Improved Lower Bounds for Privacy under Continual Release

BARDIYA ARYANFARD, Institute of Science and Technology Austria (ISTA), Austria

MONIKA HENZINGER, Institute of Science and Technology Austria (ISTA), Austria

DAVID SAULPIC, CNRS & Université Paris Cité IRIF, France

A. R. SRICHARAN, Faculty of Computer Science, UniVie Doctoral School of Computer Science DoCS, University of Vienna, Austria

We study the problem of continually releasing statistics of an evolving dataset under differential privacy. In the event-level setting, we show the first *polynomial* lower bounds on the additive error for *insertions-only* graph problems such as maximum matching, degree histogram and k-core number computation. These results represent an exponential improvement on the polylogarithmic lower bounds of Fichtenberger, Henzinger, and Ost [35] for the former two problems, and are the first lower bounds in the continual release setting for the latter problem. Our results run counter to the intuition that the difference between insertions-only vs fully dynamic updates (ie with insertions and deletions) causes the gap between polylogarithmic and polynomial additive error. Indeed, we show that for estimating the size of the maximum matching or k-core number of a vertex, allowing small multiplicative approximations is what brings the additive error down to polylogarithmic. We complement these results with improved upper bounds on the additive error when no multiplicative approximation is allowed.

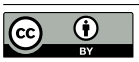
Beyond graph problems, our techniques also show that polynomial additive error is unavoidable for the Simultaneous Norm Estimation problem in the insertions-only setting. When multiplicative approximations are allowed, we circumvent this lower bound by giving the first continual mechanism with polylogarithmic additive error under $(1 + \zeta)$ multiplicative approximations, for any $\zeta > 0$, for estimating *all* monotone symmetric norms simultaneously.

In the item-level setting, we show *polynomial* lower bounds on the product of the multiplicative and the additive error of continual mechanisms for a large range of graph problems. To the best of our knowledge, these are the first lower bounds shown for any differentially private mechanism under continual release with multiplicative error. To obtain these results, we prove a new lower bound on the product of multiplicative and additive error for the 1-Way-Marginals problem, and give reductions from 1-Way-Marginals to our desired graph problems. This generalizes the prior results of Hardt and Talwar [39] and Bun, Ullmann, and Vadhan [13], who gave lower bounds on the additive error for the special case of mechanisms with no multiplicative error.

CCS Concepts: • **Security and privacy**; • **Theory of computation → Dynamic graph algorithms**;

Additional Key Words and Phrases: Differential Privacy, Continual Observation, Lower Bounds

Authors' Contact Information: Bardiya Aryanfard, bardiya.aryanfard@ist.ac.at, Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria; Monika Henzinger, monika.henzinger@ist.ac.at, Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria; David Saulpic, david.saulpic@irif.fr, CNRS & Université Paris Cité IRIF, Paris, France; A. R. Sricharan, sricharan.arunapuram@univie.ac.at, Faculty of Computer Science, UniVie Doctoral School of Computer Science DoCS, University of Vienna, Vienna, Austria.



ACM 2836-6573/2026/5-ART107
https://doi.org/10.1145/3801903

ACM Reference Format:

Bardiya Aryanfard, Monika Henzinger, David Saulpic, and A. R. Sricharan. 2026. Improved Lower Bounds for Privacy under Continual Release. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 107 (May 2026), 27 pages. https://doi.org/10.1145/3801903

1 Introduction

Differential privacy [24] has emerged as the standard for privacy-preserving data analysis. Research on releasing updated statistics with each update to an evolving dataset under differential privacy, called the *continual release (or observation) setting*, was initiated by the seminal work by Dwork et al. [26] and Chan et al. [15] on computing prefix sums of an integer-valued stream. It has since seen usage in a wide variety of topics, including subroutines for first-order methods in machine learning [20, 36, 58], dynamic maintenance of graph properties [21, 35, 47, 63], histograms [43, 45], and estimating norms of frequency vectors of data streams [14, 32] (see [41] for a survey).

The general continual release model is commonly defined as follows: Given an initially empty data set, a stream of T update operations arrive, one per time step. After each arrival a new answer has to be output, leading to the disclosure of T outputs. The goal is to have a small additive error; ideally it should be at most polylogarithmic in the problem parameters, which is typically the length T of the stream and, e.g. in the case of graphs, the number of vertices n. Usually the *incremental* setting (where each operation is an insertion) and the *fully dynamic* setting (where an operation is an insertion or a deletion) have been studied, since the deletions-only setting is often symmetric to the insertions-only setting.

Initial edge-private graph mechanisms [35, 63] focused mostly on the *incremental* setting. Fichtenberger, Henzinger and Ost [35] gave mechanisms for various graph problems with polylogarithmic additive error, albeit with small $(1 + \zeta)$ multiplicative approximations, and complemented these with logarithmic lower bounds on the additive error of any mechanisms in the *exact setting* (i.e., mechanisms with no multiplicative error). The situation in the *fully dynamic* setting, on the other hand, was more dire, where no non-trivial mechanisms were known for most problems of interest.

Towards explaining the lack of such mechanisms in the fully dynamic setting, a recent line of results [31, 59] shows that polynomial additive error is unavoidable for mechanisms in the exact setting for graph problems such as maximum matching and degree histogram under both insertions and deletions. In the non-private setting, going from the incremental to the fully dynamic setting usually leads to a decrease in performance (as measured by asymptotic running time). Thus, it is natural to conjecture that the decrease in performance in the private setting (measured by additive error) is due to the change in the set of permissible operations. This leads to the following question:

Question 1

Does the transition from the incremental to the fully dynamic setting cause the increase from polylogarithmic to polynomial additive error for differentially private mechanisms?

Prior work [31, 59] do not answer this question: The mechanisms with polylogarithmic additive error in the *incremental* setting for many of these problems have a *small multiplicative error*, while their polynomial fully dynamic lower bounds hold only for mechanisms in the exact setting.

Surprisingly, we are able to refute the intuition that the inherent hardness of obtaining mechanisms with polylogarithmic additive error is caused by the fully dynamic setting for many of these

problems: We give polynomial lower bounds for them even for the incremental version. We study both graph problems – such as matching or k-core – but also use our techniques for a standard data streaming problem, that of simultaneous norm estimation. This answer motivates us to refine our understanding of incremental mechanisms in the exact setting, for which we present improved algorithms – reducing the gap between the upper and lower bounds.

Secondly, note that in the incremental model, any object appears only once in the stream – namely when it is added. However, for the fully-dynamic setting, objects can appear multiple times, and this can lead to two different notions of *neighboring inputs* and, thus, of differential privacy, namely *item-level* neighboring and *event-level* neighboring. In graphs, for the *item-level* setting of privacy, privacy is guaranteed against *all* updates corresponding to the same edge – which can be inserted and deleted multiple times over the stream history. This is opposed to the *event-level* setting, where privacy is guaranteed against a single update operation, i.e., one update of one edge.

Question 1 applies to the event-level setting. However, depending on the application, the item-level setting with its stronger privacy guarantees is more relevant. Therefore we also investigate the accuracy guarantees that are possible for differentially private mechanisms in the item-level fully dynamic setting. Previous item-level lower bounds for graphs by Raskhodnikova and Steiner [59] were slightly higher than the corresponding event-level bounds, but were still only for the *exact* setting, i.e., without multiplicative error. Our dichotomy between exact mechanisms and mechanisms with multiplicative error for incremental results raises the question whether there is a similar dichotomy in the item-level fully dynamic setting. Said differently, is it possible that the item-level lower bounds of [59] can be circumvented by giving mechanisms with small multiplicative approximations and polylogarithmic additive error? Or are there polynomial lower bounds even with small multiplicative error? This leads us to the following question:

Question 2

Does there exist a dichotomy between the additive error of exact mechanisms and mechanisms with multiplicative error for the item-level setting as in the event-level setting?

We answer this question negatively by giving suitable polynomial lower bounds.

To conclude, we make the following contributions: Firstly, we prove polynomial lower bounds on a variety of graph problems and for symmetric norms estimation in the event-level incremental setting. This runs counter to the intuition of these problems being difficult only in the fully-dynamic setting. Then, we complement those lower bounds with better upper bounds for same problems. Finally, we present a lower bound on the product of additive and multiplicative error for 1-Way-Marginals and then use it to prove lower bounds on the product of additive and multiplicative error for a variety of graph problems in the item-level setting, which answers Question 2 above.

1.1 Our Results

Question 1. We refute the common intuition that the hardness of obtaining mechanisms with polylogarithmic additive error is caused by the ability to remove elements from the stream, by giving polynomial lower bounds for the incremental version of many of these problems. Our results show that for problems such as maximum matching and estimating the k-core number of a vertex, the hardness arises not from allowing deletions, but rather from the choice of not allowing small multiplicative approximations. Specifically, we show the following result:

Table 1. Results on private event-level problems. A tuple η, $O(\alpha)$ says there exists an η-multiplicative approximation mechanism with $O(\alpha)$ additive error, while η, $\Omega(\alpha)$ shows a lower bound of $\Omega(\alpha)$ on any η-approximation. The superscript min on a multivariate asymptotic notation means that the notation holds for the min of all operands, e.g., $O^{\min}(A, B) = O(\min\{A, B\})$. Results that only hold for pure differential privacy are subscripted with ε. We hide the dependence on ε and $\log(1/\delta)$ for simplicity. The lower bound on SNE in Theorem 3 holds even for a subset of monotone symmetric norms, namely TopK norms.

Problem	Incremental			Fully Dynamic		
	Mult. Err.	Add. Err.	Reference	Mult. Err.	Add. Err.	Reference
Matching	1	$\Omega_\varepsilon(\log T)$	[35]			
	1	$\Omega^{\min}(\sqrt[4]{T}, \sqrt[4]{n})$	Section 4	1	$\Omega^{\min}(\sqrt[4]{T}, \sqrt{n})$	[31, 59]
	1	$\widetilde{O}^{\min}(\sqrt[3]{T}, \sqrt[3]{n})$	Lemma 3	1	$O^{\min}(\sqrt[3]{T}, n)$	[59]
	$1+\zeta$	$O_\varepsilon(\log^2 n/\zeta)$	[35]			
KCore(v)	1	$\Omega^{\min}(\sqrt[8]{T}, \sqrt[4]{n})$	Section 4			
	1	$\widetilde{O}^{\min}(\sqrt[6]{T}, \sqrt[3]{n})$	Lemma 3			
	$1+\zeta$	$O_\varepsilon(\log^2 n/\zeta)$	[35]			
	$2+\zeta$	$O_\varepsilon(\log^3 n/\zeta^2)$	[31]			
DegHist	1	$\Omega_\varepsilon(\log T)$	[35]			
	1	$\Omega^{\min}(\sqrt[4]{T}, \sqrt{n})$	Section 4	1	$\Omega^{\min}(\sqrt[4]{T}, \sqrt{n})$	[59]
	1	$\widetilde{O}^{\min}(\sqrt[3]{T}, \sqrt{n})$	Lemma 4	1	$O^{\min}(\sqrt[3]{T}, n)$	[59]
SNE	1	$\Omega^{\min}(\sqrt[4]{T}, \sqrt{n})$	Theorem 3			
	$1+\zeta$	$O_\varepsilon(\log^6(nT)/\zeta^5)$	Theorem 4			

Theorem 1. *Let $\varepsilon \in [0, 1]$ and $\delta \in [0, 1/3]$. Any (ε, δ)-dp incremental mechanism in the exact setting for maintaining the size of the maximum matching, degree histogram, or core number of a vertex*[1] *must have additive error $\alpha = \Omega(\min\{T^a, n^b\})$, for some problem-specific constants $a, b \in [1/8, 1/2]$ which are given in Table 1.*

Our polynomial lower bounds for the real-valued graph problems are tight in the sense that relaxing to either the static setting or to small multiplicative approximations admit private mechanisms with polylogarithmic additive error. Our bounds exponentially improve the previous lower bounds of Fichtenberger, Henzinger and Ost [35] for maximum matching and degree histogram, and are the first lower bounds for k-core estimation under continual release.

We also give improved upper bounds for these problem, significantly reducing the gap between lower and upper bounds in the item-level setting. More specifically, when T was much larger than n, the best mechanisms in the exact setting given by Raskhodnikova and Steiner [59] in the fully dynamic case (for the problems whose lower bound reduction was from inner product queries) was the trivial mechanism that returns 0 as the answer at all time steps. This led to a $\sqrt{n}$ gap between the lower and upper bounds for the problems in Table 1. We present new *incremental* upper bounds that match our lower bounds up to a $\sqrt[12]{n}$ factor, drastically reducing the gap.

[1]These problems are formally defined in Section 2.

Theorem 2. *There exist (ε, δ)-dp incremental mechanisms in the exact setting for maintaining the size of the maximum matching, degree histogram, or core number of a vertex with additive error*[2] *$\alpha = \widetilde{O}_{\varepsilon,\delta}(\min\{T^c, n^d\})$ for problem-specific constants $c, d \in [1/6, 1/2]$ which are given in Table 1.*

Norm estimation problem. We show that our lower bound techniques also extend from graph mechanisms to a standard problem for data streams, namely the incremental *norm estimation problem*. This problem requires maintaining a data structure whose output estimates various norms of the frequency vector under continual insertions of elements. It is especially important in the private setting to estimate multiple norms simultaneously, since querying for each norm separately using composition theorems requires privacy budget proportional to the square root of the number of norms queried, which is prohibitive as the number of queries grow. This problem has been studied extensively under differential privacy in the streaming setting [7, 11, 52, 62, 66], and there exist mechanisms with polylogarithmic additive error at the cost of a small $(1 + \zeta)$ multiplicative error. Since the goal of these mechanisms is to minimize space usage in addition to the additive error, most works use a variant of a bucketing technique which inherently loses some multiplicative error. Towards discovering the limits of the problem in the continual setting even without space constraints, we show that such multiplicative approximations are necessary for simultaneous norm estimation problems (regardless of space usage) by showing polynomial lower bounds on the additive error of any continual mechanism that solves a special case, namely estimating all the Top-k norms (where the Top-k norm is the sum of the k largest frequencies).

Theorem 3. *Let $\varepsilon \in [0, 1]$ and $\delta \in [0, 1/3]$. Any incremental (ε, δ)-dp mechanism in the exact setting for TopK that runs with n elements for T timesteps must have additive error $\alpha = \Omega(\min\{\sqrt[4]{T}, \sqrt{n}\})$.*

This is the first lower-bound for this problem, even in the fully-dynamic setting. We complement this result by showing that the problem admits polylogarithmic additive error both in the static setting and when allowing small multiplicative error. The static problem can be reduced to computing prefix sums of the sorted frequency vector. For the incremental setting, *we show a strong upper bound on estimating all monotone symmetric norms* simultaneously (SNE) with small multiplicative error and polylogarithmic additive error (See Definition 1 for a formal definition). As a special case, this problem consists of estimating all the ℓ_p-norms and the Top-k norms of the frequency vector simultaneously. As point of comparison, there is no known way of getting a mechanism for estimating *all* ℓ_p norms from a mechanism that just estimates one, since you cannot use composition theorems over all possible ℓ_p norms. For Top-k norms, using a composition theorem would have an accuracy loss that is at least $\sqrt{n}$, while our result has additive error $\mathrm{polylog}(n)$.

Theorem 4. *Let $\varepsilon > 0$ and $\zeta \in (0, 1/2]$. There exists an ε-dp $(1 + \zeta)$-approximate mechanism for IncSNE with additive error $\widetilde{O}\left(\frac{1}{\varepsilon^2\zeta^5} \cdot \log^6(nT)\right) \cdot L(e_1)$ for any monotone symmetric norm L, where $L(e_1)$ is the norm of the first standard basis vector.*[3]

Question 2. As our final result, we show that the item-level lower bounds of [59] cannot be circumvented by allowing multiplicative error: we give a polynomial lower bound on the product of the additive and multiplicative errors. This implies that the item-level fully dynamic setting is too strong to admit mechanisms with small additive error, even with polylog multiplicative error.

[2]Here $O_{\varepsilon,\delta}$ hides $\mathrm{poly}(1/\varepsilon, \log(1/\delta))$ factors, and $\widetilde{O}(X) = O(X\,\mathrm{polylog}(X))$.

[3]$L(e_1)$ is allowed to not be $O(1)$ here, since this holds for *any* monotone symmetric norm L, not only a fixed one.

Theorem 5. *Let $\varepsilon, \delta \in [0,1]$. Any (ε,δ)-dp mechanism in the item-level fully dynamic setting with multiplicative approximation η and additive error α for maintaining the size of the maximum matching, subgraph counts, degree histogram, mincut, s-t mincut, or core number of a vertex on n-node graphs for $T \geq \Omega(1/\delta)$ time steps must have*

$$\eta \cdot \alpha = \begin{cases} \Omega(\min\{\sqrt{T}, n\}) & \text{when } \delta = 0 \\ \widetilde{\Omega}(\min\{\sqrt[3]{T}, n\}) & \text{when } 0 < \delta < o(1/T) \end{cases}$$

for edge count, the lower bounds are $\Omega(\min\{\sqrt{T}, n^2\})$ for $\delta = 0$ and $\Omega(\min\{\sqrt[3]{T}, n^2\})$ for $\delta \in (0, o(1/T))$.

Even in the exact setting, these constitute the first lower bounds for mincut, s-t-mincut and k-core.

Our results show that even relaxing to prohibitive $o(n/\text{polylog}(n))$-multiplicative approximations still does not result in polylogarithmic additive error, and any $O(\sqrt{n})$-approximate mechanism still needs to have $\Omega(\sqrt{n})$ additive error for long streams (with $T \gg n$). For all the fully dynamic graph problems discussed above, while the item-level setting offers high privacy guarantees, we show that no private mechanism can offer competitive accuracy guarantees. To achieve these bounds we extend the fingerprinting lower bound technique of [13, 39] to work for mechanisms for 1-Way-Marginals that have both a multiplicative *and* an additive error. 1-Way-Marginals has served as a central hard problem in differential privacy and reductions from this problem have been used repeatedly to show lower bounds [2, 44, 46, 59]. Thus, our strengthened lower bound is likely to be of independent interest. A different extension of mechanisms with only multiplicative error was given in [57]. We compare with their work in the related work section (Section 1.2).

Due to space constraints, we provide a high-level technical overview and focus on selected results, while presenting some results only in a very concise way. All missing details can be found in the full version of the paper [1].

1.2 Related Work

Other work on fingerprinting-based lower bounds. Peter, Tsfadia, and Ullman [57] use a variation of Bun et al. [13]'s method to show a hardness result for fingerprinting codes with additional constraints on the codewords. Namely, they consider the case that for $b \in \{0,1\}$, at least a $\frac{1-\alpha}{2}$ fraction of columns are b in all codewords. For the 1-Way-Marginals problem, they use a reduction from their constrained version of fingerprinting codes to show a lower bound on the additive error (where the additive error is measured by the ℓ_2 error), which is a function of the maximum pairwise Euclidean distance of the rows. They use the assumption that there is only a small fraction of columns with mixed entries (in their reduction using the constrained fingerprinting hardness result), to bound the maximum pairwise Euclidean distance of the rows of the matrix. The idea is that if at least d' out of d columns are uniform in all of their entries, then every pair of rows is different in at most $d - d'$ entries, so their Euclidean distance is at most $\sqrt{d - d'}$. Since we deal with problems whose accuracy is measured with the ℓ_∞ norm, and since their technique does not achieve different bounds for the ℓ_∞ error than that of Bun et al. [13], their methods are orthogonal to ours and cannot be used to show our multiplicative approximation results.

They apply their technique to show a lower bound for differentially private mechanisms for (k,z)-clustering for all $z \geq 1$ that allow both multiplicative and additive error. Specifically, they show a lower bound on the additive error which has inverse dependency on the square root of the multiplicative error (ignoring polylogarithmic factors). As they do not analyze graph problems and work with a different norm, their results are incomparable with ours.

Privacy for graph problems. Private release of graph statistics has been an avenue of fruitful results with the definition of edge differential privacy by Nissim et al. [55], including results on graph cuts [18, 30, 37, 38], effective resistances [51], All Pair Shortest Paths [9, 16, 33, 60], densest subgraph [22, 34, 54], triangle counting [40], and minimum spanning trees [56]. This area has also been explored in other settings such as local DP [53]. Song et al. [63] and Fichtenberger et al. [35] initiated the study of graphs in the dynamic setting, and Fichtenberger et al. gave mechanisms in the exact setting with polylogarithmic error for minimum spanning tree, number of high-degree nodes, and edge count in the incremental setting. For other problems with low sensitivity, they gave $(1+\zeta)$-approximation mechanisms based on the sparse vector technique with polylogarithmic additive error. Epasto et al. [31] and Raskhodnikova and Steiner [59] give the first polynomial lower bounds for fully dynamic mechanisms in the event-level setting, as mentioned in Table 1. [59] also give lower bounds in the item-level setting. Our results extend their framework in both settings.

Simultaneous Norm Estimation problem. Estimating norms of the frequency vector is a fundamental problem which is heavily investigated because of its applications in areas such as data management [3, 64]. This line of work includes a privacy angle [5, 48, 67], and has many practical applications, for example in estimating join sizes [50, 68]. Blasiok et al. [6] study the *non-private* monotone symmetric norm estimation (SNE) problem in the streaming model where the focus is on having low memory, and the mechanism only provides an output for the last time step. They give a mechanism with multiplicative error $(1+\zeta)$, and small memory that depends quadratically on a parameter they call *maximum modulus of concentration mmc*(L) ($O(\log n)$ for ℓ_p norm with $p \in [1,2]$, but $\sqrt{n/k}$ for Top-k). Braverman et al. [11] give a differentially private mechanism for this problem in the streaming model. Using sublinear memory (that depends on the *mmc* parameter), they can output at the end of the stream a vector which solves the SNE problem with multiplicative error $(1+\zeta)$, but their privacy analysis relies on the fact that the mechanism only output once, while our mechanism gives T outputs.

The private estimation of singular ℓ_p frequencies has received attention in the streaming model, which is the model that outputs only once. In this model prior work focuses on estimating a *single* ℓ_p norm with p given as input. Specifically results are known for the case of $p = 0, 1, 2$ [7, 12, 17, 27, 52, 61, 62], $p \in (0,1]$ [66], and $p \in [0,\infty)$ [8]. This differs from our setting which *simultaneously estimates multiple norms* under *continual observation.* There has been one prior work in the continual observation setting. For the special case of a *given* ℓ_p norm, Epasto et al. [32] presented a mechanism to privately estimate the norm, which works for any fixed p given to the mechanism at start. Our work extends this result by allowing *simultaneous* estimation of all monotone symmetric norms – covering all ℓ_p and Top-k norms. Note that our mechanism does not need to know which norms will be asked for at the start. It only needs to disclose the requested norm when the user wants to estimate it.

2 Preliminaries

Notation. We denote $\{1, 2, \ldots, h\}$ by $[h]$. All our problems have a universe $\mathcal{U} = [h] \cup \{\perp\}$ of elements including the empty element $\perp$, and all our streams have a fixed length $T \in \mathbb{N}$. The empty element has no influence on the value of the function that is computed on any set S of elements, so the value of the function on S, $S \setminus \{\perp\}$, and $S \cup \{\perp\}$ are identical. For frequency estimation problems, we use $n = h$ to denote the number of elements, and for graph problems, we use n to denote the number of nodes where each element $i \in \mathcal{U}$ corresponds to an edge in the graph.

Incremental and fully dynamic problems. A *fully-dynamic* stream, which allows both insertion and deletion of elements, is a sequence $\mathbf{u} = u^1, \ldots, u^T$ with $u^t \in \{+,-\} \times \mathcal{U}$. An *incremental* stream $\mathbf{u}$

is a fully dynamic stream with only insertions, i.e., $u^t \in \{+\} \times \mathcal{U}$ for all $t \in T$. An insertions-only problem is prefixed with Inc, while a fully-dynamic problem is prefixed with FD.

Frequency vector. The frequency vector $f^t \in \mathbb{N}^h$ at time t counts the number of times each item is present in the dataset after taking into account all insertions and deletions up to time t. Formally,

$$f^t = (f_1^t, \dots, f_h^t), \text{ and } \quad f_i^t = \sum_{t \in [T]} \mathbb{1}\left(u^t \in \{+,-\} \times \{i\}\right) \cdot \operatorname{sgn}(u^t),$$

where $\operatorname{sgn}(u^t) = 1$ if the first coordinate of u^t is $+$, and -1 otherwise.

We consider the most general setting where we allow the frequency vector to be non-binary and negative. Thus, items can be inserted to the stream even when they are already present, and also deleted when absent. Note that the $\perp$ element is not counted in the frequency vector, so having an update of $\perp$ corresponds to a time step with no changes for all the problems we consider.

Static problems. In the static version of the dynamic problems discussed, all the updates are provided upfront, and only one estimate of the value is required after all the updates have been performed.

Neighboring streams. Two streams are *event-level* neighboring if they are identical at all time steps except for one. Formally, $\mathbf{u}$ and $\mathbf{v}$ are event-level neighboring if there exists a $t^* \in [T]$ such that $u^t = v^t$ for all $t \neq t^*$. Two streams are *item-level* neighboring if the streams are identical in the occurrences of all elements except for those of a single element $x \in \mathcal{U}$, and one stream can be obtained from the other by replacing some subset of occurrences of x with $\perp$. Formally, $\mathbf{u}$ and $\mathbf{v}$ are item-level neighboring if there exists an element $x \in \mathcal{U}$ such that either

$$v^t \in \begin{cases} \{u^t\} & \text{for all } t \in [T] \text{ such that } u^t \notin \{+,-\} \times \{x\} \\ \{u^t\} \cup (\{+,-\} \times \perp) & \text{otherwise} \end{cases}$$

or the above restriction holds with the roles of u and v swapped.

Differential privacy. Let $\varepsilon, \delta \geq 0$. A randomized mechanism $\mathcal{A} : \mathcal{U}^T \to O$ is said to be (ε, δ)-differentially private (DP) if for all neighboring streams $\mathbf{u}$ and $\mathbf{v}$ and for all subsets of outputs $S \subseteq O$, we have $\Pr[\mathcal{A}(\mathbf{u}) \in S] \leq e^\varepsilon \cdot \Pr[\mathcal{A}(\mathbf{v}) \in S] + \delta$. If $\delta = 0$, then we say the mechanism is ε-DP.

Continual observation. An algorithm in the continual observation setting gets as input a stream $\mathbf{u}$. At each time step $t \in [T]$, it obtains as input u^t, and must output some $x^t \in \mathcal{X}$ only using $u^1, \dots, u^t$ before moving on to the next time step. In this setting, the output space O for defining privacy is the product of T many output spaces $\mathcal{X}$, one at each time step, namely $O = \mathcal{X}^T$.

Here, we mention two key properties of differential privacy (post-processing and composition) that are used in this paper.

Theorem 6 (Post-processing [29]). *Let $\mathcal{M}$ be an (ε, δ)-DP mechanism with range $\mathcal{R}$. For any arbitrary randomized mapping $f : \mathcal{R} \to \mathcal{R}'$, $f \circ \mathcal{M}$ is also (ε, δ)-DP.*

Now, we mention two composition theorems for differential privacy.

Theorem 7 (Basic Composition [29]). *Let $\mathcal{M}_1, \dots, \mathcal{M}_k$ be differentially private mechanism on the same domain such that $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-DP (for all $i \in [k]$). Then $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_k(x))$ is $(\sum_{i=1}^k \varepsilon_i, \sum_{i=1}^k \delta_i)$-DP.*

Theorem 8 (Advanced Composition [29]). *Given target privacy parameters $0 < \varepsilon' < 1$ and $\delta' > 0$, To ensure $(\varepsilon', k\delta + \delta')$ cumulative privacy loss over k mechanisms it suffices that each mechanism is (ε, δ)-DP, where $\varepsilon = \frac{\varepsilon'}{2\sqrt{2k\ln(1/\delta')}}$.*

Approximations. Since we work with randomized mechanisms whose accuracy guarantee holds with some probability, we say that a randomized mechanism $\mathcal{A}$ for a real-valued problem g has η multiplicative error and α additive error if the output of the mechanism satisfies, on all inputs x, with probability at least $2/3$, $\eta^{-1} \cdot g(x) - \alpha \leq \mathcal{A}(x) \leq \eta \cdot g(x) + \alpha$. When $\eta = 1$, we call this the *exact setting*, since there is no multiplicative error. For vector-valued functions, the output of the mechanism must satisfy on all inputs x, with probability at least $2/3$,

$$\|\mathcal{A}(x) - \eta^{-1} \cdot g(x)\|_\infty \leq \alpha \quad \text{and} \quad \|\mathcal{A}(x) - \eta \cdot g(x)\|_\infty \leq \alpha.$$

The output in the continual observation setting is a vector in the output space $O = \mathcal{X}^T$, and thus the accuracy guarantee holds with probability at least $2/3$ *over all time steps* as well.

TopK. Let $\mathcal{U}$ be a universe with n elements. The TopK norms problem is to output an n-dimensional vector containing the estimate of the norms $\|f\|_{\text{Top-}k} = \sum_{i=1}^{k} |f_{\sigma(i)}|$ of the frequency vector f for all $k \in \{1, \ldots, n\}$, where σ is the sorting permutation, i.e., $|f_{\sigma(1)}| \geq |f_{\sigma(2)}| \geq \cdots |f_{\sigma(n)}|$. This interpolates between the ℓ_1 and the ℓ_∞ norms, with Top-1 $= \ell_\infty$ and Top-n $= \ell_1$.

Monotone symmetric norms. A norm L is monotone if for all $f, f' \in \mathbb{R}^n$ with $|f_i| \leq |f'_i|$ for all i, it holds that $L(f) \leq L(f')$. It is symmetric if for all permutations σ of n elements, it holds that $L(f) = L(f_{\sigma(1)}, \ldots, f_{\sigma(n)})$. It is known [4] that a norm is monotone if and only if $L(f) = L(|f_1|, \ldots, |f_n|)$, which is called *gauge-invariance*.

Definition 1 (The SNE problem)**.** The Monotone Symmetric Norm Estimation (SNE) problem is to estimate the frequency vector f for all monotone symmetric norms simultaneously. This comes in two flavours. Given a frequency vector $f \in \mathbb{N}^n$ with $n = h$ elements, the *vector version* of SNE is to maintain a vector $\hat{f} \in \mathbb{R}^n$ where $\hat{f}$ is an approximation with η multiplicative error and α additive error if the following holds for all symmetric norms L simultaneously:

$$|L(\hat{f}) - \eta^{-1} \cdot L(f)| \leq \alpha \quad \text{and} \quad |\mathcal{L}(\hat{f}) - \eta \cdot L(f)| \leq \alpha.$$

The second version, called the *data structure version* of SNE, is to maintain a data structure $\mathfrak{d}$ that can be queried to estimate any particular monotone symmetric norm, where $\mathfrak{d} : \mathbb{N}^n \times \mathcal{L} \to \mathbb{R}^n$ is an approximation with η multiplicative error and α additive error if the following holds for all symmetric norms L simultaneously: $|\mathfrak{d}(f, L) - \eta^{-1} \cdot L(f)| \leq \alpha$ and $|\mathfrak{d}(f, L) - \eta \cdot L(f)| \leq \alpha$.

As earlier, privacy is defined with respect to all possible outputs of $\mathfrak{d}$ for any query. For the mechanisms we design, the data structure will maintain a finite set of frequency vectors and return one of them for any queried norm, so it suffices to argue privacy with respect to returning the set of output vectors since the choice of vector to return can be performed as post-processing.

Constructing a graph from a stream. Given a stream $\mathbf{u}$ where $\mathcal{U}$ is the set of edges of a graph, we construct the graph G^t at time t with all positive frequency edges, namely $e \in G^t$ if and only if $f_e^t > 0$, where f_e^t is the cumulative frequency of edge e in the first t updates of $\mathbf{u}$ as defined above. Thus event-level neighoring streams correspond to the standard edge-neighboring event-level setting, and item-level neighboring streams correspond to the standard edge-neighboring item-level setting, since one can replace every operation that moves f_e^t outside $\{0, 1\}$ by $\perp$.

Graph problems. A graph problem g is a vector-valued function $g : \mathcal{G} \to \mathbb{R}^k$ where $\mathcal{G}$ is the space of all undirected, unweighted, simple graphs. We consider the following problems on graphs, and the property to be returned by a mechanism for the problem. Some of the problems have designated special vertices (such as s and t for s-t-MinCut), which are assumed to be encoded in the graph representation (such as being the first two vertices in the representation).

- Matching: maximum number of vertex disjoint edges present,
- DegHist: degree histogram $= (d_1, \ldots, d_{n-1})$ where $d_i =$ number of vertices of degree i,
- KCore(v): core number of a particular vertex v, which is defined as

$$\max\{k \in [n] \mid v \in \text{a subgraph of } G \text{ with min degree} \geq k\},$$

- MinCut: minimum number of edge removals needed to disconnect the graph,
- s-t-MinCut: minimum number of edge removals needed to disconnect vertices s and t,
- $f_{\geq\tau}$: number of vertices with degree at least τ,
- EdgeCount: number of edges in the graph.

For our lower bounds, we use the following two problems.

InnerProduct(d, m). Given $d, m \in \mathbb{N}$, the input is a private dataset $x \in \{0,1\}^d$, and a sequence of m public query vectors $q^1, \ldots, q^m$ with $q^j \in \{0,1\}^d$. The output consists of private estimates of the inner products $\langle x, q^1\rangle, \ldots, \langle x, q^m\rangle$. Two neighboring instances differ in one bit of x.

Based on the results from [19, 23, 25, 52], Jain et al. [44] used the following lower bound on a linear number of inner product queries to show lower bounds for fully dynamic count of distinct elements.

Theorem 9 (19, 23). *There exists a constant $\zeta \in \mathbb{N}$ such that for any $d \in \mathbb{N}$, $\varepsilon \in [0,1]$, and $\delta \in [0, 1/3]$, any (ε, δ)-dp mechanism in the exact setting that solves InnerProduct$(d, \zeta d)$ up to additive error α satisfies $\alpha = \Omega(\sqrt{d})$.*

1-Way-Marginals(n, d). Given $n, d \in \mathbb{N}$, the input is a private matrix $Y \in (\{0,1\}^d)^n$, and the output consists of private estimates of the normalized column sums $\frac{1}{n}\sum_{i=1}^n Y_{i,j}$ for each column $j \in [d]$. Two neighboring matrices for this problem differ in a single row.

We extend the following lower bounds on the additive error of 1-Way-Marginals in Section 6, to a lower bound on the product of multiplicative and additive error.

Theorem 10 (13, 39). *For every $\varepsilon, \delta \in [0,1]$ and $d, n \in \mathbb{N}$, there exists $\alpha = O(1)$, such that if there exists an (ε, δ)-differentially private mechanism to solve 1-Way-Marginals(n, d) with additive error α, then if $\delta = 0$, $n = \Omega(d/\varepsilon)$ and if $\delta > 0$ and $\delta = o(1/n)$, $n = \widetilde{\Omega}(\sqrt{d}/\varepsilon)$.*

Jain et al. [46] used a sequential embedding technique from this problem to prove polynomial lower bounds under continual observation, which was then extended by other works to show similar bounds for other problems under continual observation [42, 44, 59].

3 Technical Overview

3.1 Event-level incremental graph problems

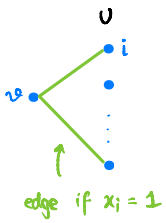
Our lower bounds in this setting reduce from the inner product query problem, which consists of a secret dataset $x \in \{0,1\}^d$, and a sequence of $m = \Theta(d)$ query vectors $q^1, \ldots, q^m$ with $q^j \in \{0,1\}^d$. It has been shown [19, 23] that answering all the inner product queries $\langle x, q^1\rangle, \ldots, \langle x, q^m\rangle$ privately must incur an additive error of $\Omega(\sqrt{d})$ on at least one of the answers.

Previous dynamic lower bound reductions from this problem were restricted to functions that were *additive across connected components*. These reductions proceeded by constructing a gadget where each component corresponded to one bit of x and one bit of q^j, where the function value of the i^{th} component was 1 whenever the bits x_i and q^j_i were simultaneously equal to 1, and was 0 otherwise. Since the function was additive, one could sum up the function values across all components to obtain $\langle x, q^j \rangle$. The edges corresponding to the current query were then deleted, and edges for the next query were inserted. A generalization of these gadgets to the weighted setting, called 2-edge distinguishing gadgets, were the main tool used by the event-level reductions of [59].

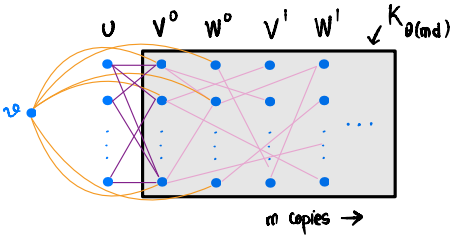
We extend these lower bound reductions in two ways. (1) First, by taking all bits of the vectors x and q^j into account simultaneously, we can obtain lower bounds even for *non-additive* functions such as estimating the core number of a vertex. (2) Prior constructions insert suitable edges when a new query vector q^j arrives, next ask a query to determine the suitable output and finally "reset the state of the construction" by deleting the previously inserted edges. We show how to construct gadgets for "incremental erasure", i.e., the state is reset using edge insertions instead of deletions.

We give the general lower bound framework in Section 4, and describe here only the lower bound for computing the core number of one fixed vertex v (i.e., the largest k such that v belongs to a subgraph with minimum degree $\geq k$), which implies a lower bound for computing the core number of all vertices. The general version of this lower bound is formally stated in Theorem 1. Our reduction places v into a subgraph where its core number is the sum of two quantities: a value independent of the current query, and the desired inner product value.

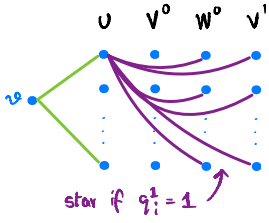
The input to our reduction is a private dataset $x \in \{0,1\}^d$ and (non-private) vectors $q^1, ..., q^m \in \{0,1\}^d$. Our goal is to build a graph and a sequence of edge insertions such that the core number of vertices of the graph after specific updates allow us to compute all the inner-products $\langle x, q^i \rangle$.



Our graph consists of a dedicated vertex v whose core-number at various times in the input sequence will be the value of inner products; a set $U = \{u_1, \ldots, u_d\}$ that will encode the private dataset x; and sets of vertices V^j, W^j (each with d vertices) that will be used for the j^{th} query. This creates a graph with $\theta(md)$ vertices. To encode the dataset x, we add an edge (v, u_i) if the i^{th} bit of the secret dataset x is 1. The answer to each query could be as high as d, and we want to increase the core number of v by one for each query bit. The core number of v at the end of the reduction could thus be as high as $\Omega(md)$. We want the query vertices to have high core numbers, since the core number of v should increase by 1 whenever a query bit and dataset bit are set to 1, which requires v to belong to progressively larger cores. Towards this, we insert a complete graph between all V and W vertices, giving a $\theta(md)$-complete graph with $\theta(m^2d^2)$ edges.

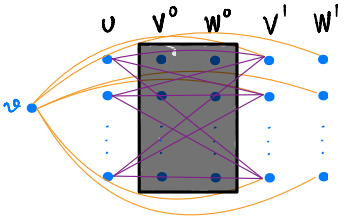


Answering the first query goes as follows. We start with v being connected to all of $V^0 \cup W^0$. There is a complete bipartite graph between U and V^0, and *no edges* between U and W^0. Before the first query is inserted, it holds that the degree of v is $2d + \|x\|_1$, while its core number is $2d$ since all of its neighbors in U do not belong to any subgraph with degree $\geq 2d$. Since the core number is at most the degree of a vertex, this gives a slack of $\|x\|_1$ for possible increase in the core number of v. Whenever a query bit *and* dataset bit are simultaneously set to 1, we use this slack to increase the core number of v by 1.



More specifically, to answer the first query $\langle x, q^1 \rangle$, we insert specific "query edges". For each i with $q_i^1 = 1$, we add all the edges from u_i to W^0 and V^1. If $q_i^1 = 0$, then we do not insert any edges. We show that after all the query edges are inserted, the core number of v is exactly $k = 2d + \langle x, q^1 \rangle$. Firstly, any vertex u_i with $q_i^1 = 0$ cannot belong to a subgraph with induced degree $\geq k$, since the degree of u_i in the whole graph is $\leq d + 1$. This means that all neighbors of v in U whose query bit is set to 0 do not contribute to its core number. This upper bounds the core number of v by k.

To show that the core number of v is at least k as well, we show that there exists a subgraph with minimum degree k as certificate: this is the subgraph induced by the set of vertices $S = \{v\} \cup \{u_i : q_i^1 = 1\} \cup V^0 \cup W^0 \cup V^1 \cup W^1$. First, all vertices in V^0, V^1, W^0 and W^1 have degree more than k: indeed, they induce a complete graph on $4d$ vertices, hence have degree at least $4d - 1 \geq k$. Second, the vertices in $\{u_i : q_i^1 = 1\}$ are connected to all of V^0 (as all the vertices of U), and by construction to all vertices in W^0 and V^1. Hence, they have degree at least $3d \geq k$. Finally, the degree of v is precisely k: it has $2d$ neighbors in $V^0 \cup W^0$, and by construction exactly $\langle x, q^1 \rangle$ neighbors in $\{u_i : q_i^1 = 1\}$. Thus, the core number of v is at least k. Together with the previous argument it follows that the core number of v is exactly k and the answer $\langle x, q^1 \rangle$ can be computed by subtracting $2d$ from the core number of v.



To complete the processing of the first query, we then "reset" the gadget by adding all missing edges from U to W^0 and V^1, and also adding edges from v to all of $V^1 \cup W^1$. This transforms the gadget to the same "state" it was before the first query, with V^1 and W^1 replacing the roles of V^0 and W^0: v is connected to all of V^1 and W^1, vertices in U are connected to V^1 but not W^1. The only change is the core number of v: as there is a complete graph between vertices in V and W, it increases by exactly $2d$. Therefore, one can easily repeat this process for each q^j in order to answer all queries. Doing so, the number of edges insertions to the graph is $T = O(m^2 d^2) = O(d^4)$ and the number of vertices $n = O(md) = O(d^2)$. Hence, the lower bound for inner product queries translate to a lower bound of $\Omega(\min\{\sqrt[8]{T}, \sqrt[4]{n}\})$ for computing core numbers in insertion-only streams.

We now discuss mechanisms for incremental graph problems with no multiplicative (but with additive) error. We call this the *exact setting*. While the earlier upper bounds in the exact setting were achieved by a combination of (i) the mechanism that recomputes every $\sqrt[3]{T}$ time steps and (ii) the mechanism that returns 0 at all time steps [59], we show that in the incremental setting, the natural mechanism is to use the sparse vector technique (SVT) [28] to track the true answer and then to update it every time the true value differs significantly from the currently output value. While earlier use of SVT in the incremental setting by Fichtenberger et al. [35] was only for $(1 + \zeta)$-approximate mechanisms, we show that it can be directly used to obtain upper bounds for mechanisms in the exact setting as well. We present the details in Section 5. Our mechanism is similar to the mechanism of Henzinger et al. [42] which works in the fully dynamic setting. Applying this approach to estimating the core number of a vertex gives an upper bound of $O(\min\{\sqrt[6]{T}, \sqrt[3]{n}\})$.

3.2 Item-level fully dynamic graph problems

Previous lower bounds by Raskhodnikova and Steiner [59] for mechanisms in the exact setting were reductions from the one-way marginals problem, which is the problem of estimating the

column sums of an $n \times d$ binary matrix. To show multiplicative approximation lower bounds, we white-box the one-way marginals lower bounds on the additive error shown by Hardt and Talwar [39] and Bun et al. [13], and use them to give multiplicative approximation lower bounds. While earlier work concentrated only on the additive error, we show that even when the *product of the multiplicative and the additive error* is small, one can decode column sums accurately for the one-way marginals problem. Our reduction makes use of the specific structure of the instances that are used in the 1-Way-Marginals lower bound reductions, which we expand on next.

We give a short overview of our modified reduction. In the original reduction of Bun et al. [13], each row of the matrix, an element of $\{0,1\}^d$, corresponds to a so-called *fingerprinting code* of a user. The goal is to return the normalized column sums, rounded in some way to $\{0,1\}^d$, such that the following property is true for most columns: If all the users have the same bit in a column, then the returned bit has to match this common bit. In [13] it is shown that any (ε, δ)-differentially private algorithm for the one-way marginals problem when rounded to the closest bit, has to have additive error at least a constant (for the normalized setting, which corresponds to linear additive error in the unnormalized setting).

We show that for the same fingerprinting matrices, the rounding condition can be extended to give a lower bound even for mechanisms with multiplicative and additive error. The key contribution here is to notice that the only columns that are constrained by the above condition are the uniform columns: the all-1s column and the all-0s column; for all other columns, any answer is acceptable. Since only the uniform columns are constrained, we show that even when the product of the multiplicative and additive error is smaller than some constant, the rounding algorithm can still distinguish the all-1s column from the all-0s column. Informally, our lower bound is as follows (see Corollary 2 and Theorem 11 for formal statements).

Theorem 11 (Informal). *Any (ε, δ)-dp mechanism for (normalized) 1-Way-Marginals(n, d) with multiplicative error η and additive error α satisfies $\eta \cdot \alpha = \Omega(1)$, with $n = O(d)$ for $\delta = 0$ and $\widetilde{O}(\sqrt{d})$ for $0 < \delta = o(1/n)$.*

Since the range of (normalized) 1-Way-Marginals(n, d) is $[0,1]^d$, for the unnormalized version of 1-Way-Marginals(n, d) this lower bound translates into $\eta \cdot \alpha = \Omega(n)$. We can use this fact to show that mechanisms with constant multiplicative error η cannot have additive error linear in the length of the output range. The hardness results shown by [39] and [13] follow as special cases of our theorem by setting $\eta = 1$, which correspond to mechanisms in the exact setting.

Armed with our lower bound, we show that a class of reductions that includes those presented earlier by Raskhodnikova and Steiner [59] using 1-edge distinguishing gadgets for item-level fully dynamic lower bounds are actually approximation-preserving reductions to the 1-Way-Marginals problem. While their reductions only work for functions that are additive across connected components (such as maximum matching), our general reductions work for a much larger class of graph functions that includes estimating the (global) mincut, *s-t* mincut, and core number of a vertex. As did [59] we define a general framework as follows. We first define a class of graphs, called *marginals solving family* (and informally called *gadgets*), that have to fulfill certain properties. Our gadgets are different from the gadgets in [59] in the sense that we do not necessarily have vertex-disjoint gadgets corresponding to each row of the matrix. In fact, this is the reason why our lower bounds are not limited to additive functions. Then we show that if a graph problem has a gadget that fulfills the properties, then there is a lower bound for the product of the additive and multiplicative error for any continual mechanism maintaining that graph problem. The gadgets

have parameters, allowing the lower bounds to vary for different graph problems depending on the value of the parameters.

If a graph problem has a marginals solving family $\mathcal{F}$, then any all-0, resp. all-1 column in the matrix M of the1-Way-Marginals problem is mapped by $\mathcal{F}$ into a specific graph whose value for the desired graph problem is 0, resp. a large value (which equals the lower bound that we can prove). The transformation of the graph from one column to the next column is done by edge updates and gives rise to a sequence of graphs. The mapping from 1-Way-Marginals to a graph sequence needs to be smooth with respect to the neighboring relation, so that two neighboring databases for 1-Way-Marginals map to two item-level neighboring graph sequences. For this, our reduction for each graph problem is designed such that all of the updates corresponding to the i^{th} row (for the different columns) of the 1-Way-Marginals instance are mapped to (potentially multiple) updates on the *same* edge, say e_i, of the desired graph problem for every $i \in [n]$, and each of the edge updates changes the value of the graph problem of interest by exactly 1 (or a constant, depending on the problem).

We describe our general lower bound framework for the item-level setting in Section 7, and here, in order to show the general idea of our lower bounds, we only explain our lower bound for calculating s-t-MinCut (for fixed vertices s and t). As mentioned above, we do so by reducing from 1-Way-Marginals. The input to our reduction is a private matrix $M \in \{0,1\}^{n\times d}$. We introduce a sequence of $T = n + 2nd$ updates on a graph G with $n+2$ vertices (including s and t) such that the size of its s-t-MinCut at time step $t_i = 2ni$ is equal to the i^{th} column sum ($\sum_{j\in[n]} M_{j,i}$) for all $i \in [d]$. Let the vertex set of G be $V = \{s, t, v_1, \ldots, v_n\}$. The initial edge set is empty and the update sequence is as follows:

Initialization: In the first n time steps, we insert the edges (v_i, t) for every $i \in [n]$.

Calculating the i^{th} column sum (for every $i \in [d]$): Time steps $[(2i-1)n+1, (2i+1)n]$ are corresponding to the i^{th} column sum. In the first half of these time steps, in time step $(2i-1)n + j$, we insert the edge (s, v_j) if $M_{j,i} = 1$ and have a no-op time step otherwise. Note that after time step $2in$, for every j with $M_{j,i} = 1$, v_j has an edge to both s and t (and only to them), and for every other j, v_j only has an edge to t. Note that $(\{s\}, V \setminus \{s\})$ is a minimum (s,t)-cut, and the edge set corresponding to this cut is $\{(s, v_j) | M_{j,i} = 1\}$. Therefore, the value of s-t-MinCut at this time step is $\sum_{j\in[n]} M_{j,i}$, which is the i^{th} column sum. Then, in the next n time steps, at time step $2in + j$ (for every $j \in [n]$), we undo the update on time step $(2i-1)n + j$, i.e, remove the edge (s, v_j) if $M_{j,i} = 1$ and have a no-op time step otherwise. Note that after time step $(2i+1)n$, the edge set of the graph is, regardless of the entries of the matrix M, equal to $\{(v_j, t) | j \in [n]\}$.

Note that in the update sequence above, changing the j^{th} row of M, only affects insertions and deletions of the edge (s, v_j). Thus, the sequences corresponding to any two neighboring instances of 1-Way-Marginals, will be item-level neighboring update sequences for the s-t-MinCut problem. Note that the s-t-MinCut instance mentioned above has $N = \theta(n)$ nodes and $T = \theta(nd)$ time steps. Thus, by using an (ε, δ)-DP mechanism that solves s-t-MinCut with multiplicative error η and additive error α for inputs with at most N nodes and at most T updates as a blackbox, one can have an (ε, δ)-DP mechanism that solves 1-Way-Marginals(n, d) with multiplicative error η and additive error $\frac{\alpha}{n}$. Where for $\delta > 0$, we can find $d = \theta((T\varepsilon)^{2/3})$ and $n = \widetilde{\theta}(\min(\sqrt[3]{T/\varepsilon^2}, N, T))$, and for $\delta = 0$, $d = \theta(\sqrt{T\varepsilon})$ and $\theta(\min(\sqrt{T/\varepsilon}, N, T))$, such that the number of nodes in the reduction above is less than N and the number of updates is less than T, and for both cases the number of

samples is less than the sample complexity stated in Theorem 11. Thus, from our lower bound on 1-WAY-MARGINALS we have for every (ε, δ)-DP and (η, α)-accurate mechanism for s-t-MINCUT, if $0 < \delta < o(1/T)$, then $\eta \cdot \alpha = \widetilde{\Omega}(\min(\sqrt[3]{T/\varepsilon^2}, N, T))$, and if $\delta = 0$, then $\eta \cdot \alpha = \widetilde{\Omega}(\min(\sqrt{T/\varepsilon}, N, T))$.

3.3 Incremental simultaneous norm estimation problems

We next show our techniques can also be extended to non-graph problems, namely to the popular streaming problem of estimating norms of the frequency vector of the stream. Specifically, we show polynomial lower bounds for simultaneous norm estimation (SNE), by showing that estimating all TOPK norms of the frequency vector in the incremental model requires large additive error. This result was stated more formally in Theorem 3.

Our reduction constructs a stream of insertions on n elements such that for each query q^j, the number of elements with the highest frequency corresponds to the value of the inner product $k^* = \langle x, q^j \rangle$ at a certain time step, and the remaining elements have strictly smaller frequency. In our reduction, this highest frequency will be $j+1$ for the j^{th} query and there will be exactly k^* many items with frequency $j+1$. Thus, TOP-k will equal [TOP-$(k-1)$] + $(j+1)$ for all $k \le k^*$, and for all $k > k^*$, TOP-k will be at most [TOP-$(k-1)$] + j. When viewing the plot of k against TOP-k, the slope of this plot is $j+1$ until k^*, and it is at most j afterwards. We use the private output of the mechanism to determine when the slope of the plot changes from $j+1$ to j. The function that maps k to TOP-k is stable in the sense that

$$\text{privateEstimate}(k^*) - k^* \le O(\text{error of the TOP-}k\text{ mechanism}).$$

This ensures that the accuracy guarantees of the TOP-k mechanism translate into accuracy guarantees for estimating this "change point" k^*, which is then the accuracy guarantee for inner product as well. The known lower bounds for inner product then give a $\Omega(\min\{\sqrt[4]{T}, \sqrt{n}\})$ lower bound on the additive error for estimating TOPK norms.

We complement this lower-bound by an upper bound that shows that mechanisms with an arbitrarily small constant multiplicative approximation of $(1+\zeta)$, it is possible to reduce the additive error to polylogarithmic (see Theorem 4 for an exact statement). This mirrors our previous graph results, and shows again that the difficulty of the problem lies in avoiding any multiplicative approximation. Note that the SNE problem comes in two flavors, where the goal of the first is to maintain a vector (called the *vector version*) whose norm approximates the frequency vector for all the relevant norms, while the goal of the second (called the *data structure version*) only requires to maintain a data structure that can be queried to obtain any norm. We give mechanisms for both versions and discuss the vector version first.

The first straightforward mechanism for the incremental monotone symmetric norm estimation problem would be to just compute a private histogram of the input stream, i.e., for each element i maintain a private estimate $\hat{f}_i$ of its frequency. One can guarantee using a continual histogram mechanism that $\hat{f}_i - f_i \le \log^2 T$. While this is good for any individual element – close to the best one can do, in light of the lower bound of $\log T$ [26] – this yields a very poor approximation for some norms: for instance, the ℓ_1 norm of f and $\hat{f}$ may differ by $n \log^2 T$.

To improve over this, let $\tau^f \approx \log^2(T)/\zeta$ be a value that we will use as a threshold. We first note that whenever f_i is large, $\hat{f}_i$ is a very good multiplicative approximation to f_i. More specifically, when f_i is larger than the threshold τ^f, then $\hat{f}_i \in (1 \pm \zeta) f_i$ for some constant ζ. Thus, we are left with dealing with low-frequency elements, for which the error can accumulate. Following the approach of Blasiok et al. [6] from the non-private setting, we deal with these elements as follows.

(1) First, we group them into *levels* of elements with approximately the same frequency: ideally, we would like the j^{th} level to contain all elements with frequency $f_i \in [2^j, 2^{j+1})$. (2) If that were possible, then we could estimate the number of elements in each level, called the *size of the level*. As earlier, if the size of the level b_j is large enough, then we have a correct private estimate $\hat{b}_j$. Thus, we could represent the level j by $\hat{b}_j$ elements with frequency 2^j which is a private proxy for those elements. (3) What remains are elements with low frequency present in levels with small size: the combination of those two conditions ensure that there cannot be too many elements satisfying this condition, and that they also cannot contribute significantly to any norm. We replace their frequencies by 0, and incorporate the error accrued in the norm estimate into the additive error.

While the above argument almost works, there is one crucial issue with respect to privatizing the frequencies. To privately determine if an element has high frequency, we can check if the private estimate $\hat{f}_i$ is larger than the threshold τ^f. However, to determine the level size for the elements with low frequency, i.e., $f_i < \tau^f$, we *must* use their true frequency. This is because even small variations in the privatized frequency could cause high errors for the estimated level sizes. We can easily deal with elements that have $\hat{f}_i \geq \tau^f$, or that have $f_i < \tau^f$ *and* $\hat{f}_i < \tau^f$, however, the critical issue comes from the fact that there are elements for which $f_i \geq \tau^f > \hat{f}_i$. Indeed, the guarantee we have on $\hat{f}_i$ is only that $\hat{f}_i \leq f_i$ and $\hat{f}_i - f_i \leq \log^2(T)$, but $\hat{f}_i$ is computed from f_i by adding a random noise which cannot be exactly estimated in advance. Thus, one of our key contributions is to design the threshold such that the dropped elements, namely those with $f_i \geq \tau^f > \hat{f}_i$, have a negligible contribution to the norm. To show this, we need the threshold τ^f to be chosen from a carefully designed probability distribution where, with probability $2/3$, only a tiny fraction of elements are dropped; using properties of monotone symmetric norms, we can conclude that any such norm of the frequency vector does not change much even after dropping these elements.

Our mechanism thus first picks τ^f and then computes the following vector: for every element i with $\hat{f}_i \geq \tau^f$, the estimated frequency is $\hat{f}_i$, and for any level of elements with frequency in $[2^j, 2^{j+1})$ and $2^{j+1} < \tau^f$, the mechanism computes the size of the level $\hat{b}_j$ privately. If $\hat{b}_j > \log^2(T)/\zeta$, then, as before, $\hat{b}_j$ is almost the true size b_j, i.e., $\hat{b}_j = (1 \pm \zeta)b_j$. Therefore the mechanism adds $\hat{b}_j$ many elements with estimated frequency 2^j to the output vector. If $\hat{b}_j \leq \log^2(T)/\zeta$, then there are few elements and their frequency is small, so we can simply ignore them. This incurs an additive error $\text{polylog}(T) \cdot L(e_1)$ for any monotone symmetric norm L, where $L(e_1)$ is the L-norm of the first standard basis vector. The accuracy guarantee holds with constant probability for this mechanism at any fixed time step. For the data structure version of the problem, we boost the accuracy by running a main mechanism which internally runs $O(\log T)$ copies of the mechanism for the vector version. The main mechanism, thus, maintains $O(\log T)$ frequency vectors. When given a query, the main mechanism calculates the queried norm on all the frequency vectors, determines the median frequency vector based on the norm, and outputs its answer. This then gives Theorem 4.

We note that the basic approach of separating high and low frequency elements, and estimating the levels' sizes, was introduced in the non-private setting by Blasiok et al. [6], and using continual histogram to estimate the relevant quantity is quite a natural idea. Our key contribution of this result is the randomization of the threshold τ^f, and the accompanying accuracy analysis.

4 Incremental Graph Lower Bounds

To show our lower bounds for algorithmic problems in graphs, we introduce the notion of *extendable bitwise AND gadgets*, which is an extension of 2-edge distinguishing gadgets of Raskhodnikova and Steiner [59] to the incremental graph setting. Intuitively, our gadgets allow embedding an inner

product query using the "AND part" of the gadget, then the "extensibility" of the gadgets allows "resetting the impact" of the previous query for inserting the next query.

Since our gadgets are defined to embed inner product queries into our incremental graph problems, we only consider the case when $m = \zeta d$ for the constant ζ from Theorem 9 to simplify notation, even though it could be defined for any sequence of queries in the most general case.

Extendable bitwise AND gadget(d) for graph problem g. Given a dimension $d \in \mathbb{N}$, an extendable bitwise AND gadget consists of the following: positive and increasing functions $v(d)$ and $t(d)$, where the former denotes the number of nodes in the gadget, and the latter an upper bound on the number of edge insertions, an initial graph $H_{\text{init}} = (V, E_0)$ with $|V| = v(d)$, along with

(1) edges $e_1, \dots, e_d \in \binom{V}{2}$ such that for any $x \in \{0,1\}^d$, one can define $H_x^1 = H_{\text{init}} \cup \{e_i\}_{i \in [d]: x_i = 1}$ satisfying the conditions below, and

(2) for any sequence of $m = \zeta d$ query vectors q^j with $j \in [m]$ as in Theorem 9, a sequence of pairs of edge sets (S^j, T^j) and decoding functions $\text{dec}_j : \text{Range}(g) \to \mathbb{R}$ such that the following conditions hold:

   (a) dec_j is 1-Lipschitz continuous with respect to the error norm $\|\cdot\|$ on g, i.e., for any pair $x, y \in \text{Range}(g)$, $|\text{dec}_j(x) - \text{dec}_j(y)| \le \|x - y\|$.

   (b) inserting S^j into H_x^j creates a graph $Q_x^j = H_x^j \cup S^j$ which satisfies $\text{dec}_j(g(Q_x^j)) - \text{dec}_j(g(H_x^j)) = \langle x, q^j \rangle$,

   (c) inserting T^j into Q_x^j creates the graph $H_x^{j+1} = Q_x^j \cup T^j$, i.e., it "resets the query".

When the function g is real-valued, we use the identity function $\text{dec}_j = \text{id}$ as the decoding functions. In general we use the gadgets for all dimensions $d \in \mathbb{N}$ together to construct our lower bounds, so we use $v(d)$ and $t(d)$ to denote the relevant parameters with their dependence on d. We hide the dependence on g when talking about a single function g. We may insert $O(d)$ many $\perp$s in the stream, but since it will always hold that $t(d) = \Omega(d)$,[4] these $O(d)$ insertions of $\perp$ are asymptotically subsumed by $t(d)$. The lower bound of $\Omega(\sqrt{d})$ from inner product then directly leads to lower bounds for the variables of the graph stream, T and n.

Lemma 1. *Let $\varepsilon \in [0,1]$ and $\delta \in [0, 1/3]$, and let g be a graph problem that admits an extendable bitwise AND gadget with $v(d)$ vertices and $t(d)$ edges. For any (ε, δ)-dp mechanism for the event-level setting that is (i) solving g in the exact setting and (ii) running on n-node graphs for T timesteps with additive error α, it holds that*

$$\alpha = \Omega\left(\min\left\{\sqrt{t^-(T)}, \sqrt{v^-(n)}\right\}\right),$$

where f^- is the generalized inverse of a function $f : \mathbb{N} \to \mathbb{N}$ defined as $f^-(x) = \sup\{d \mid f(d) \le x\}$.

We use this to prove lower bounds for natural graph problems by presenting such gadgets.

Lemma 2. *There exist extendable bitwise AND gadgets for the following problems and parameters:*

$$\begin{aligned} \textsc{IncMatching}: &\quad v(d) = O(d^2) \quad \text{and} \quad t(d) = O(d^2), \\ \textsc{IncDegHist}: &\quad v(d) = O(d) \quad \text{and} \quad t(d) = O(d^2), \\ \textsc{IncKCore}: &\quad v(d) = O(d^2) \quad \text{and} \quad t(d) = O(d^4). \end{aligned}$$

[4] This is because $t(d)$ is a positive and increasing function.

Algorithm 1: Reduction from InnerProduct to solving g in the exact setting

Input: A secret dataset $x \in \{0,1\}^d$, and $m = \zeta d$ queries $q_1, \ldots, q_m$ with $q^j \in \{0,1\}^d$.
Output: Inner product answers $\langle x, q^1 \rangle, \ldots, \langle x, q^m \rangle$

Set $d = \min\{t^-(T), v^-(n)\}$, $n' = v(d)$, and $T' = t(d)$.
Initialize an (ε, δ)-dp mechanism $\mathcal{A}$ for solving g in the exact setting with n' vertices for T' timesteps with additive error α.
Insert edges of H_{init} in any arbitrary order.
for $t = 1, \ldots, d$ **do**
  if $x_t = 1$ **then** insert edge e_t **else** insert $\perp$ ▷ insert dataset x
for $j = 1, \ldots, m$ **do**
  $\widetilde{h} \leftarrow \text{dec}_j$ evaluated on the query value from $\mathcal{A}$ on the current graph H_x^j.
  Insert all edges in S^j in any arbitrary order. ▷ insert query j
  $\widetilde{q} \leftarrow \text{dec}_j$ evaluated on the query value from $\mathcal{A}$ on the current graph Q_x^j.
  Set $\text{inprod}^j = \widetilde{q} - \widetilde{h}$.
  Insert all edges in T^j in any arbitrary order. ▷ neutralize query j

5 Insertions-only Graph Upper Bounds

In this section, we first state our insertions-only upper bound for a set of graph problems including IncMatching and IncKCore, then we mention our upper bound for IncDegHist.

The following can be shown using the sparse vector technique (SVT).

Lemma 3. *Let $g : \mathcal{G} \to \mathbb{R}$ be a monotone function on graphs with constant sensitivity such that for every incremental stream of length T on graphs with n vertices, the output of g lies in the range $[L(n,T), R(n,T)]$ of length $\ell = R(n,T) - L(n,T)$. Then, for $\varepsilon > 0$ and $\delta \in [0,1)$ there is an (ε, δ)-DP mechanism M that answers g with additive error α, where for $\delta > 0$, $\alpha = \widetilde{O}(\ell^{1/3}\sqrt{\ln(1/\delta)}/\varepsilon)$ and for $\delta = 0$, $\alpha = \widetilde{O}(\sqrt{\ell}/\varepsilon)$.*

Using the lemma above, we have the following:

Corollary 1. *For $\varepsilon > 0$ and $\delta \in [0,1)$ there we have the following upper bound for the additive error of (ε, δ)-DP mechanisms for IncMatching: for $\delta > 0$, $\alpha = \tilde{O}(\min(n,T)^{1/3}\sqrt{\ln(1/\delta)}/\varepsilon)$ and for $\delta = 0$, $\alpha = \tilde{O}(\sqrt{\min(n,T)}/\varepsilon)$. Also, for IncKCore$(v)$, we have the following bounds: for $\delta > 0$, $\alpha = \tilde{O}(\min(n,\sqrt{T})^{1/3}\sqrt{\ln(1/\delta)}/\varepsilon)$ and for $\delta = 0$, $\alpha = \tilde{O}(\sqrt{\min(n,\sqrt{T})}/\varepsilon)$*

To show the following upper bound, we use n separate counters (one for each degree) to calculate the number of vertices with each degree. Then we use the advanced composition theorem (Theorem 8) to find the right privacy parameter for the counters, so that the whole mechanism is private.

Lemma 4. *For any $\varepsilon, \delta \in (0,1)$, there is an (ε, δ)-differentially private mechanism in the exact setting that solves IncDegHist with additive error α, where $\alpha = \widetilde{O}(\sqrt{n\log(1/\delta)}/\varepsilon)$.*

6 Lower Bounds for 1-Way-Marginals

In this section, we will first review the definition of sample complexity and 1-Way-Marginals and then prove two different lower bounds for its sample complexity. The first lower bound is for ε-DP mechanism and uses a more restrictive notion of accuracy, namely (η, α)-accuracy than the second lower bound that applies for (ε, δ)-DP mechanisms and uses (η, α, β)-accuracy.

Definition 2. A set of counting queries Q has *sample complexity* n^* for (ε, δ)-DP and (η, α)-accuracy, resp., (η, α, β)-accuracy if n^* is the smallest number of samples such that there exists an (ε, δ)-DP and (η, α)-accurate, resp. (η, α, β)-accurate mechanism for Q.

Definition 3 (1-Way-Marginals(n, d))**.** In this problem, given a private matrix $Y \in \{0,1\}^{n \times d}$, the mechanism must output an estimation of the average of the j^{th} column $(\frac{1}{n}\sum_i Y_{i,j})$ for each $j \in [d]$.

6.1 Pure Differential Privacy

Now, we prove the following bound on the sample complexity of approximating (with multiplicative error η and additive error α) this problem under pure differential privacy. This is an extension of the proofs of Hardt and Talwar [39] and De [19] to allow multiplicative approximations.

Definition 4 ((η, α)-Accuracy for 1-Way-Marginals(n, d))**.** Let $\eta \geq 1$ and $\alpha \in [0, 1]$ be parameters. An output sequence $(a_1, \ldots, a_d)$ is an (η, α)-accurate answer to 1-Way-Marginals(n, d) on database $D \in (\{0,1\}^d)^n$ if

$$\frac{1}{\eta} \cdot \frac{1}{n}\sum_{i=1}^{n} D_{i,j} - \alpha \leq a_j \leq \eta \cdot \frac{1}{n}\sum_{i=1}^{n} D_{i,j} + \alpha$$

for all columns $j \in [d]$.

A mechanism $\mathcal{M}$ is (η, α)-accurate for 1-Way-Marginals(n, d) if for every database $D \in (\{0,1\}^d)^n$,

$$\Pr[\mathcal{M}(D) \text{ is an } (\eta, \alpha)\text{-accurate answer to 1-Way-Marginals}(n, d) \text{ for } D] \geq 2/3,$$

Theorem 12. *Let $\eta \geq 1$ and $\alpha > 0$ such that $\eta \cdot \alpha < \frac{1}{2}$. For every ε-DP mechanism $\mathcal{M} \colon \{0,1\}^{n \times d} \to [0,1]^d$ for 1-Way-Marginals(n, d) that is (η, α)-accurate, it holds that $n = \Omega(d/\varepsilon)$.*

Thus, we have the following lower bound on the additive error on ε-DP η-approximations of 1-Way-Marginals(n, d):

Corollary 2. *There exist constants $c_1, c_2 > 0$, such that if $n \leq c_1 \cdot d/\varepsilon$, then for every ε-DP mechanism $\mathcal{M} \colon \{0,1\}^{n \times d} \to [0,1]^d$ that is (η, α)-accurate it holds that $\eta \cdot \alpha > c_2$.*

6.2 Approximate Differential Privacy

We show that the lower bound for approximate DP given by [13] can be extended to mechanisms with multiplicative errors. We follow their proof structure, and recall two definitions that will be used in this section. We first introduce the definition of accuracy that we will be using in this subsection, which allows for a β fraction of the queries to be answered with unbounded error.

Definition 5 ((η, α, β)-Accuracy for 1-Way-Marginals(n, d))**.** Let $\eta \geq 1$ and $\alpha, \beta \in [0, 1]$ be parameters. An output sequence $(a_1, \ldots, a_d)$ is an (η, α, β)-accurate answer to 1-Way-Marginals(n, d) on database $D \in (\{0,1\}^d)^n$ if

$$\frac{1}{\eta} \cdot \frac{1}{n}\sum_{i=1}^{n} D_{i,j} - \alpha \leq a_j \leq \eta \cdot \frac{1}{n}\sum_{i=1}^{n} D_{i,j} + \alpha$$

for at least a $1 - \beta$ fraction of columns $j \in [d]$. A mechanism $\mathcal{A}$ is (η, α, β)-accurate for 1-Way-Marginals(n, d) if for every database $D \in (\{0,1\}^d)^n$,

$$\Pr[\mathcal{A}(D) \text{ is an } (\eta, \alpha, \beta)\text{-accurate answer to 1-Way-Marginals}(n, d) \text{ for } D] \geq 2/3$$

Note that if a mechanism $\mathcal{M}$ is (η, α, β)-accurate, then for every β' such that $\beta \leq \beta' \leq 1$, $\mathcal{M}$ is also (η, α, β')-accurate. Thus, if there is no (η, α, β')-accurate mechanism for 1-Way-Marginals(n, d), then there does not exist an (η, α, β)-accurate mechanism for 1-Way-Marginals(n, d) (for $0 \leq \beta \leq \beta' \leq 1$). This means proving an impossibility result for (η, α, β)-accurate mechanisms for 1-Way-Marginals(n, d) will imply the impossibility of $(\eta, \alpha, 0)$-accurate mechanisms (which is equivalent to (η, α)-accuracy).

Definition 6 (Reidentifiable Distribution)**.** For $n, d \in \mathbb{N}$, let $\mathcal{D}$ be a distribution on n-row databases with d columns, $D \in (\{0,1\}^d)^n$. The distribution $\mathcal{D}$ is γ-reidentifiable from (η, α, β)-accurate answers to 1-Way-Marginals(n, d) if there exists an adversary $\mathcal{B}\colon (\{0,1\}^d)^n \times [0,1]^d \to [n] \cup \{\perp\}$ such that for every mechanism $\mathcal{A}\colon (\{0,1\}^d)^* \to [0,1]^d$, the following both hold:

(1) $\Pr_{D \leftarrow_R \mathcal{D}}[(\mathcal{B}(D, \mathcal{A}(D)) = \perp) \wedge (\mathcal{A}(D) \text{ is } (\eta, \alpha, \beta)\text{-accurate for 1-Way-Marginals}(n, d))] \leq \gamma$.

(2) For every $i \in [n]$, $\Pr_{D \leftarrow \mathcal{D}}[\mathcal{B}(D, \mathcal{A}(D_{-i})) = i] \leq \gamma$.

Here, D_{-i} corresponds to D after removing its i^{th} row and the probabilities are taken over the choice of D, i, and coin flips of $\mathcal{A}$ and $\mathcal{B}$. Also, $\mathcal{D}$ and $\mathcal{B}$ can share a common state.

The common state shared by $\mathcal{D}$ and $\mathcal{B}$, is an auxiliary string *aux* that $\mathcal{D}$ sends to $\mathcal{B}$ (but not $\mathcal{A}$) about the realization of D and is supposed to help $\mathcal{B}$ reidentify the index i.

Intuitively, the first condition says that with high probability $1-\gamma$, it does not simultaneously occur that $\mathcal{B}$ receives an accurate answer for the whole database D from $\mathcal{A}$ and it does not accuse any index at all. The second condition says that the probability of $\mathcal{B}$ accusing some index, whose row has been removed from the database used by $\mathcal{A}$ (which is then a false accusation) is at most γ.

We show that the existence of a reidentifiable distribution for 1-Way-Marginals implies (under a condition on the parameters) strong lower bounds on the accuracy guarantees of any (ε, δ)-differentially private mechanism for 1-Way-Marginals for certain parameters. Note that since D is public, the output of $\mathcal{B}$ is a post-processing of the output of $\mathcal{A}$.

Lemma 5. *Let $\alpha, \beta, \gamma \in [0,1]$, $\varepsilon > 0$, $\delta \in (0,1]$, $\eta \geq 1$ be parameters and let $d, n \geq 1$ be integers. If there exists a γ-reidentifiable distribution $\mathcal{D}$ from (η, α, β)-accurate answers to 1-Way-Marginals(n, d), then there is no (η, α, β)-accurate (ε, δ)-DP mechanism for 1-Way-Marginals(n, d) such that $e^{-\varepsilon} \cdot \frac{1}{n} \cdot \left(\frac{2}{3} - \gamma\right) - \delta > \gamma$. In particular, for $\varepsilon = 1$, this holds for any $\delta < 2/(3en) - \gamma - \gamma/(en)$, and so choosing $\gamma = 1/(100n)$ and $\delta \leq 1/(5n)$ suffices.*

Now, we restate a variant of the definition of fingerprinting codes introduced in [13]. Fingerprinting codes were initially introduced by Boneh and Shaw [10]. Assume there is a set of n users, $\{1, \ldots, n\}$. A fingerprinting code is a pair (Gen, Trace) of randomized mechanisms where Gen outputs a binary codebook $C \in (\{0,1\}^d)^n$ containing n codewords of length d such that c_i is the codeword of user i and Trace$\colon (\{0,1\}^d)^n \times \{0,1\}^d \to [n]$ is supposed to trace back one of the codewords used in a *coalition*'s codeword. In particular, a set $S \subseteq [n]$ can form a coalition to create a certain codeword c' from their own codewords in the codebook (C_S), maintaining some constraints (formalized in the set $F_\beta(C_S)$ of feasible codewords). Trace, given the codebook and the new codeword c', has to trace back one of the codewords i or does not accuse anyone of participating in the coalition. However, the probability of the new codeword maintaining the constraint and Trace not accusing anybody should be low; moreover, Trace should not accuse somebody who has not participated in the coalition (except with a small probability). The definition below is a variant of fingerprinting

codes with a weaker (compared to the standard definition of fingerprinting codes) constraint on feasible codewords ($F_\beta(C_S)$).

Definition 7 (Error-Robust Fingerprinting Codes [13])**.** For any $n, d \in \mathbb{N}$, $\gamma, \beta \in [0,1]$, a pair of mechanisms (Gen, Trace) is an (n,d)*-fingerprinting code with security* γ *robust to a* β *fraction of errors* if Gen outputs a codebook $C \in \{0,1\}^{n\times d}$ such that c_i is the codeword of user i and for every (possibly randomized) adversary $\mathcal{A}_{FP}$, and every coalition $S \subseteq [n]$ with codewords set $C_S = \cup_{i\in S}\{c_s\}$, if we set $c' \leftarrow \mathcal{A}_{FP}(C_S)$, then

(1) $\Pr[c' \in F_\beta(C_S) \wedge \text{Trace}(C, c') = \perp] \le \gamma$,
where $F_\beta(C_S) = \{c' \in \{0,1\}^d \mid \Pr_{j\leftarrow[d]}[\exists i \in S, c'_j = c_{ij}] \ge 1-\beta\}$

(2) $\Pr[\text{Trace}(C, c') \in [n] \setminus S] \le \gamma$,

where the probability is taken over the coins of Gen, Trace, and $\mathcal{A}_{FP}$. The mechanisms Gen and Trace may share a common state.

The existence of such Gen and Trace is proved in the same paper. (For the standard definition of fingerprinting codes, their existence was shown in [65].)

Theorem 13 (13). *For every* $d \in \mathbb{N}$*, and* $\gamma \in (0,1]$*, there exists an* (n,d)*-fingerprinting code with security* γ *robust to a* $1/75$ *fraction of errors for* $n = n(d,\gamma) = \widetilde{\Omega}(\sqrt{d/\log(1/\gamma)})$.

Note that for any $n \in \mathbb{N}$, if there exists an (n,d)-fingerprinting code with security γ robust to a β fraction of errors, then there also exists one for any $n' \in \mathbb{N}$ where $n' \le n$.

It then follows that for every $d \in \mathbb{N}$, and $\gamma \in (0,1]$, there exists a function $n'(d,\gamma) = \widetilde{\theta}(\sqrt{d/\log(1/\gamma)})$ (possibly slower growing than $n(d,\gamma)$) such that for every $n \le n'(d,\gamma)$, there exists an (n,d)-fingerprinting code with security γ robust to a $1/75$ fraction of errors. Since we will need the exact terms inside the asymptotic notation in our proofs, we use constants $c_1 > 0$ and $c_2 < 0$ such that for large enough d, the above-mentioned fingerprinting code exists for every $n \le c_1\sqrt{d}\log^{c_2}\frac{d}{\gamma}$.

The following lemma shows that the existence of fingerprinting codes implies the existence of reidentifiable distribution for 1-Way-Marginals(n,d).

Lemma 6. *Given* $n, d \in \mathbb{N}$ *and* $\beta, \gamma \in [0,1]$*, if there exists an* (n,d)*-fingerprinting code with security parameter* γ *robust to a* β *fraction of errors, then there exists a distribution* $\mathcal{D}$ *on databases* $D \in (\{0,1\}^d)^n$ *that is* γ*-reidentifiable from* (η,α,β)*-accurate answers to 1-Way-Marginals*(n,d) *if* $\eta \cdot \alpha < \frac{1}{2}$.

Now, we are ready to prove the main result of this section.

Theorem 14. *For* $d \in \mathbb{N}$*,* $n = \widetilde{O}(\sqrt{d})$*,* $\alpha \in [0,1]$ *and* $\eta \ge 1$ *such that* $\eta \cdot \alpha < \frac{1}{2}$*, there exists no* $(1, O(1/n))$*-DP mechanism for 1-Way-Marginals*(n,d) *that is* $(\eta,\alpha,1/75)$*-accurate.*

In particular, this shows that any (η,α)-approximation mechanism must have $\eta\cdot\alpha \ge 1/2 = \Omega(1)$.

Now, to extend the result above from (ε_0,δ)-DP for constant ε_0 to (ε_0,δ)-DP for any $\varepsilon \le 1$,

we use the following lemma which is similar to Lemma 2.6 of [13] implied by [49]. However, since they only showed the lemma for the case of $\eta = 1$, we show it for the general case here.

Lemma 7. *Let $\alpha, \beta \in [0,1]$, $1 \leq \eta$, let constant $\varepsilon_0 > 0$ and $0 < \varepsilon \leq \varepsilon_0$. If 1-Way-Marginals$(n,d)$ has sample complexity n^* for (η,α,β)-accuracy and (ε_0,δ)-differential privacy for $\delta = O(1/n)$, then 1-Way-Marginals(kn,d) (for $k = \lfloor \frac{\varepsilon_0}{\varepsilon} \rfloor$) has sample complexity $\Omega(\frac{n^*}{\varepsilon})$ for (η,α,β)-accuracy and (ε,δ')-differential privacy where $\delta'(kn) = O(1/(kn)) = O(\varepsilon/n)$.*

Note that the above arguments also hold for any set of counting queries. Now, by Lemma 7 and Theorem 14, we have the following:

Theorem 15. *For $\varepsilon \leq 1$, $d \in \mathbb{N}$, $n = \widetilde{O}(\sqrt{d}/\varepsilon)$, $\eta \geq 1$ and $\alpha \in [0,1]$ such that $\eta \cdot \alpha < \frac{1}{2}$, there exists no $(\varepsilon, O(1/n))$-DP mechanism for 1-Way-Marginals that is $(\eta,\alpha,1/75)$-accurate.*

As $(\eta,\alpha,1/75)$-accuracy implies (η,α)-accuracy, the corresponding statement also holds for (η,α)-accuracy.

7 Item-Level Lower Bounds for Fully Dynamic Graph Mechanisms

In this section, we will use reductions from 1-Way-Marginals to show strong lower bounds on fully dynamic item-level graph problems.

We first introduce the framework of graph problems for which we show our lower bounds.

Definition 8 (Marginals Solving Family)**.** Let $n_0 \in \mathbb{N}$ and g be a real-valued graph function. We call a family of tuples $\{(H_n = (V_n, E_n), (e_1, \ldots, e_n))\}_{n>n_0}$ a *marginals solving family for function g with weight w and size $(\nu(n), \xi(n))$* (where $\nu\colon \mathbb{N} \to \mathbb{N}$ is an increasing function of n, $\xi\colon \mathbb{N} \to \mathbb{N}$ is an increasing polynomial functions of n, and w is a positive number) if the following holds for all $n > n_0$:

- H_n is a graph with $\nu(n)$ vertices and $\xi(n)$ edges, i.e., $|V_n| = \nu(n), |E_n| = \xi(n)$,
- One of the following cases happen: (Note that the same case should hold for all values of $n > n_0$)
  - $e_1, \ldots, e_n \notin E_n$ and for every subset $S \subseteq [n]$, $g((V_n, E_n \cup \{e_i \mid i \in S\})) = w \cdot |S|$, or
  - $e_1, \ldots, e_n \in E_n$ and for every subset $S \subseteq [n]$, $g((V_n, E_n \setminus \{e_i \mid i \in S\})) = w \cdot |S|$

If there exists such a sequence for a problem g, we say that g has a marginal solving family with weight w and size (ν, ξ).

Recall that inserting $\perp$ is equivalent to a no-op and has no influence on the graph (and thus the value of the function).

Now, we use a reduction from 1-Way-Marginals to prove a lower bound for graph problems that have marginal solving families.

Theorem 16. *Let $\varepsilon, \delta, \alpha \in [0,1]$ and $\eta \geq 1$. Let $g : \mathcal{G}[V] \to \mathbb{R}$ be a function that has a marginals solving family with weight w and size (ν, ξ), where $\mathcal{G}[V]$ is the set of all graphs with vertex set V. Then, for all large enough T and N, if there exists an (ε,δ)-differentially private mechanism that solves g with multiplicative error η and additive error α, for inputs with at most T time steps on graphs with at most N vertices, then*

$$\eta \cdot \alpha = \begin{cases} \widetilde{\Omega}\left(w \cdot \min\left(\sqrt[3]{T/\varepsilon^2}, \nu^-(N), \xi^-(T)\right)\right) & \text{when } \delta > 0 \text{ and } \delta = o(1/T) \\ \widetilde{\Omega}\left(w \cdot \min\left(\sqrt{T/\varepsilon}, \nu^-(N), \xi^-(T)\right)\right) & \text{when } \delta = 0 \end{cases}$$

where $f^-(x) = \sup\{y \mid f(y) \leq x\}$ is the generalized inverse as earlier.

We show a reduction from 1-Way-Marginals to graph problems with a marginal solving family. The proof of Theorem 16 follows by fixing the right parameters in the reduction.

Lemma 8 (Reduction from 1-Way-Marginals(n, d) to graph problems that have marginals solving family)**.** *Let $n, d \in \mathbb{N}$ and $Y \in \{\{0,1\}^d\}^n$. Then for all $w \in \mathbb{R}^+$ and functions v and ξ that satisfy conditions of Definition 8 and each function $g : \mathcal{G}(V) \to \mathbb{R}$ with a marginal solving family with weight w and size (v, ξ), there exists a transformation from Y to a fully dynamic graph stream on N-node graphs, where $T = \xi(n) + 2nd$, $N = v(n)$, and the graph after the update on time step t is G_t ($\forall t \in [T]$), such that*

- *The transformations of neighboring $Y, Y' \in \{\{0,1\}^d\}^n$ give item-level neighboring streams, and*
- *defining $t_0 = \xi(n)$, for all $j \in [d]$ and $t_j = \xi(n) + 2jn$, we have $g(G_{t_j - n}) = w \sum_{i=1}^{n} Y_{i,j}$.*

Now, we use our framework to show lower bounds for a variety of problems using Theorem 16. First, let us mention the definition of 1-edge distinguishing gadgets by Raskhodnikova and Steiner [59]. This gadgets are defined for additive functions, which are defined as follows:

Definition 9 (Additive Graph Functions)**.** A function g is additive if for every graph G with connected components $C_1, \ldots, C_m$, we have $g(G) = \sum_{i \in [m]} g(C_i)$.

Definition 10 (1-Edge Distinguishing Gadgets [59])**.** Let g be a real-valued additive graph function, $H = (V', E')$ be a graph and $e \in V' \times V'$. Set $n_g = |V'|$ and $m_g = |E'|$. The pair (H, e) is called a 1-edge distinguishing gadget for g of size (n_g, m_g) and weight w if either: $e \notin E'$ and $g((V', E' \cup \{e\})) = g(H) + w$, or $e \in E'$ and $g((V', E' \setminus \{e\})) = g(H) + w$. If there exists such a pair for a problem g, we say that *g has a 1-edge distinguishing gadget.*

In their theorem, Raskhodnikova and Steiner show the following lower bound for every function g that has a 1-edge distinguishing gadget. In Corollary 3, among other problems, we show a stronger lower bound for all functions g that have a 1-edge-distinguishing gadget such that $g(H) = 0$, as defined below:

Definition 11 (0-based 1-Edge Distinguishing Gadgets)**.** Let g be a real-valued additive graph function, $H = (V', E')$ be a graph and $e \in V' \times V'$. Set $n_g = |V'|$ and $m_g = |E'|$. The pair (H, e) is called a 0-based 1-edge distinguishing gadget for g of size (n_g, m_g) and weight w if $g(H) = 0$ and either: $e \notin E'$ and $g((V', E' \cup \{e\})) = w$ or $e \in E'$ and $g((V', E' \setminus \{e\})) = w$. If there exists such a pair for a problem g, we say that *g has a 0-based 1-edge distinguishing gadget.*

Here are a few examples of 0-based 1-edge distinguishing functions:

- Matching: For this problem setting H to be two isolated vertices and e to be the potential edge between them, gives a 0-based 1-edge distinguishing gadget of size $(2, 0)$ and weight 1.
- Subgraph Counting: For the problem of counting instances of any subgraph G, a 0-based 1-edge distinguishing gadget for it is by setting H to be $G - e$ for any arbitrary edge e in G. Then $(G - e, e)$ will be such a gadget of size $(|V(G)|, |E(G) - 1|)$ and weight 1. When $G = K_3$, the cycle graph on 3 vertices, this gives the well-studied triangle counting problem.

Our lower bounds are stronger than that of [59] since our bounds hold even for mechanisms that have multiplicative error. Now, we are ready to show our lower bounds.

Corollary 3. *Let $\varepsilon, \delta \in [0, 1]$ and $\eta \geq 1$. For every large enough T and N, if there is a mechanism that is (ε, δ)-DP and (η, α)-accurate for the functions below, then the bound holds for η and α:*

- *Every g that has a 0-based 1-edge distinguishing gadget of size (n_g, m_g) and weight w:*

$$\eta \cdot \alpha = \begin{cases} \widetilde{\Omega}\left(w \cdot \min\left(\sqrt[3]{\frac{T}{\varepsilon^2}}, \frac{N}{n_g}, \frac{T}{m_g}\right)\right) & \text{when } \delta > 0 \text{ and } \delta = o(1/T) \\ \widetilde{\Omega}\left(w \cdot \min\left(\sqrt{\frac{T}{\varepsilon}}, \frac{N}{n_g}, \frac{T}{m_g}\right)\right) & \text{when } \delta = 0 \end{cases}$$

- *$f_{\geq\tau}$, KCore, and MinCut:*

$$\eta \cdot \alpha = \begin{cases} \widetilde{\Omega}\left(w \cdot \min\left(\sqrt[3]{T/\varepsilon^2}, N, \sqrt{T}\right)\right) & \text{when } \delta > 0 \text{ and } \delta = o(1/T) \\ \widetilde{\Omega}\left(w \cdot \min\left(\sqrt{T/\varepsilon}, N\right)\right) & \text{when } \delta = 0 \end{cases}$$

- *s-t-MinCut:*

$$\eta \cdot \alpha = \begin{cases} \widetilde{\Omega}\left(w \cdot \min\left(\sqrt[3]{T/\varepsilon^2}, N, T\right)\right) & \text{when } \delta > 0 \text{ and } \delta = o(1/T) \\ \widetilde{\Omega}\left(w \cdot \min\left(\sqrt{T/\varepsilon}, N, T\right)\right) & \text{when } \delta = 0 \end{cases}$$

- *EdgeCount:*

$$\eta \cdot \alpha = \begin{cases} \widetilde{\Omega}\left(w \cdot \min\left(\sqrt[3]{T/\varepsilon^2}, N^2\right)\right) & \text{when } \delta > 0 \text{ and } \delta = o(1/T) \\ \widetilde{\Omega}\left(w \cdot \min\left(\sqrt{T/\varepsilon}, N^2\right)\right) & \text{when } \delta = 0 \end{cases}$$

Remark. For s-t-MinCut and MinCut, our arguments will still hold if we extend the model to weighted graphs with maximum weight W where neighboring sequences can differ arbitrarily in updates corresponding to the edge between a single pair of vertices. The reductions would have the weight of all of the edges set to W. In that case, the lower bounds will be multiplied by W.

Acknowledgments

Bardiya Aryanfard and Monika Henzinger were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101019564). For open access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission. Funded by the European union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Bardiya Aryanfard, Monika Henzinger, David Saulpic, and A. R. Sricharan. 2025. Improved Lower Bounds for Privacy under Continual Release. *CoRR* abs/2512.15981 (2025). arXiv:2512.15981 doi:10.48550/ARXIV.2512.15981

[2] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. 2014. Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds. In *Foundations of Computer Science (FOCS)*. doi:10.1109/FOCS.2014.56

[3] Omri Ben-Eliezer, Rajesh Jayaram, David P. Woodruff, and Eylon Yogev. 2022. A Framework for Adversarially Robust Streaming Algorithms. *J. ACM* 69, 2, Article 17 (Jan. 2022), 33 pages. doi:10.1145/3498334

[4] Rajendra Bhatia. 1997. *Symmetric Norms*. Springer New York, 84–111. doi:10.1007/978-1-4612-0653-8_4

[5] Ari Biswas, Graham Cormode, Yaron Kanza, Divesh Srivastava, and Zhengyi Zhou. 2024. Differentially private hierarchical heavy hitters. *Proceedings of the ACM on Management of Data* 2, 5 (November 2024). doi:10.1145/3695826

[6] Jaroslaw Blasiok, Vladimir Braverman, Stephen R. Chestnut, Robert Krauthgamer, and Lin F. Yang. 2017. Streaming symmetric norms via measure concentration. In *Symposium on Theory of Computing (STOC)*. doi:10.1145/3055399.3055424

[7] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2012. The Johnson-Lindenstrauss Transform Itself Preserves Differential Privacy. In *Foundations of Computer Science (FOCS)*. doi:10.1109/FOCS.2012.67

[8] Jeremiah Blocki, Elena Grigorescu, Tamalika Mukherjee, and Samson Zhou. 2023. How to Make Your Approximation Algorithm Private: A Black-Box Differentially-Private Transformation for Tunable Approximation Algorithms of Functions with Low Sensitivity. In *Randomization and Computation (RANDOM)*. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.59

[9] Greg Bodwin, Chengyuan Deng, Jie Gao, Gary Hoppenworth, Jalaj Upadhyay, and Chen Wang. 2024. The Discrepancy of Shortest Paths. In *International Colloquium on Automata, Languages, and Programming (ICALP)*. doi:10.4230/LIPICS.ICALP.2024.27

[10] Dan Boneh and James Shaw. 1998. Collusion-Secure Fingerprinting for Digital Data. *IEEE Trans. Inf. Theory* 44 (1998), 1897–1905. https://doi.org/10.1109/18.705568

[11] Vladimir Braverman, Joel Manning, Zhiwei Steven Wu, and Samson Zhou. 2023. Private Data Stream Analysis for Universal Symmetric Norm Estimation. In *Randomization and Computation (RANDOM)*. doi:10.4230/LIPICS.APPROX/RANDOM.2023.45

[12] Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Yin Tat Lee, Judy Hanwen Shen, and Uthaipon Tantipongpipat. 2021. Fast and Memory Efficient Differentially Private-SGD via JL Projections. In *Neural Information Processing Systems (NeurIPS)*. https://openreview.net/forum?id=WwZbupAKWo

[13] Mark Bun, Jonathan R. Ullman, and Salil P. Vadhan. 2018. Fingerprinting Codes and the Price of Approximate Differential Privacy. *SIAM J. Comput.* 47, 5 (2018), 1888–1938. doi:10.1137/15M1033587

[14] T.-H. Hubert Chan, Mingfei Li, Elaine Shi, and Wenchang Xu. 2012. Differentially Private Continual Monitoring of Heavy Hitters from Distributed Streams. In *Privacy Enhancing Technologies (PETS)*. doi:10.1007/978-3-642-31680-7_8

[15] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and Continual Release of Statistics. *ACM Trans. Inf. Syst. Secur.* 14, 3 (2011), 26:1–26:24. doi:10.1145/2043621.2043626

[16] Justin Y. Chen, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Shyam Narayanan, Jelani Nelson, and Yinzhan Xu. 2023. Differentially Private All-Pairs Shortest Path Distances: Improved Algorithms and Lower Bounds. In *Symposium on Discrete Algorithms (SODA)*. doi:10.1137/1.9781611977554.CH184

[17] Seung Geol Choi, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich. 2020. Differentially-Private Multi-Party Sketching for Large-Scale Statistics. *Proc. Priv. Enhancing Technol.* 2020, 3 (2020), 153–174. doi:10.2478/POPETS-2020-0047

[18] Mina Dalirrooyfard, Slobodan Mitrovic, and Yuriy Nevmyvaka. 2023. Nearly Tight Bounds For Differentially Private Multiway Cut. In *Neural Information Processing Systems (NeurIPS)*. http://papers.nips.cc/paper_files/paper/2023/hash/4e8f257e054abd24c550d55e57cec274-Abstract-Conference.html

[19] Anindya De. 2012. Lower Bounds in Differential Privacy. In *Theory of Cryptography Conference (TCC)*. doi:10.1007/978-3-642-28914-9_18

[20] 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 *Neural Information Processing Systems (NeurIPS)*. http://papers.nips.cc/paper_files/paper/2022/hash/271ec4d1a9ff5e6b81a6e21d38b1ba96-Abstract-Conference.html

[21] Laxman Dhulipala, Monika Henzinger, George Z. Li, Quanquan C. Liu, A. R. Sricharan, and Leqi Zhu. 2025. Near-Optimal Differentially Private Graph Algorithms via the Multidimensional AboveThreshold Mechanism. In *33rd Annual European Symposium on Algorithms, ESA 2025, Warsaw, Poland, September 15-17, 2025 (LIPIcs, Vol. 351)*, Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 91:1–91:20. doi:10.4230/LIPICS.ESA.2025.91

[22] Laxman Dhulipala, Quanquan C. Liu, Sofya Raskhodnikova, Jessica Shi, Julian Shun, and Shangdi Yu. 2022. Differential Privacy from Locally Adjustable Graph Algorithms: k-Core Decomposition, Low Out-Degree Ordering, and Densest Subgraphs. In *Foundations of Computer Science (FOCS)*. doi:10.1109/FOCS54457.2022.00077

[23] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Principles of Database Systems (PODS)*. doi:10.1145/773153.773173

[24] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference (TCC)*. doi:10.1007/11681878_14

[25] Cynthia Dwork, Frank McSherry, and Kunal Talwar. 2007. The price of privacy and the limits of LP decoding. In *Symposium on Theory of Computing (STOC)*. doi:10.1145/1250790.1250804

[26] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. 2010. Differential privacy under continual observation. In *Symposium on Theory of Computing (STOC)*. doi:10.1145/1806689.1806787

[27] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. 2010. Pan-Private Streaming Algorithms. In *Innovations in Theoretical Computer Science (ITCS)*. http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/6.html

[28] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Symposium on Theory of Computing (STOC)*. doi:10.1145/1536414.1536467

[29] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407. doi:10.1561/0400000042

[30] Marek Eliás, Michael Kapralov, Janardhan Kulkarni, and Yin Tat Lee. 2020. Differentially Private Release of Synthetic Graphs. In *Symposium on Discrete Algorithms (SODA)*. doi:10.1137/1.9781611975994.34

[31] Alessandro Epasto, Quanquan C. Liu, Tamalika Mukherjee, and Felix Zhou. 2025. Sublinear Space Graph Algorithms in the Continual Release Model. In *Randomization and Computation (RANDOM)*. doi:10.48550/ARXIV.2407.17619

[32] 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 *Innovations in Theoretical Computer Science (ITCS)*. doi:10.4230/LIPICS.ITCS.2023.48

[33] Chenglin Fan, Ping Li, and Xiaoyun Li. 2022. Private Graph All-Pairwise-Shortest-Path Distance Release with Improved Error Rate. In *Neural Information Processing Systems (NeurIPS)*. http://papers.nips.cc/paper_files/paper/2022/hash/71b17f00017da0d73823ccf7fbce2d4f-Abstract-Conference.html

[34] Alireza Farhadi, MohammadTaghi Hajiaghayi, and Elaine Shi. 2022. Differentially Private Densest Subgraph. In *Artificial Intelligence and Statistics (AISTATS)*. https://proceedings.mlr.press/v151/farhadi22a.html

[35] Hendrik Fichtenberger, Monika Henzinger, and Lara Ost. 2021. Differentially Private Algorithms for Graphs Under Continual Observation. In *European Symposium on Algorithms (ESA)*. doi:10.4230/LIPIcs.ESA.2021.42

[36] Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. 2023. Constant Matters: Fine-grained Error Bound on Differentially Private Continual Observation. In *International Conference on Machine Learning (ICML)*. https://proceedings.mlr.press/v202/fichtenberger23a.html

[37] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. 2010. Differentially Private Combinatorial Optimization. In *Symposium on Discrete Algorithms (SODA)*. doi:10.1137/1.9781611973075.90

[38] Anupam Gupta, Aaron Roth, and Jonathan R. Ullman. 2012. Iterative Constructions and Private Data Release. In *Theory of Cryptography Conference (TCC)*. doi:10.1007/978-3-642-28914-9_19

[39] Moritz Hardt and Kunal Talwar. 2010. On the geometry of differential privacy. In *Symposium on Theory of Computing (STOC)*. doi:10.1145/1806689.1806786

[40] Yizhang He, Kai Wang, Wenjie Zhang, Xuemin Lin, Ying Zhang, and Wei Ni. 2025. Robust Privacy-Preserving Triangle Counting under Edge Local Differential Privacy. *Proc. ACM Manag. Data* 3, 3, Article 211 (June 2025), 26 pages. doi:10.1145/3725348

[41] Monika Henzinger and Roodabeh Safavi. 2025. Securing Dynamic Data: A Primer on Differentially Private Data Structures (Invited Talk). In *33rd Annual European Symposium on Algorithms, ESA 2025, Warsaw, Poland, September 15-17, 2025 (LIPIcs, Vol. 351)*, Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2:1–2:14. doi:10.4230/LIPICS.ESA.2025.2

[42] Monika Henzinger, A. R. Sricharan, and Teresa Anna Steiner. 2024. Private Counting of Distinct Elements in the Turnstile Model and Extensions. In *Randomization and Computation (RANDOM)*. doi:10.4230/LIPICS.APPROX/RANDOM.2024.40

[43] 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, Vol. 258)*, Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Mohammad Emtiyaz Khan (Eds.). PMLR, 1990–1998. https://proceedings.mlr.press/v258/henzinger25a.html

[44] 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 *Neural Information Processing Systems (NeurIPS)*. http://papers.nips.cc/paper_files/paper/2023/hash/0ef1afa0daa888d695dcd5e9513bafa3-Abstract-Conference.html

[45] 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, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 14654–14678. https://proceedings.mlr.press/v202/jain23b.html

[46] 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)*. https://proceedings.mlr.press/

v202/jain23b.html
[47] Palak Jain, Adam Smith, and Connor Wagaman. 2024. Time-Aware Projections: Truly Node-Private Graph Statistics under Continual Observation. *CoRR* abs/2403.04630 (2024). arXiv:2403.04630 doi:10.48550/ARXIV.2403.04630
[48] Christian Janos Lebeda and Jakub Tetek. 2024. Better Differentially Private Approximate Histograms and Heavy Hitters using the Misra-Gries Sketch. *SIGMOD Rec.* 53, 1 (May 2024), 7–14. doi:10.1145/3665252.3665255
[49] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2011. What Can We Learn Privately? *SIAM J. Comput.* 40, 3 (2011), 793–826. doi:10.1137/090756090
[50] Mahmoud Abo Khamis, RelationalAI, Dan Olteanu, and Dan Suciu. 2023. Join Size Bounds using lp-Norms on Degree Sequences. *Proceedings of the ACM on Management of Data* 2 (2023), 1 – 24. https://api.semanticscholar.org/CorpusID:259252687
[51] Jingcheng Liu, Jalaj Upadhyay, and Zongrui Zou. 2024. Optimal Bounds on Private Graph Approximation. In *Symposium on Discrete Algorithms (SODA)*. doi:10.1137/1.9781611977912.39
[52] Darakhshan J. Mir, S. Muthukrishnan, Aleksandar Nikolov, and Rebecca N. Wright. 2011. Pan-private algorithms via statistics on sketches. In *Principles of Database Systems (PODS)*. doi:10.1145/1989284.1989290
[53] Pranay Mundra, Charalampos Papamanthou, Julian Shun, and Quanquan C. Liu. 2025. Practical and Accurate Local Edge Differentially Private Graph Algorithms. *Proc. VLDB Endow.* 18, 11 (July 2025), 4199–4213. doi:10.14778/3749646.3749687
[54] Dung Nguyen and Anil Vullikanti. 2021. Differentially Private Densest Subgraph Detection. In *International Conference on Machine Learning (ICML)*. http://proceedings.mlr.press/v139/nguyen21i.html
[55] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Symposium on Theory of Computing (STOC)*. doi:10.1145/1250790.1250803
[56] Rasmus Pagh, Lukas Retschmeier, Hao Wu, and Hanwen Zhang. 2025. Optimal Bounds for Private Minimum Spanning Trees via Input Perturbation. *Proc. ACM Manag. Data* 3, 2, Article 103 (June 2025), 26 pages. doi:10.1145/3725240
[57] Naty Peter, Eliad Tsfadia, and Jonathan R. Ullman. 2024. Smooth Lower Bounds for Differentially Private Algorithms via Padding-and-Permuting Fingerprinting Codes. In *Conference on Learning Theory (COLT)*. https://proceedings.mlr.press/v247/peter24a.html
[58] 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 (2025). arXiv:2506.08201 doi:10.48550/ARXIV.2506.08201
[59] Sofya Raskhodnikova and Teresa Anna Steiner. 2025. Fully Dynamic Algorithms for Graph Databases with Edge Differential Privacy. In *Principles of Database Systems (PODS)*. https://arxiv.org/abs/2409.17623
[60] Adam Sealfon. 2016. Shortest Paths and Distances with Differential Privacy. In *Principles of Database Systems (PODS)*. doi:10.1145/2902251.2902291
[61] Or Sheffet. 2017. Differentially Private Ordinary Least Squares. In *International Conference on Machine Learning (ICML)*. https://proceedings.mlr.press/v70/sheffet17a.html
[62] Adam D. Smith, Shuang Song, and Abhradeep Thakurta. 2020. The Flajolet-Martin Sketch Itself Preserves Differential Privacy: Private Counting with Minimal Space. In *Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper/2020/hash/e3019767b1b23f82883c9850356b71d6-Abstract.html
[63] Shuang Song, Susan Little, Sanjay Mehta, Staal Amund Vinterbo, and Kamalika Chaudhuri. 2018. Differentially Private Continual Release of Graph Statistics. *CoRR* abs/1809.02575 (2018). arXiv:1809.02575 http://arxiv.org/abs/1809.02575
[64] William Swartworth and David P. Woodruff. 2025. Finding Heavy-Hitters with Optimal State Changes. *Proc. ACM Manag. Data* 3, 5, Article 278 (Nov. 2025), 22 pages. doi:10.1145/3767714
[65] Gábor Tardos. 2008. Optimal probabilistic fingerprint codes. *J. ACM* 55 (2008), 10:1–10:24. https://api.semanticscholar.org/CorpusID:2175998
[66] Lun Wang, Iosif Pinelis, and Dawn Song. 2022. Differentially Private Fractional Frequency Moments Estimation with Polylogarithmic Space. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=7I8LPkcx8V
[67] Chen Zeng, Jeffrey F. Naughton, and Jin-Yi Cai. 2012. On differentially private frequent itemset mining. *Proc. VLDB Endow.* 6, 1 (Nov. 2012), 25–36. doi:10.14778/2428536.2428539
[68] Haozhe Zhang, Christoph Mayer, Mahmoud Abo Khamis, Dan Olteanu, and Dan Suciu. 2025. LpBound: Pessimistic Cardinality Estimation Using ℓ_p-Norms of Degree Sequences. *Proc. ACM Manag. Data* 3, 3, Article 184 (June 2025), 27 pages. doi:10.1145/3725321

Received December 2025; accepted February 2025