

Computational Challenges for Higher k. Note that, even in this simplified setting where $D = 1$, naively computing $\mathrm{sens}_2(R, \mathcal{S}_{1,k}) = \max_{\vec{\Delta} \in \mathcal{S}_{1,k}} \|R\vec{\Delta}\|_2$ is computationally prohibitive. A naive brute-force enumeration would have to iterate over $\ge \binom{T}{\min(\lceil T/2 \rceil, k)}$ vectors: a set of size exponential in k. Thus, unlike the $k = 1$ setting, we cannot hope to solve the optimization problem by direct enumeration. Moreover, a convex quadratic *maximization* problem over a discrete set can be NP-hard—we have no reason to believe that $\mathrm{sens}_2(R, \mathcal{S}_{1,k})$ can be computed efficiently and exactly for arbitrary k and R. Some matrices R may be innately hard once $k = \omega(1)$.

Identifying the Maximizing Sensitivity Vector via Structure of Matrix. One may instead hope to identify the maximizer for particular structured matrices, like for the square-root matrix $R = \sqrt{A}$. Indeed, experiments on the square-root matrix for various values of k exhibit an encouraging heuristic pattern: the nonzero entries of the maximizer $\vec{\Delta} \in \mathcal{S}_{1,k}$ tend to be somewhat evenly spaced (see Figure 2 for $T = 30$ and $k \in [1, 10]$).

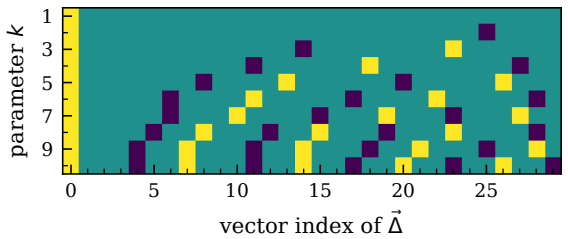


Fig. 2. Heatmap showing the sensitivity vector $\vec{\Delta} \in \mathcal{S}_{1,k}$ which maximizes $\|\sqrt{A}\vec{\Delta}\|_2$ for each $k \in [1, 10]$ and $n = 30$. The yellow squares represent 1, the purple squares represent -1 and the teal squares represent 0.

Unfortunately, there does not appear to be a clean closed-form description of these vectors. The alternating constraint introduces subtle interactions: to optimize $\|\sqrt{A}\vec{\Delta}\|_2$, we are interested in the norm of a linear combination of k columns (with some columns added, others subtracted). Inserting a -1 at a particular position in $\vec{\Delta}$ can increase the absolute value of some coordinates in $\sqrt{A}\vec{\Delta}$ (e.g.,

by making a previously small negative entry more negative), while simultaneously decreasing others (e.g., by cancelling positive contributions). How these effects trade off depends opaquely on the values of the matrix $\sqrt{A}$.

Lacking a usable closed-form characterization for the square-root, we ultimately abandoned the direct optimization approach from prior work and instead developed the combinatorial and algebraic upper-bound techniques and the probabilistic lower bound techniques described earlier.

Finally, we remark that while the sensitivity of $R = \sqrt{A}$ appears challenging to compute exactly, the same is not true for all structured matrices. Indeed, we give a polynomial-time algorithm for tree-structured R, and when $R = A$ the problem is trivial.

6 Algebraic Proof of Theorem 5.1

In this section, we give the algebraic proof of Theorem 5.1. Let R be any lower-triangular Toeplitz matrix with non-negative and non-increasing lower diagonal entries. Let $c(i,j) \equiv (R^\top R)[i,j]$. In the full version of the paper, we prove the following general monotonicity properties for $c(\cdot,\cdot)$ using the monotonicity properties of R.

Claim 6.1. *$c(i,j)$ satisfies the following properties:*

(1) **Symmetry.** *For all $0 \le i,j \le T-1$, we have that $c(i,j) = c(j,i)$.*
(2) **Monotonicity along row (beyond diagonal entry).** *Let $0 \le i \le j \le T-1$. For $0 \le m \le T-j-1$, we have that $c(i,j) \ge c(i,j+m)$.*
(3) **Monotonicity of diagonal entries.** *Let $0 \le m \le j \le T - max(i,j) - 1$. We have that $c(i,j) \ge c(i+m, j+m)$.*

Now, using these monotonicity properties, we are ready to prove the upper bound in Theorem 5.1.

Proof via Summation by Parts. Consider arbitrary $\vec{\Delta} \in \mathcal{S}_{D,k} \subseteq \mathbb{R}^T$. We can write

$$\|R\vec{\Delta}\|_2^2 = \sum_{i,j\in[T]} c(i,j)\vec{\Delta}[i]\vec{\Delta}[j] = \sum_{i=0}^{T-1} \vec{\Delta}[i] \underbrace{\sum_{j=0}^{T-1} c(i,j)\vec{\Delta}[j]}_{Q_i},$$

where we continue working with the inner sum. Define the infinite sequences $(f_t)_{t\in\mathbb{Z}}, (g_t)_{t\in\mathbb{Z}}$ with bounded support:

$$f_t = \begin{cases} c(i,t) & \text{for } 0 \le t \le T-1, \\ 0 & \text{otherwise}, \end{cases} \qquad g_t = \begin{cases} \vec{\Delta}[t] & \text{for } 0 \le t \le T-1, \\ 0 & \text{otherwise}. \end{cases}$$

Defining $G_m = \sum_{t=-\infty}^{m} g_t$, we write

$$Q_i = \sum_{j=0}^{T-1} c(i,j)\vec{\Delta}[j] = \sum_{t\in\mathbb{Z}} f_t(G_t - G_{t-1}) = \sum_{t\in\mathbb{Z}} f_t G_t - \sum_{t\in\mathbb{Z}} f_t G_{t-1} = \sum_{t\in\mathbb{Z}} (f_t - f_{t+1}) G_t,$$

where the last step changes the indexing in the second sum and then merges the sums. This transformation is sometimes called *summation by parts*. By the monotonicity, non-negativity and symmetry of $c(i,j)$ from Claim 6.1, we have that $c(i,j) - c(i,j+1) \ge 0$ for $j \ge i$, and

$c(i,j) - c(i,j+1) \leq 0$ for $j < i$. We can thus write

$$\begin{aligned} Q_i &= \sum_{j \geq i} \left[c(i,j) - c(i,j+1)\right] G_j - \sum_{j<i} \left[c(i,j+1) - c(i,j)\right] G_j \\ &\leq \sup_{j \in \mathbb{Z}}(G_j) \sum_{j \geq i} \left[c(i,j) - c(i,j+1)\right] - \inf_{j \in \mathbb{Z}}(G_j) \sum_{j<i} \left[c(i,j+1) - c(i,j)\right] \\ &= \left(\sup_{j \in \mathbb{Z}}(G_j) - \inf_{j \in \mathbb{Z}}(G_j)\right) c(i,i)\,, \end{aligned}$$

where the inequality uses that $\inf_k(b_k) \sum_k a_k \leq \sum_k a_k b_k \leq \sup_k(b_k) \sum_k a_k$ for a nonnegative sequence $(a_k)_k$, and the final equality that the sums telescope. By an analogous argument (swap the role of the sums), we can also derive a lower bound:

$$Q_i \geq \left(\inf_{j \in \mathbb{Z}}(G_j) - \sup_{j \in \mathbb{Z}}(G_j)\right) c(i,i)\,.$$

Combining the two bounds, we get

$$|Q_i| \leq \left(\sup_{j \in \mathbb{Z}}(G_j) - \inf_{j \in \mathbb{Z}}(G_j)\right) c(i,i) \leq D \cdot c(i,i)\,,$$

where the inequality uses that G_j is a prefix on $\vec{\Delta}$, and so the difference $\sup_{j \in \mathbb{Z}}(G_j) - \inf_{j \in \mathbb{Z}}(G_j)$ is a (signed) *interval sum on* $\vec{\Delta}$, and so bounded by D from the definition of $\mathcal{S}_{D,k}$.

We thus have (taking absolute value of the summand)

$$\|R\vec{\Delta}\|_2^2 = \sum_{i=0}^{T-1} \vec{\Delta}[i] Q_i \leq D \sum_{i=0}^{T-1} |\vec{\Delta}[i]| c(i,i)\,.$$

Using that $c(i,i) \leq c(0,0) = \|R\|_{1\to 2}^2$ and that $\|\vec{\Delta}\|_1 \leq k$ by definition:

$$\|R\vec{\Delta}\|_2^2 \leq D \cdot \|R\|_{1\to 2}^2 \sum_{i=0}^{T-1} |\vec{\Delta}[i]| \leq Dk \cdot \|R\|_{1\to 2}^2\,.$$

Invoking the final bound for the sensitivity finishes the proof:

$$\mathsf{sens}_2(R, \mathcal{S}_{D,k}) = \max_{\vec{\Delta} \in \mathcal{S}_{D,k}} \|R\vec{\Delta}\|_2 \leq \sqrt{Dk} \cdot \|R\|_{1\to 2}\,. \qquad \square$$

7 Directly Related Work.

The most directly related work is that of [38] who study counting distinct elements in the fully dynamic streaming model and introduce the maximum flippancy parameter. Their setting differs slightly from ours (neighboring streams in their notion are at most distance-2 away according to ours), but their algorithm extends to our model, and we compare the resulting bounds against ours. As shown in Figure 1, our error improves over theirs across the full range of the maximum flippancy parameter. An alternative parameterization based on total flippancy was proposed by Henzinger et al. [32]; while we focus on maximum flippancy—typically smaller for realistic streams—our improvements apply in that setting as well.

For TriangleCount and DegreeCount, Raskhodnikova and Steiner [50] give strong polynomial (in T) lower bounds under item-level DP for expected ℓ_∞ error. Their model assumes only a single update per time step, and so while their lower bounds continue to apply in our setting, their upper bounds that are based on periodic recomputation do not apply. To the best of our knowledge, the

previous best upper bounds for these problems come from naively running a batch DP algorithm at each time step and applying composition theorems. We introduce new parameters that are significantly smaller than the time horizon in realistic streams and obtain substantially improved error guarantees in terms of these parameters. Although our accuracy guarantees are stated in terms of MaxSE and MeanSE, they also yield improvements for ℓ_∞ error via standard Gaussian tail bounds (see, for example, Section 2.4 of [18]), providing a way around the lower bounds in [50] for realistic streams.

We also note that there is additional incomparable work on releasing TriangleCount and DegreeCount [51, 24, 40]. These works focus on the insertion-only setting, which is less demanding than the fully dynamic case we study, though some of this work (e.g., [40]) addresses the more challenging node-privacy model, whereas we focus on edge privacy. Moreover, [40] gives a TriangleCount mechanism that is private for all insertion-only graph sequences, while our mechanism for TriangleCount on fully dynamic streams is private only for a restricted class of sequences.

Finally, we emphasize that our focus differs from all of the above: we aim to obtain strong asymptotic guarantees and to optimize constants, whereas prior work targets asymptotics alone.

8 Conclusion

Our work gives improved error bounds for a range of cardinality estimation problems. We do so by recasting private cardinality estimation on fully dynamic streams as private linear queries over their difference streams and studying the associated sensitivity vector sets in detail. There are many possible directions for future work.

Triangle Counting with Privacy for All Graphs. As already mentioned, the privacy afforded by reducing triangle counting to continual counting on the difference stream is conditioned on the graph stream having degree bounded by D and maximum triangle contribution bounded by k_{tri}. Coming up with an algorithm that guarantees privacy for all graphs, but continues to give accuracy matching our approach for streams satisfying the parameter constraints, is a natural open question.

Beyond Cardinality Estimation. We emphasize that our algorithmic techniques are not specific to the problems studied here, and apply more broadly than cardinality estimation alone. As a simple example, consider tracking the total number of items in users' shopping carts on an e-commerce platform, where the multiplicities of items matter beyond just their presence or absence (unlike in the cardinality estimation problems we study). Each user generates a stream of updates to their cart size, which we may assume always lies in $[0, D]$ and changes by at most k items per session. Providing user-level differential privacy for this task then reduces naturally to continual counting over $\mathcal{S}_{D,k}$-streams. More generally, private continual release of an (integer-valued) function f, on input streams with neighboring relation $\sim$, can be reduced to continual counting on $\mathcal{S}_{D,k}$-streams if, for all neighboring input streams $\vec{\mathbf{x}} \sim \vec{\mathbf{y}}$ to f, we have that:

(1) $f(\vec{\mathbf{x}}) - f(\vec{\mathbf{y}}) \in [-D, 0]^T \cup [0, D]^T$,
(2) $\|d_f(\vec{\mathbf{x}}) - d_f(\vec{\mathbf{y}})\|_1 \leq k$.

We leave the exploration of additional applications of our techniques to future work.

Other Linear Queries. While this paper exclusively deals with continual counting applied to difference streams, alternative linear queries on these streams might be interesting to explore. For example, instead of computing the exact number of distinct elements, one might be interested in just knowing whether the number of distinct elements *recently has been increasing or decreasing*. Such a statistic could naturally be derived from postprocessing the running count, but it might be more efficient to directly adapt the query matrix to the statistic in mind. Studying such query matrices via our framework is a natural direction for future work.

Space Efficiency. There is evidence that the square-root factorization we employ does not admit any sublinear space algorithm [5]. Recent work [18] introduced a Toeplitz factorization approximating its error up to an additive constant in space $O(\log^2 T)$, but their construction does not enforce a monotonicity condition on the diagonals of the right factor. Designing an explicit space-efficient Toeplitz factorization with such a monotonicity constraint and near-optimal error for standard continual counting remains an open question, and would give error bounds for continual counting on $\mathcal{S}_{D,k}$-streams via our results.

Lower Bounds. Finally we raise the question of establishing lower bounds for factorization mechanisms on $\mathcal{S}_{D,k}$-streams. We know that for $\mathcal{S}_{1,1}$, the square-root factorization is leading-constant optimal for our error metrics [46, 25, 35], but it clearly sub-optimal for $\mathcal{S}_{1,T}$ (the naive factorization $R = A, L = I$ is better in this regime). As such, different factorizations are expected to be optimal for different parameter regimes.

Acknowledgments

We thank Nikita Kalinin and Adam Smith for valuable comments on an early draft of this paper, and Sofya Raskhodnikova for valuable discussions on the results of [50].

Joel Daniel Andersson was funded by the European Union. In particular, he received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (MoDynStruct, No. 101019564), with additional funding from Providentia, a Data Science Distinguished Investigator grant from Novo Nordisk Fonden. Part of the work was carried out while visiting Boston University.

Palak Jain was supported in part by Boston University's Hariri Institute for Computational Science and Engineering.

Satchit Sivakumar was supported by the Apple Scholars in AI/ML PhD fellowship.

References

[1] Aditya Akella, Ashwin R. Bharambe, Michael K. Reiter, and Srinivasan Seshan. 2003. Detecting DDoS attacks on ISP networks. In *Workshop on Management and Processing of Data Streams*.

[2] Joel Daniel Andersson and Rasmus Pagh. 2023. A smooth binary mechanism for efficient private continual observation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. http://papers.nips.cc/paper%5C_files/paper/2023/hash/99c41fb9fd53abfdd4a0259560ef1c9d-Abstract-Conference.html.

[3] Joel Daniel Andersson and Rasmus Pagh. 2025. Streaming private continual counting via binning. In *IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025, Copenhagen, Denmark, April 9-11, 2025*. IEEE, 575–589. doi:10.1109/SATML64287.2025.00038.

[4] Joel Daniel Andersson, Rasmus Pagh, Teresa Anna Steiner, and Sahel Torkamani. 2025. Count on your elders: laplace vs gaussian noise. In *6th Symposium on Foundations of Responsible Computing, FORC 2025, June 4-6, 2025, Stanford University, CA, USA* (LIPIcs). Vol. 329. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 10:1–10:24. doi:10.4230/LIPICS.FORC.2025.10.

[5] Joel Daniel Andersson and Amir Yehudayoff. 2025. On the space complexity of online convolution. *CoRR*, abs/2505.00181. arXiv: 2505.00181. doi:10.48550/ARXIV.2505.00181.

[6] Daniel N. Baker and Ben Langmead. 2018. Dashing: fast and accurate genomic distances with hyperloglog. *Genome Biology*, 20.

[7] Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. 2008. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*. Ying Li, Bing Liu, and Sunita Sarawagi, (Eds.) ACM, 16–24. doi:10.1145/1401890.1401898.

[8] Jean Bolot, Nadia Fawaz, S. Muthukrishnan, Aleksandar Nikolov, and Nina Taft. 2013. Private decayed predicate sums on streams. In *Joint 2013 EDBT/ICDT Conferences, ICDT '13 Proceedings, Genoa, Italy, March 18-22, 2013*. Wang-Chiew Tan, Giovanna Guerrini, Barbara Catania, and Anastasios Gounaris, (Eds.) ACM, 284–295. doi:10.1145/2448496.2448530.

[9] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: simplifications, extensions, and lower bounds. In *Theory of Cryptography*. Martin Hirt and Adam Smith, (Eds.) Springer Berlin Heidelberg, Berlin, Heidelberg, 635–658. ISBN: 978-3-662-53641-4.

[10] Adrian Rivera Cardoso and Ryan Rogers. 2022. Differentially private histograms under continual observation: streaming selection into the unknown. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event* (Proceedings of Machine Learning Research). Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, (Eds.) Vol. 151. PMLR, 2397–2419. https://proceedings.mlr.press/v151/rivera-cardoso22a.html.

[11] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and Continual Release of Statistics. *ACM Transactions on Information and System Security*, 14, 3, (Nov. 2011), 26:1–26:24. doi:10.1145/2043621.2043626.

[12] Christina Chandra, Martina Morris, Connor Van Meter, Steven M. Goodreau, Travis Sanchez, Patrick Janulis, Michelle Birkett, and Samuel M. Jenness. 2022. Comparing sexual network mean active degree measurement metrics among men who have sex with men. *medRxiv*. eprint: https://www.medrxiv.org/content/early/2022/05/16/2022.02.11.22270855.full.pdf. doi:10.1101/2022.02.11.22270855.

[13] Christopher A. Choquette-Choo, Arun Ganesh, Ryan McKenna, H. Brendan McMahan, John Rush, Abhradeep Guha Thakurta, and Zheng Xu. 2023. (amplified) banded matrix factorization: A unified approach to private training. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, (Eds.) http://papers.nips.cc/paper%5C_files/paper/2023/hash/ecc28b4ce9b39f5f23c3efb03e25b7bf-Abstract-Conference.html.

[14] Christopher A. Choquette-Choo, H. Brendan McMahan, J. Keith Rush, and Abhradeep Guha Thakurta. 2023. Multi-epoch matrix factorization mechanisms for private machine learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA* (Proceedings of Machine Learning Research). Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, (Eds.) Vol. 202. PMLR, 5924–5963. https://proceedings.mlr.press/v202/choquette-choo23a.html.

[15] Rachel Cummings, Alessandro Epasto, Jieming Mao, Tamalika Mukherjee, Tingting Ou, and Peilin Zhong. 2025. Differentially private space-efficient algorithms for counting distinct elements in the turnstile model. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025* (Proceedings of Machine Learning Research). Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, (Eds.) Vol. 267. PMLR / OpenReview.net. https://proceedings.mlr.press/v267/cummings25a.html.

[16] Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. 2014. On estimating the average degree. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*. Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel, (Eds.) ACM, 795–806. doi:10.1145/2566486.2568019.

[17] Sergey Denisov, H. Brendan McMahan, John Rush, Adam D. Smith, and Abhradeep Guha Thakurta. 2022. Improved differential privacy for SGD via optimal private linear operators on adaptive streams. In *NeurIPS*. http://papers.nips.cc/paper%5C_files/paper/2022/hash/271ec4d1a9ff5e6b81a6e21d38b1ba96-Abstract-Conference.html.

[18] Krishnamurthy Dvijotham, H. Brendan McMahan, Krishna Pillutla, Thomas Steinke, and Abhradeep Thakurta. 2024. Efficient and near-optimal noise generation for streaming differential privacy. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2306–2317.

[19] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing* (STOC '10). Association for Computing Machinery, New York, NY, USA, (June 2010), 715–724. ISBN: 978-1-4503-0050-6. doi:10.1145/1806689.1806787.

[20] Jean-Pierre Eckmann and Elisha Moses. 2002. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the National Academy of Sciences*, 99, 9, 5825–5829. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.032093399. doi:10.1073/pnas.032093399.

[21] Alessandro Epasto, Quanquan C. Liu, Tamalika Mukherjee, and Felix Zhou. 2025. Sublinear space graph algorithms in the continual release model. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2025, Berkeley, CA, USA, August 11-13, 2025* (LIPIcs). Alina Ene and Eshan Chattopadhyay, (Eds.) Vol. 353. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 40:1–40:27. doi:10.4230/LIPICS.APPROX/RANDOM.2025.40.

[22] Alessandro Epasto, Jieming Mao, Andres Muñoz Medina, Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong. 2023. Differentially private continual releases of streaming frequency moment estimations. In *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA* (LIPIcs). Yael Tauman Kalai, (Ed.) Vol. 251. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 48:1–48:24. doi:10.4230/LIPICS.ITCS.2023.48.

[23] Alessandro Epasto, Tamalika Mukherjee, and Peilin Zhong. 2023. Differentially private clustering in data streams. *CoRR*, abs/2307.07449. arXiv: 2307.07449. doi:10.48550/ARXIV.2307.07449.

[24] Hendrik Fichtenberger, Monika Henzinger, and Lara Ost. 2021. 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)* (LIPIcs). Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, (Eds.) Vol. 204. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 42:1–42:16. doi:10.4230/LIPICS.ESA.2021.42.

[25] Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. 2023. Constant matters: fine-grained error bound on differentially private continual observation. In *Proceedings of the 40th International Conference on Machine Learning* (Proceedings of Machine Learning Research). Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, (Eds.) Vol. 202. PMLR, (July 2023), 10072–10092. https://proceedings.mlr.press/v202/fichtenberger23a.html.

[26] Badih Ghazi, Ravi Kumar, Jelani Nelson, and Pasin Manurangsi. 2023. Private counting of distinct and k-occurring items in time windows. In *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA* (LIPIcs). Yael Tauman Kalai, (Ed.) Vol. 251. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 55:1–55:24. doi:10.4230/LIPICS.ITCS.2023.55.

[27] Google Research. 2025. Differentially private machine learning at scale with jax privacy. https://research.google/blog/differentially-private-machine-learning-at-scale-with-jax-privacy/. Google Research Blog. Accessed: 2025-12-05. (2025).

[28] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3, 1–2, (Sept. 2010), 1021–1032. doi:10.14778/1920841.1920970.

[29] Monika Henzinger, Nikita P. Kalinin, and Jalaj Upadhyay. 2025. Binned group algebra factorization for differentially private continual counting. *CoRR*, abs/2504.04398. arXiv: 2504.04398. doi:10.48550/ARXIV.2504.04398.

[30] Monika Henzinger, Nikita P. Kalinin, and Jalaj Upadhyay. 2025. Normalized square root: sharper matrix factorization bounds for differentially private continual counting. *CoRR*, abs/2509.14334. arXiv: 2509.14334. doi:10.48550/ARXIV.2509.14334.

[31] Monika Henzinger, A. R. Sricharan, and Teresa Anna Steiner. 2025. Differentially private continual release of histograms and related queries. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025* (Proceedings of Machine Learning Research). Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Mohammad Emtiyaz Khan, (Eds.) Vol. 258. PMLR, 1990–1998. https://proceedings.mlr.press/v258/henzinger25a.html.

[32] Monika Henzinger, A. R. Sricharan, and Teresa Anna Steiner. 2024. Private counting of distinct elements in the turnstile model and extensions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, London School of Economics, London, UK, August 28-30, 2024* (LIPIcs). Amit Kumar and Noga Ron-Zewi, (Eds.) Vol. 317. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 40:1–40:21. doi:10.4230/LIPICS.APPROX/RANDOM.2024.40.

[33] Monika Henzinger and Jalaj Upadhyay. 2025. Improved differentially private continual observation using group algebra. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*. Yossi Azar and Debmalya Panigrahi, (Eds.) SIAM, 2951–2970. doi:10.1137/1.9781611978322.95.

[34] Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. 2024. A unifying framework for differentially private sums under continual observation. In *Proceedings of Symposium on Discrete Algorithms (SODA)*. SIAM, 995–1018.

[35] Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. 2023. Almost tight error bounds on differentially private continual counting. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*. Nikhil Bansal and Viswanath Nagarajan, (Eds.) SIAM, 5003–5039. doi:10.1137/1.9781611977554.CH183.

[36] James Honaker. 2015. Efficient use of differentially private binary trees. *Theory and Practice of Differential Privacy (TPDP 2015), London, UK*, 2, 26–27.

[37] Ben Jacobsen and Kassem Fawaz. 2025. Private continual counting of unbounded streams. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=G8WnkCYEZ6.

[38] Palak Jain, Iden Kalemaj, Sofya Raskhodnikova, Satchit Sivakumar, and Adam D. Smith. 2023. Counting distinct elements in the turnstile model with differential privacy under continual observation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, (Eds.) http://papers.nips.cc/paper%5C_files/paper/2023/hash/0ef1afa0daa888d695dcd5e9513bafa3-Abstract-Conference.html.

[39] Palak Jain, Sofya Raskhodnikova, Satchit Sivakumar, and Adam D. Smith. 2023. The price of differential privacy under continual observation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA* (Proceedings of Machine Learning Research). Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, (Eds.) Vol. 202. PMLR, 14654–14678. https://proceedings.mlr.press/v202/jain23b.html.

[40] Palak Jain, Adam D. Smith, and Connor Wagaman. 2024. Time-aware projections: truly node-private graph statistics under continual observation. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*. IEEE, 127–145. doi:10.1109/SP54263.2024.00196.

[41] Peter Kairouz, H. Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. 2021. Practical and Private (Deep) Learning Without Sampling or Shuffling. en. In *Proceedings of the 38th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, (July 2021), 5213–5225. Retrieved Jan. 24, 2023 from https://proceedings.mlr.press/v139/kairouz21b.html.

[42] Nikita Kalinin and Christoph H. Lampert. 2024. Banded square root matrix factorization for differentially private model training. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, (Eds.) http://papers.nips.cc/paper%5C_files/paper/2024/hash/1f7b3b0dd7710af02aac0db5be4cfc8d-Abstract-Conference.html.

[43] Nikita P. Kalinin, Ryan McKenna, Jalaj Upadhyay, and Christoph H. Lampert. 2025. Back to square roots: an optimal bound on the matrix factorization error for multi-epoch differentially private SGD. *CoRR*, abs/2505.12128. arXiv: 2505.12128. doi:10.48550/ARXIV.2505.12128.

[44] Max Dupré la Tour, Monika Henzinger, and David Saulpic. 2024. Making old things new: A unified algorithm for differentially private clustering. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net. https://openreview.net/forum?id=3ajK5xplDL.

[45] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. 2015. The matrix mechanism: optimizing linear counting queries under differential privacy. en. *The VLDB Journal*, 24, 6, (Dec. 2015), 757–781. doi:10.1007/s00778-015-0398-x.

[46] Jiří Matoušek, Aleksandar Nikolov, and Kunal Talwar. 2020. Factorization norms and hereditary discrepancy. *International Mathematics Research Notices*, 2020, 3, 751–780.

[47] H. Brendan McMahan, Zheng Xu, and Yanxiang Zhang. 2024. A hassle-free algorithm for strong differential privacy in federated learning systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - Industry Track, Miami, Florida, USA, November 12-16, 2024*. Franck Dernoncourt, Daniel Preotiuc-Pietro,

and Anastasia Shimorina, (Eds.) Association for Computational Linguistics, 842–865. doi:10.18653/V1/2024.EMNLP-INDUSTRY.64.

[48] Krishna Pillutla, Jalaj Upadhyay, Christopher A. Choquette-Choo, Krishnamurthy Dvijotham, Arun Ganesh, Monika Henzinger, Jonathan Katz, Ryan McKenna, H. Brendan McMahan, Keith Rush, Thomas Steinke, and Abhradeep Thakurta. 2025. Correlated noise mechanisms for differentially private learning. *CoRR*, abs/2506.08201. arXiv: 2506.08201. doi:10.48550/ARXIV.2506.08201.

[49] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2013. Understanding hierarchical methods for differentially private histograms. *Proc. VLDB Endow.*, 6, 14, (Sept. 2013), 1954–1965. doi:10.14778/2556549.2556576.

[50] Sofya Raskhodnikova and Teresa Anna Steiner. 2025. Fully dynamic algorithms for graph databases with edge differential privacy. *Proc. ACM Manag. Data*, 3, 2, 99:1–99:28. doi:10.1145/3725236.

[51] Shuang Song, Susan Little, Sanjay Mehta, Staal Amund Vinterbo, and Kamalika Chaudhuri. 2018. Differentially private continual release of graph statistics. *CoRR*, abs/1809.02575. http://arxiv.org/abs/1809.02575 arXiv: 1809.02575.

[52] Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher A. Choquette-Choo, Peter Kairouz, H. Brendan McMahan, Jesse Rosenstock, and Yuanbo Zhang. 2023. Federated learning of gboard language models with differential privacy. In *Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023, Toronto, Canada, July 9-14, 2023*. Sunayana Sitaram, Beata Beigman Klebanov, and Jason D. Williams, (Eds.) Association for Computational Linguistics, 629–639. doi:10.18653/V1/2023.ACL-INDUSTRY.60.

A Additional Related Work

Differential Privacy under Continual Observation. As described in the introduction, the seminal works of Dwork et al. [19] and Chan et al. [11] defined the continual observation model of differential privacy where inputs are received as a stream and the goal is to track a statistic over time while maintaining privacy. The version of the model they considered involves a single element arriving per time step and a single output being released per time step. We consider a more general model where many elements can arrive/updates can occur per time step, with a single output still released over each time step. This more general model has been studied in papers on private continual observation for graph problems [51, 24, 40].

Continual Counting and the Binary Tree Mechanism. Continual counting (or summation of bits) is a fundamental primitive used in many data analysis tasks. The papers defining the continual observation model ([11] and [19]) also gave the binary tree mechanism, a mechanism for continual counting with error scaling polylogarithmically in the time horizon T.

Mechanisms for continual counting are the main building block used to give mechanisms for many tasks in the continual observation literature, for example, continual observation of histograms [10, 39, 31], state-of-the-art mechanisms for differentially private machine learning [41, 17, 14, 13, 52, 48], various frequency statistics on a stream [8, 26, 38, 22, 15], continual observation of various graph statistics [51, 24, 40, 50, 21], and continual observation for clustering problems [23, 44]. Hence, recent works that we will discuss next attempt to give mechanisms for continual counting that improve upon the constants of the binary tree mechanism (though they achieve the same asymptotic error). Most of these mechanisms are factorization mechanisms that exploit clever factorizations to achieve improved error bounds for private continual counting. Note that the neighboring notion studied in all these works is that adjacent streams differ in a single bit at one time-step.

Factorization Mechanisms for Continual Counting. Inspired by applications to cardinality estimation, we identify new neighboring relations under which we study private continual counting. Our work is effectively a study of the factorization mechanism [45] applied to the lower-triangular matrix of all-ones, $A \in \mathbb{R}^{T\times T}$ under these new neighboring notions. There is an extensive literature studying optimal matrix factorizations for private continual counting under the regular neighboring notion [17, 2, 25, 35, 18, 3, 33, 29, 5, 37]. We now describe these in more detail.

Tree-based factorizations [19, 11, 2, 4] were the first class of factorizations to achieve $O(\log T)$ error, and do so efficiently with $O(\log T)$ space and $O(T)$ time. Their key structure is sparsity, which allows for not only efficient computation, but also for a low ℓ_2 and ℓ_1 sensitivity, making them also state-of-the-art for the separate problem of ε-differentially private continual counting. Techniques of "denoising" tree-based mechanisms have also been proposed, where the left factor is made denser for a constant factor improvement at the cost of efficiency [36, 28, 41].

More general dense factorizations have subsequently been proposed [17, 25, 35, 33, 30], and give improved leading constants of the error, often at a cost in efficiency. We especially highlight the *square-root factorization* $L = R = \sqrt{A}$, where $\sqrt{A}$ is Toeplitz and lower-triangular with entries $\sqrt{A}[i,j] = \binom{2(i-j)}{i-j}/4^{i-j}$ for $i \geq j$. It attains an error of $\frac{\ln(T)}{\pi\sqrt{2\rho}} \pm O(1)$, which is optimal up to the additive constant for both of our error notions [25, 35, 46]. Subsequent work showed how to approximate this error up to the additive constant with a lower-triangular Toeplitz factorization in $O(\log^2 T)$ space and $O(T \log^2 T)$ time using rational-function approximation [18].

Factorization Mechanisms under 'Non-Standard' Neighboring Relations. We finally note that our work is not the first to study factorization mechanisms for releasing prefixes under

continual observation for a "non-standard" neighboring relation. Motivated by allowing a user to contribute multiple data points in a machine learning pipeline [41], the notion of *b-min separated k-repeated participation* was introduced [14, 13]. Here any training example can appear at most k times, each example separated by at least b time steps. For the factorization mechanism design problem, this translates to a sensitivity vector set $\mathcal{S}_{(k,b)} \subseteq [-1,1]^T$ where vectors are k-sparse and with non-zero entries having a spacing of at least b. As our k-sparse alternating sensitivity vector set $(S_{1,k})$ makes up a strict subset of $\mathcal{S}_{(k,1)}$, any upper bound on the error for a factorization mechanism under $\mathcal{S}_{(k,1)}$-participation extends to our setup. Crucially, as there is no analogous structure on $\mathcal{S}_{(k,1)}$ that enforces bounded prefix sums, these upper bounds are asymptotically worse than what we show. In particular, factorizations based on a banded (inverse) square-root factorization [42, 43] that are asymptotically optimal for this problem achieve a bound of $O(\sqrt{kT})$ [43, Theorem 2]. In contrast, we achieve a logarithmic dependence on T in our setting.

B Comparison Between Our Setting(s) and Related Ones

Single-Update vs. Multiple-Update Model. Most prior work on continual observation for fully dynamic streams (e.g., [38, 15]) considers the setting in which there is at most one update per time step (the *single-update model*). We instead study the more general setting that allows arbitrarily many updates per time step (the *multiple-update model*). This model has also been examined in the insertion-only graph setting [40]. We highlight several nuanced differences between the two settings:

- **Practicality.** In real-world streaming systems (e.g., login events for a streaming service), many updates typically occur between successive computations of a statistic. Thus, the multiple-update model provides a more realistic abstraction of deployed systems.
- **Micro-time step reduction.** One might attempt to reduce a problem in the multiple-update model to the single-update model by transforming a stream $\vec{\mathbf{x}}$ into a stream $\vec{\mathbf{y}}$ that introduces many "micro" time steps, each containing exactly one update. In effect, a single time step in $\vec{\mathbf{x}}$ is expanded into multiple micro-steps in $\vec{\mathbf{y}}$. However, this transformation increases the time horizon T to the total number of updates. Since accuracy in continual observation typically degrades with T, directly applying single-update results would lead to substantially worse guarantees. As we show, this loss in accuracy is not inherent for the problems we study.
- **Algorithmic limitations.** While some algorithmic strategies extend unchanged to the multiple-update model, others do not. For example, the periodic recomputation technique [11, 39] recomputes and adds noise to a low-sensitivity function at fixed intervals, releasing the statistic in the clear between recomputations. This approach crucially relies on the single-update assumption: because the statistic changes by only a small amount per time step, previously released estimates remain accurate. In the multiple-update model, a statistic may change substantially within a single time step, invalidating this reasoning. Concretely, it remains unclear whether the known $O(T^{1/3})$ worst-case upper bound for CountDistinct in the single-update model extends to the multiple-update setting; the best known upper bound in our model is $O(\sqrt{T})$, obtained by recomputing at every time step and incurring composition costs.

Neighboring Relations. We note that prior works for the problems we consider [38, 15, 50] define a slightly different item-neighboring relation. Formally, they say that two streams $\vec{\mathbf{x}}$ and $\vec{\mathbf{x}}' \in \mathcal{U}_{\pm}^T$ are neighboring if there exists at most one element $u \in \mathcal{U}$ such that replacing **a subset of** updates corresponding to u (i.e. $-u$ or $+u$) with $\perp$ in one of the streams results in the other. On the other hand, in our neighboring notion, two streams $\vec{\mathbf{x}}$ and $\vec{\mathbf{x}}' \in \mathcal{U}_{\pm}^T$ are neighboring if there exists at most one element $u \in \mathcal{U}$ such that replacing **all** updates corresponding to u (i.e. $-u$ or $+u$)

with $\perp$ in one of the streams results in the other. We find our notion semantically easier to reason about, as it captures the guarantee that very little is revealed about the updates corresponding to any single item in the stream: replacing all such updates with $\perp$ does not significantly affect the distribution of outputs.

We note that our results extend to their neighboring notion by simple group privacy, since neighbors in their notion are at most distance two from each other in our neighboring notion. Their results are also easily adaptable to our neighboring notion.

We also note that while we focus on item-level privacy where neighboring streams may differ in all updates corresponding to a single item, other works [38, 15, 50] additionally study these problems under event-level privacy, in which neighboring streams differ in a single update.

User-Level vs. Item-Level Privacy. The setting we consider—where two streams are neighboring if they differ in all updates corresponding to a single item—is closely related to *user-level* privacy [19] (and sometimes used synonymously). When items correspond directly to user identifiers, the two notions coincide: modifying all updates associated with a single item is equivalent to modifying the data of a single user.

However, in many applications this identification is not exact. For example, in social network streams where updates correspond to edge insertions and deletions, a single user (vertex) may participate in many edges and therefore influence many distinct items. In such settings, users may hold data corresponding to multiple items. To avoid ambiguity, we adopt the term "item-level privacy", following the terminology introduced in [38] and subsequently used in related work [15, 50].

Received December 2025; accepted February 2026