


Private Counting of Distinct Elements in the Turnstile Model and Extensions

Monika Henzinger  

Institute of Science and Technology, Klosterneuburg, Austria

A. R. Sricharan  

Faculty of Computer Science, Doctoral School Computer Science, University of Vienna, Austria

Teresa Anna Steiner  

Technical University of Denmark, Lyngby, Denmark

Abstract

Privately counting distinct elements in a stream is a fundamental data analysis problem with many applications in machine learning. In the turnstile model, Jain et al. [NeurIPS2023] initiated the study of this problem parameterized by the maximum flippancy of any element, i.e., the number of times that the count of an element changes from 0 to above 0 or vice versa. They give an item-level (ϵ, δ) -differentially private algorithm whose additive error is tight with respect to that parameterization. In this work, we show that a very simple algorithm based on the sparse vector technique achieves a tight additive error for item-level (ϵ, δ) -differential privacy and item-level ϵ -differential privacy with regards to a different parameterization, namely the sum of all flippancies. Our second result is a bound which shows that for a large class of algorithms, including all existing differentially private algorithms for this problem, the lower bound from item-level differential privacy extends to event-level differential privacy. This partially answers an open question by Jain et al. [NeurIPS2023].

2012 ACM Subject Classification Security and privacy

Keywords and phrases differential privacy, turnstile model, counting distinct elements

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.40

Category RANDOM

Related Version *Full Version*: <https://arxiv.org/abs/2408.11637>

Funding *Monika Henzinger*: This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (MoDynStruct, No. 101019564) and the Austrian Science Fund (FWF) grant DOI 10.55776/Z422, grant DOI 10.55776/I5982, and grant DOI 10.55776/P33775 with additional funding from the netidee SCIENCE Stiftung, 2020–2024.

Teresa Anna Steiner: Supported by a research grant (VIL51463) from VILLUM FONDEN.



1 Introduction

Counting distinct elements in a stream is a fundamental data analysis problem that is widely studied [12, 13, 17, 18, 20] and has many applications [1, 10, 19, 2, 21, 5], including network analysis [21] and detection of denial of service attacks [1, 5]. If the data includes sensitive information, the essential challenge is to give accurate answers while providing privacy guarantees to the data owners. Differential privacy is the de-facto standard in private data analysis and is widely employed both in research and in industry. In the insertions-only model, the problem of counting distinct elements while preserving differential privacy is well-studied [3, 9, 14].

Recent work by Jain, Kalemaj, Raskhodnikova, Sivakumar, and Smith [16] (which was concurrent with an earlier version of the results presented in this paper, see [15, Section 5]) initiated the study of this problem in the more general turnstile model. They give an algorithm which is *item-level*, (ϵ, δ) -differentially private and analyze the additive error



© Monika Henzinger, A. R. Sricharan, and Teresa Anna Steiner;
licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024).

Editors: Amit Kumar and Noga Ron-Zewi; Article No. 40; pp. 40:1–40:21



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

parameterized in the *maximum flippancy* of any element, i.e., the number of times that the count of an element changes from 0 to above 0 or vice versa. They also give lower bounds which show that the additive error of the algorithm is tight for item-level differential privacy (up to log factors) with respect to their parameterization. There is still a gap for event-level differential privacy, which is posed as an open question. The algorithm is based on several instantiations of the binary tree mechanism.

In this paper we show that a simple algorithm based on the sparse vector technique achieves a tight additive error (up to log factors) for item-level (ϵ, δ) -differential privacy and item-level ϵ -differential privacy, with regards to a different parameterization, namely the *total flippancy*, i.e., the sum of the flippancies of all elements. The additive error depends polynomially on the total flippancy with a smaller exponent than the exponent of the maximum flippancy in the additive error in [16]. Thus, if there are few elements in total, or few elements which change their count from 0 to above 0 or vice versa, then our algorithm achieves a better additive error. Additionally, we give a reduction which shows that for a large class of algorithms, including all existing differentially private algorithms for this problem, the lower bound from item-level differential privacy extends to event-level differential privacy. This is a step towards answering the open question posed in [16].

1.1 Problem Definition

More formally, we assume there are d different items, and our goal is to maintain a multiset of them and to determine at each time step how many of them are currently at least once in the multiset, i.e., the number of distinct elements in the multiset. The update operations are modeled as follows: The input at every time step is a d -dimensional vector $x^t \in \{-1, 0, 1\}^d$, such that $x_i^t = 1$ if element i gets inserted at time t , $x_i^t = -1$ if element i gets deleted at time t , and $x_i^t = 0$ otherwise. Note that this means that we allow *multiple non-zero entries* in x^t , corresponding to *multiple updates* at every time step. However, the lower bound also extends to the case where we assume that at most one element may be inserted or deleted at any time step, i.e., $\|x^t\|_1 \leq 1$, which we call *singleton update streams*. At every time step t , we want to output the number of distinct elements in the multiset. By our definition of the input stream, an element i is present at time t if and only if $\sum_{t' \leq t} x_i^{t'} > 0$.

► **Definition 1** (COUNTDISTINCT). *Let x^1, x^2, \dots, x^T be an input stream with $x^t \in \{-1, 0, 1\}^d$ for all $t \in [1, T]$. We define $\text{COUNTDISTINCT}(x)^t = \sum_{i=1}^d \mathbb{1}(\sum_{t' \leq t} x_i^{t'} > 0)$, where $\mathbb{1}(E)$ is the indicator function that is 1 if E is true and 0 otherwise. Then, the COUNTDISTINCT problem is to output $\text{COUNTDISTINCT}(x)^t$ at all time steps t . The error of COUNTDISTINCT is defined to be the maximum additive error over all time steps.*

In this paper, we consider two privacy notions: *event-level* differential privacy, and *item-level* differential privacy. They differ in their definition of neighboring input streams. Two input streams x and y are *event-level neighboring*, if there exists a time step t^* and an item $i^* \in [1, d]$ such that we have $x_i^t = y_i^t$ for all $(i, t) \neq (i^*, t^*)$. That is, two event-level neighboring streams may differ in *at most one item in at most one update operation*. Two input streams x and y are *item-level neighboring*, if there exists an item $i^* \in [1, d]$ such that $x_i^t = y_i^t$ for all t and for all $i \in [1, d] \setminus \{i^*\}$. That is, two item-level neighboring streams may differ in *all update operations related to one item*.

Finally, we consider two models regarding the input stream. In the *general model* the counts for any item at any time step t is given by $\sum_{t' \leq t} x_i^{t'}$, which can be any integer in $[-t, t]$ and we only care about whether $\sum_{t' \leq t} x_i^{t'}$ is larger than zero or not. In the *“likes”-model*¹

¹ The name was chosen as it models the count of “likes” on a social media website, as motivated by [16].

for every item i at any time step t , it must hold that $\sum_{t' \leq t} x_i^{t'} \in \{0, 1\}$, i.e., the multiset is a set. Said differently, an item can only be inserted if it is absent in the set and it can only be deleted when it is present.

1.2 Summary of Results

In this paper, we give new upper and lower bounds for *item-level* differential privacy, parameterized in the *total flippancy* K , which is defined as the total number of times any item switches from a non-zero count to a zero count, or vice versa. In detail, let $f^t(x_i) = \mathbb{1}(\sum_{t' < t} x_i^{t'} > 0)$. The total flippancy is formally defined as $K = \sum_{i=1}^d \sum_{t=2}^T \mathbb{1}(f^t(x_i) \neq f^{t-1}(x_i))$. Note that in the “likes”-model, the total flippancy is equal to the total number of updates. As $\text{COUNTDISTINCT}(x)^t = \sum_{i=1}^d f^t(x_i)$, it follows that K is an upper bound on the number of changes in $\text{COUNTDISTINCT}(x)$ over time.

Upper Bounds

As our first main result, we give algorithms solving COUNTDISTINCT while providing item-level differential privacy, which work in the general model (thus also in the “likes”-model). In the following, we state the exact bounds for *given* K . If K is not given to the algorithm, the error bounds worsen by at most a $\ln^2 K$ factor.

► **Theorem 2.** *Let d be a non-zero integer, $\beta > 0$, K be a known upper bound on the total flippancy, and let T be a known upper bound on the number of time steps. Then there exists*

1. *an item-level ϵ -differentially private algorithm for COUNTDISTINCT in the general model with additive error $O(\min(d, K, \sqrt{\epsilon^{-1} K \ln(T/\beta)}, \epsilon^{-1} T \log(T/\beta)))$ with probability at least $1 - \beta$ at all time steps simultaneously, for any $\epsilon > 0$ and $\beta \in (0, 1)$;*
2. *an item-level (ϵ, δ) -differentially private algorithm for COUNTDISTINCT in the general model with additive error $O(\min(d, K, (\epsilon^{-2} K \ln(1/\delta) \ln^2(T/\beta))^{1/3}, \epsilon^{-1} \sqrt{T \ln(1/\delta) \log(T/\beta)}))$ with probability at least $1 - \beta$ at all time steps simultaneously, for any $\delta \in (0, 1)$, $\epsilon \in (0, 1)$, and $\beta \in (0, 1)$.*

As our lower bounds (discussed below) show, our bounds for ϵ -differential privacy are *tight*, if K is known and $K \leq T$. If $K > T$, we incur at most an extra $\ln T$ factor, and if K is not known, we incur extra $\ln K$ factors (see Section 6). For (ϵ, δ) -differential privacy, the upper bounds are tight up to $\ln T$, $\ln K$ and $\ln(1/\delta)$ factors.

Lower bounds

We complement our upper bounds by almost tight lower bounds on the additive error which hold for any item-level differentially private algorithm in the “likes”-model. As this is the “more restricted” of the two models, the lower bounds also carry over to the general model. For ϵ -differential privacy, our lower bound follows from a packing argument.

► **Theorem 3** (Simplified version of Theorem 16). *For any $L \leq T$, there exists an input stream x of d -dimensional vectors from $\{-1, 0, 1\}^d$, which is valid in the “likes”-model, with length T and flippancy $K = \Theta(L)$, such that any item-level, ϵ -differentially private algorithm for COUNTDISTINCT must with constant probability have an error at least $\Omega(\min(d, K, \sqrt{\epsilon^{-1} K \max(\ln(T/K), 1)}))$.*

The lower bound above also holds for singleton updates. When multiple updates are allowed, then K could potentially be larger than T . In that case, Theorem 16 in Section 5 shows that for any $T \leq L \leq dT$, there exists a stream with flippancy $K = \Omega(T)$,

■ **Table 1** Comparison of our results (in blue) and the results in [16] for the different models. K denotes the total flippancy and w denotes the maximum flippancy of an input stream x . For simplicity of exposition, we consider singleton insertions and omit factors polynomial in $\ln T$, $\ln(1/\delta)$, and ϵ^{-1} . The bounds marked with * hold for *output dependent* algorithms (see the discussion before Theorem 5 for details). The bounds in the last line follow from a simple application of a continual counting algorithm on the difference sequence.

	Item-level ϵ -dp	Item-level (ϵ, δ) -dp	Event-level ϵ -dp	Event-level (ϵ, δ) -dp
general model [16]	$\Omega(\min(w, \sqrt{T}))$	$O(\min(\sqrt{w}, T^{1/3}))$ $\Omega(\min(\sqrt{w}, T^{1/3}))$		$O(\min(\sqrt{w}, T^{1/3}))$ $\Omega(\min(\sqrt{w}, T^{1/4}))$
“likes”-model [16]	$\Omega(\min(w, \sqrt{T}))$	$O(\min(\sqrt{w}, T^{1/3}))$ $\Omega(\min(\sqrt{w}, T^{1/3}))$		$O(\min(\sqrt{w}, T^{1/3}))$
general model this work	$O(\sqrt{K})$ $\Omega(\sqrt{K})$	$O(K^{1/3})$ $\Omega(K^{1/3})$	$O(\sqrt{K})$ $\Omega(\min(w, \sqrt{K}))^*$	$O(K^{1/3})$ $\Omega(\min(\sqrt{w}, K^{1/3}))^*$
“likes”-model this work	$O(\sqrt{K})$ $\Omega(\sqrt{K})$	$O(K^{1/3})$ $\Omega(K^{1/3})$	$O(\sqrt{K})$	$O(K^{1/3})$
“likes”-model			$O(1)$	$O(1)$

$K = O(L)$, such that any item-level, ϵ -differentially private algorithm to the COUNT-DISTINCT problem must have error at least $\Omega(\min(d, \epsilon^{-1}T, \sqrt{\epsilon^{-1}L \max(\ln(T/L), 1)})) = \Omega(\min(d, \epsilon^{-1}T, \sqrt{\epsilon^{-1}K \max(\ln(T/K), 1)}))$ with constant probability. For (ϵ, δ) -differential privacy, we can use a similar strategy as in [16] to get the following bounds:

► **Theorem 4** (Simplified version of Theorem 19). *Let $\epsilon, \delta \in (0, 1]$. Let K, T be sufficiently large parameters. There exists a dimension $d \in \mathbb{N}$ and an input stream x of d -dimensional vectors from $\{-1, 0, 1\}^d$ of length T with flippancy at most K which is valid in the “likes”-model, such that any item-level, (ϵ, δ) -differentially private algorithm for COUNTDISTINCT must have error at least $\Omega\left(\epsilon^{-1} \cdot \min\left(\frac{\sqrt{T}}{\log T}, \frac{(K\epsilon)^{1/3}}{\log(K\epsilon)}\right)\right)$ with constant probability.*

Note that this lower bound holds for the case where *multiple insertions* are allowed in every time step. In Theorem 19 we also give a lower bound of $\Omega\left(\frac{K^{1/3}}{\epsilon \log K}\right)$ for singleton-updates.

Time and Space Complexity

Our main algorithm (achieving the $O(\sqrt{\epsilon^{-1}K \ln T})$ error bound for ϵ -differential privacy and $O((K \ln(1/\delta) \ln^2 T)^{1/3}/\epsilon^{2/3})$ error bound for (ϵ, δ) -differential privacy) can be implemented using constant time per update, assuming that drawing from a Laplace distribution takes constant time. Specifically, the total running time is $O(\#\text{updates} + S_K t_{\text{Lap}})$, where t_{Lap} is the time to draw one Laplace random variable, $S_K = O(\sqrt{K\epsilon/\ln T} + 1)$ for ϵ -dp, and $S_K = O\left(\left(\frac{K\epsilon}{\sqrt{\ln(1/\delta) \ln(T/\beta)}}\right)^{2/3}\right)$ for (ϵ, δ) -dp. The algorithm uses $O(d)$ words of space. The only information our algorithm needs to store are the true counts for each item, plus a constant number of words of extra information. This holds even for the case where K is unknown, since we *sequentially* run our known K algorithm with increasing guesses for K .

Comparison to the recent work by Jain, Kalemaj, Raskhodnikova, Sivakumar, and Smith [16]

In recent work, [16] considered the COUNTDISTINCT problem with a similar, but different parameterization. In [16], they parameterize the additive error in the *maximum flippancy*, i.e., they parameterize on $w_x = \max_{i \in [d]} (\sum_{t=2}^T \mathbb{1}(f^t(x_i) \neq f^{t-1}(x_i)))$. Recall that K denotes the

total flippancy of a stream x and note that $w_x \leq K \leq d \cdot w_x$. [16] consider only streams with singleton updates and give algorithms for item-level, (ϵ, δ) -differential privacy in the general model, with an error bound of $\tilde{O}\left(\min\left(\left(\sqrt{w_x} \log T + \log^3 T\right) \cdot \frac{\sqrt{\log(1/\delta)}}{\epsilon}, \frac{(T \log(1/\delta))^{1/3}}{\epsilon^{2/3}}, T\right)\right)^2$.

In comparison, our bounds in this setting are $\tilde{O}\left(\min\left(\frac{(K \ln(1/\delta) \ln^2 T)^{1/3}}{\epsilon^{2/3}}, K\right)\right)$. Note that $K \leq T$ for singleton updates, and thus, our upper bounds recover their second and third bound up to a $\ln^{2/3} T$ factor. Furthermore, ignoring polynomial factors in $\log T$, $\log(1/\delta)$ and ϵ^{-1} , their bound is $O(\sqrt{w_x})$ while ours is $O(K^{1/3})$. Thus, if (roughly) $K < w_x^{3/2}$, our algorithm outperforms theirs. Specifically, if $d \leq \sqrt{w_x}$ or if there are only few items with high flippancy, we expect our algorithm to do better. In cases where the flippancy is well-distributed, i.e., many items have a similar flippancy, and $d \geq \sqrt{w_x}$, we expect the algorithm in [16] to perform better.

In terms of space and time complexity, their algorithm, like ours, needs to maintain a count for each element. Thus, the space in terms of words is $\Omega(d)$. On top of that, they run a variant of the binary tree mechanism, which depending on the implementation, uses $\Omega(\log T)$ space. In their final solution, they actually run $\log T$ copies of the binary tree mechanism in parallel, bringing their space consumption to $O(d + \log^2 T)$ words. Thus, the space of our algorithm is an additive $\log^2 T$ term better, which can be crucial for large streams. In terms of time complexity, each of the binary tree mechanism needs to draw $\Omega(T \log T)$ independent Laplace noises, thus their time complexity is at least $\Omega(T \log^2 T t_{\text{Lap}})$, where t_{Lap} is the time it takes to draw a Laplace noise. Also here, our algorithm is more efficient.

In terms of lower bounds, for item-level, ϵ -differential privacy in the “likes”-model, [16] give a lower bound of $\Omega(\min(\epsilon^{-1}w, \sqrt{\epsilon^{-1}T}, T))$ for streams of maximum flippancy at most w . For (ϵ, δ) -differential privacy, they give a lower bound of $\tilde{\Omega}(\min(\epsilon^{-1}\sqrt{w}, \epsilon^{-2/3}T^{1/3}, T))$ for item-level privacy in the “likes”-model, and $\tilde{\Omega}(\min(\epsilon^{-1}\sqrt{w}, \epsilon^{-3/4}T^{1/4}, T))$ for event-level privacy in the general model, for streams of maximum flippancy at most w . Their upper bounds in the item-level setting match their lower bounds up to factors polynomial in $\log T$ and $\log(1/\delta)$. For event-level in the general model, there is a gap for $\sqrt{T} \leq w \leq T^{2/3}$, and closing this gap was posed as an explicit open question in [16]. As our second main result, we make a step towards closing this gap, which we explain below.

Reduction from item-level, “likes”-model to output-dependent event-level, general model

All upper bounds mentioned so far hold for *item-level* differential privacy. As our upper bounds hold in the general model and our lower bounds hold in the “likes”-model, we can conclude that for item-level privacy, the “likes”-model and the general model are roughly equally hard. [16] arrived at this conclusion as well, albeit with a different parameterization.

However, for *event-level* differential privacy, the picture is different: for the “likes”-model, a very simple algorithm gives an error of $O(\epsilon^{-1} \text{polylog}(T))$ with constant probability. To see this, define the *difference sequence* for the COUNTDISTINCT problem as $\text{diff}^t(x) = \text{COUNTDISTINCT}(x)^t - \text{COUNTDISTINCT}(x)^{t-1}$ for $t > 1$. As can be easily seen, $(\text{diff}^t(x))_{t>1}$ and $(\text{diff}^t(y))_{t>1}$ differ by at most 1 in at most one time step t for any event-level neighboring streams x and y in the “likes”-model. Thus, applying a standard continual counting algorithm gives the claimed error, as shown for “well-behaved” difference sequences in general in [11].

² For simplicity of exposition, we use $\tilde{O}(X)$ to denote $O(X \cdot \text{polylog}(X))$

For event-level differential privacy and the *general model* however, the best known algorithms are the algorithms for item-level differential privacy in this paper and [16]. [16] also present lower bounds for event-level differential privacy in the general model which, however, leave a gap for certain parameter settings. Closing that gap was explicitly posed as an open question in [16]. We make a step towards closing that gap, by noting that all existing differentially private algorithms for the COUNTDISTINCT problem in *any* model share the following property: If $\text{COUNTDISTINCT}(x) = \text{COUNTDISTINCT}(y)$ for any two input streams x and y , then the output distributions of the algorithms are equal. That is, any two streams which produce the same true output, will have the same output distributions. We call such algorithms *output-determined*. We show that if we only consider output-determined algorithms for COUNTDISTINCT, then achieving *event-level differential privacy in the general model is just as hard as item-level differential privacy for the “likes”-model*. Thus our above lower bounds also apply to such algorithms. In particular, this shows that if one were trying to close the gap for event-level differential privacy in the general model, one needs to find an algorithm which does not only depend on the true answers to COUNTDISTINCT.

► **Theorem 5** (Simplified version of Theorem 15). *Let $\epsilon > 0$ and $\delta \geq 0$. Let A_1 be an event-level, (ϵ, δ) -differentially private, output-determined algorithm for COUNTDISTINCT that works in the general model and has error at most α for streams of length $T + 1$ with probability $1 - \beta$. Then there exists an item-level, $(2\epsilon, (1 + e^\epsilon)\delta)$ -differentially private algorithm A_2 for COUNTDISTINCT that works in the “likes”-model, and has error at most α for streams of length T with probability $1 - \beta$.*

Generalizations & Applications

While our algorithms are (nearly) tight for the COUNTDISTINCT problem, they are not tailored specifically to the problem and work in a more general setting as well. In particular, recall that $\text{COUNTDISTINCT}(x)^t = \sum_{i=1}^d f^t(x_i)$, where $f^t(x_i) = \mathbb{1}(\sum_{t' \leq t} x_i^{t'} > 0)$. Now consider any real-valued function Q on input streams x_1, x_2, \dots , with $x_i \in \{-1, 0, 1\}$. We use $Q^t(x)$ to denote $Q(x_1, \dots, x_t)$. Our algorithm works for any such function Q such that the following two conditions are fulfilled: (1) for any x and y which are neighboring, we have $|Q^t(x) - Q^t(y)| \leq 1$ for all time steps t , and (2) $\sum_{t=1}^T |Q^t(x) - Q^{t-1}(x)| \leq K$.

► **Theorem 6.** *Let Q be a function satisfying properties (1) and (2). Then there exists*

1. *an item-level ϵ -differentially private algorithm for computing Q with additive error*

$$O(\min(K, \sqrt{\epsilon^{-1} K \ln(T/\beta)}), \epsilon^{-1} T \log(T/\beta))$$

at all time steps with probability at least $1 - \beta$, for any $\epsilon > 0$;

2. *an item-level (ϵ, δ) -differentially private algorithm for computing Q with additive error*

$$O(\min(K, (\epsilon^{-2} K \ln(1/\delta) \ln^2(T/\beta))^{1/3}, \epsilon^{-1} \sqrt{T \ln(1/\delta) \log(T/\beta)}))$$

at all time steps with probability at least $1 - \beta$, for any $\epsilon > 0$, $\delta \in (0, 1)$;

The extension to unknown K also holds, with extra $\ln K$ factors as earlier. Thus, for a continuous function Q which has *maximum* sensitivity 1 *over all time steps*, we get a bound parameterized in the sum of all differences, i.e., the L_1 -norm of the difference sequence. While our results hold in the *turnstile model* and the additive error is parameterized by the total flippancy, [11] gave an ϵ -differentially private mechanism with additive error $O(\Gamma \log^{3/2} \log(T/\beta))$ in the *insertions-only* or *deletions-only* setting, where Γ is the *continuous global sensitivity* which is the L_1 -norm of the difference sequence of two neighboring inputs.

We can apply our algorithm to the problem considered in Fichtenberger et al. [11] of continuously counting high degree nodes under differential privacy, which counts the number of nodes with degree at least τ , where τ is given and public. For user-level, edge-differential privacy (i.e., neighboring streams may differ in all updates of the same edge), they give a lower bound of $\Omega(n)$. Our algorithm gives new parameterized bounds for this problem: In particular, choosing $Q^t(x) = \frac{\# \text{ of high degree nodes}}{2}$, Theorem 6 gives an error bound of roughly $O(\sqrt{K})$, under ϵ -differential privacy, and roughly $O(K^{1/3})$ under (ϵ, δ) -differential privacy, where we ignore an $O(\epsilon^{-1} \ln T \ln(1/\delta) \ln K)$ factor. Note that K can be as large as T , but for many applications, it could be much smaller: for example, in social networks, it has been shown that the degree distribution follows a power-law distribution, which implies that the set of high-degree nodes only changes infrequently. K does not have to be given to the algorithm.

1.3 Algorithm Overview

The main idea of our algorithm is to use the sparse vector technique first introduced by Dwork, Naor, Reingold, Rothblum, and Vadhan [7] (the form we use it in can be found in Dwork and Roth [8]) on carefully chosen queries and with carefully chosen thresholds. The sparse vector technique can be described as follows: It is given a data set x , a sequence of q queries, a threshold $Thresh$, and a stopping parameter S . It will process these queries sequentially, and for each of them answer “yes” or “no” depending on whether or not $q(x)$ is approximately (up to an additive error α) above the threshold. It stops after it has answered “yes” S times. Dwork and Roth [8] show that it is possible to design an ϵ -differentially private algorithm achieving the above with $\alpha = O(\epsilon^{-1} S \log(q/\beta))$ with probability $1 - \beta$, and an (ϵ, δ) -differentially private algorithm with $\alpha = O(\epsilon^{-1} \sqrt{S \log(1/\delta)} \log(q/\beta))$ with probability $1 - \beta$. In the following discussion, we ignore ϵ^{-1} , $\log(1/\delta)$, $\log q$ and $\log(1/\beta)$ factors.

Our main idea is to note that the total flippancy K can be seen as an upper bound on *the total change in the output*, i.e., the sum of the absolute differences in the output in every time step. Our strategy is as follows: We start by estimating the number of distinct elements at the beginning of the stream. Then, we keep reporting this estimate until a significant change occurs in the true number of distinct elements. We track whether such a change has occurred using the sparse vector technique. Once there has been a significant change, i.e., once the sparse vector technique answers “yes”, we update the output. The goal now is to balance the additive error of the sparse vector technique with the error accumulated between updates. The error between updates is roughly $Thresh$; the error of the sparse vector technique is α ; and the total change of the output is bounded by K . To balance the two we set $Thresh = \Theta(\alpha)$. Furthermore we have to choose S in a way that makes sure that the sparse vector technique does not abort before we have seen the entire stream. We can show that every time our sparse vector technique answers “yes”, the change in output has been roughly $Thresh$. Thus it is enough to set $S > K/Thresh$. As mentioned above, for ϵ -differential privacy α (and, thus, $Thresh$) must depend linearly on S , which implies that S must be chosen to be $\Theta(\sqrt{K})$, giving an additive error of $O(\sqrt{K})$. For (ϵ, δ) -differential privacy, we have $Thresh = \Theta(\alpha) = O(\sqrt{S})$. This implies that $S^{3/2}$ must be $\Theta(K)$, i.e., $S = \Theta(K^{2/3})$. Thus the additive error is $O(K^{1/3})$.

Note that this requires that K is known at the beginning of the algorithm. If K is unknown, we run the above algorithm for exponentially increasing guesses of K ($K = 2, 4, 8$, etc.). In particular, we run the algorithm for a guess of K , and if it terminates preemptively, we double our guess and repeat. Since we do not know beforehand how many instances are needed, in order to make sure the resulting algorithm is still ϵ -differentially private, we run the j th instance with privacy parameter $\epsilon_j = O(\epsilon/j^2)$, such that $\sum_{j=1}^{\infty} \epsilon_j \leq \epsilon$. At the end of the algorithm, $j = \Theta(\ln K)$, therefore we incur an extra $\ln^2 K$ factor in the additive error.

2 Preliminaries

We denote $\{1, \dots, n\}$ by $[n]$ and the input stream length by T , the number of time steps.

Continual observation algorithm

An algorithm A in the continual observation model gets an update at every time step $t \leq T$, and produces an output $a^t = A(x^1, \dots, x^t)$ which is a function of x^1 to x^t ; $A^T(x) = (a^1, a^2, \dots, a^T)$ denotes the sequence of outputs at all time steps.

► **Definition 7** (Differential privacy [6]). *A randomized algorithm A is (ϵ, δ) -differentially private $((\epsilon, \delta)$ -dp) if for all $S \in \text{range}(A^T)$ and all x, y neighboring*

$$\Pr[A^T(x) \in S] \leq e^\epsilon \Pr[A^T(y) \in S] + \delta.$$

If $\delta = 0$ then A is ϵ -differentially private (ϵ -dp).

► **Definition 8** (Laplace Distribution). *The Laplace distribution centered at 0 with scale b is the distribution with probability density function $f_{\text{Lap}}(b)(x) = (2b)^{-1} \cdot \exp(-|x|/b)$. We use $X \sim \text{Lap}(b)$ or just $\text{Lap}(b)$ to denote a random variable X distributed according to $f_{\text{Lap}}(b)(x)$.*

In our definitions below, we use χ to represent a generic universe of elements.

► **Definition 9** (Sensitivity). *Let $f : \chi \rightarrow \mathbb{R}^k$. The L_p -sensitivity Δ_p is defined as*

$$\max_{x \in \chi, y \in \chi, x \sim y} \|f(x) - f(y)\|_p,$$

where $x \sim y$ denotes that x and y are neighbouring.

► **Fact 1** (Theorem 3.6 in [8]: Laplace Mechanism). *Let f be any function $f : \chi \rightarrow \mathbb{R}^k$ with L_1 -sensitivity Δ_1 . Let $Y_i \sim \text{Lap}(\Delta_1/\epsilon)$ for $i \in [k]$. The mechanism defined as $A(x) = f(x) + (Y_1, \dots, Y_k)$ satisfies ϵ -differential privacy.*

► **Fact 2** (Laplace Tailbound). *If $X \sim \text{Lap}(b)$, then $\Pr[|X| \geq t \cdot b] \leq e^{-t}$.*

The following fact follows from Theorem A.1 in [8]:

► **Fact 3** (Gaussian Mechanism). *Let f be any function $f : \chi \rightarrow \mathbb{R}^k$ with L_2 -sensitivity Δ_2 . Let $Y_i \sim \mathcal{N}(0, \sigma^2)$ for $i \in [k]$, where $\sigma \geq \sqrt{2 \ln(2/\delta)} \Delta_2/\epsilon$. The mechanism defined as $A(x) = f(x) + (Y_1, \dots, Y_k)$ satisfies (ϵ, δ) -differential privacy.*

► **Fact 4** (Gaussian tailbound). *If $X \sim \mathcal{N}(0, \sigma^2)$, then $\Pr[|X| \geq \sigma \sqrt{\ln(2/\beta)}] \leq \beta$*

The following facts are respectively given by Theorem 3.16, 3.20 and Corollary 3.21 in [8].

► **Fact 5** (Composition Theorem). *Let A_1 be an (ϵ_1, δ_1) -differentially private algorithm $A_1 : \chi \rightarrow \text{range}(A_1)$ and A_2 an (ϵ_2, δ_2) -differentially private algorithm $A_2 : \chi \times \text{range}(A_1) \rightarrow \text{range}(A_2)$. Then $B : \chi \rightarrow \text{range}(A_1) \times \text{range}(A_2)$ defined as $B(x) = (A_1(x), A_2(x, A_1(x)))$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -differentially private.*

► **Fact 6** (Advanced Composition). *Let $\epsilon, \delta, \delta' \geq 0$. Let A_1 be an (ϵ, δ) -differentially private algorithm $A_1 : \chi \rightarrow \text{range}(A_1)$ and A_i be (ϵ, δ) -differentially private algorithms $A_i : \chi \times \text{range}(A_{i-1}) \rightarrow \text{range}(A_i)$, for $2 \leq i \leq k$. Then the composition $B : \chi \rightarrow \text{range}(A_1) \times \dots \times \text{range}(A_k)$ defined as $B(x) = (A_1(x), A_2(x, A_1(x)), \dots, A_k(x, A_{k-1}(x)))$ is $(\epsilon', k\delta + \delta')$ -differentially private, where $\epsilon' = \sqrt{2k \ln(1/\delta')} \epsilon + k\epsilon(e^\epsilon - 1)$.*

► **Corollary 10**. *Let $\epsilon^*, \delta, \delta' \geq 0$ and $\delta', \epsilon^* < 1$. Let A_1, \dots, A_k be as in Fact 6 with*

$$\epsilon = \epsilon^*/(2\sqrt{2k \ln(1/\delta')}).$$

Then the composition B (defined as in Fact 6) is $(\epsilon^, k\delta + \delta')$ -differentially private.*

3 Item-Level Algorithms in General Model

In this section, we give algorithms which work for any input sequence in the general model, and thus also for input sequences that fulfill the conditions of the “likes”-model. The upper bounds on the additive error for ϵ -differential privacy match the lower bounds in Section 5, except for the $\log(T/\beta)$ factor in the case where $K > T$.

3.1 Known Total Flippancy

We prove Theorem 11 in this section. We give some intuition first on Algorithm 1. The algorithm works by iteratively checking if the true number of distinct elements currently present (called Q) is “far” from the current output of our algorithm (called out) using a sparse vector technique (SVT) instantiation. We start the algorithm by estimating out at the beginning of the stream (line 8). Then, we keep outputting out , while we track the difference between out and the true number of distinct elements Q (line 14). Once there has been a significant change, we update the output (line 18).

There are two parameters of interest here. One is the number of times we update the output: we abort after S_K updates happen (line 21). The other is the parameter $Thresh$, which determines how large the current error needs to be such that we satisfy the condition in line 14. The parameter S_K goes into the error from composition, while the parameter $Thresh$ directly goes into the additive error bound.

The goal is to balance the error accumulated between updates (which is roughly $Thresh$), and the error from updating out privately (which is roughly S_K for ϵ -differential privacy, and roughly $\sqrt{S_K}$ for (ϵ, δ) -differential privacy due to composition). Additionally, we want to make sure our algorithm does not abort before having processed the entire stream. We show that every time SVT returns “yes”, the total flippancy in the stream has increased by at least $\Omega(Thresh)$. Since we know the total flippancy is bounded by K , in order to make sure that we do not abort preemptively, we choose S_K such that $S_K \cdot Thresh \approx K$. Balancing the two error terms yields an additive error of approximately \sqrt{K} for ϵ -differential privacy, and $K^{1/3}$ for (ϵ, δ) -differential privacy.

► **Theorem 11.** *Let d and T be non-zero integers, let $\beta > 0$, and let K be an upper bound on the total flippancy which is given. Let T be a known upper bound on the number of time steps. Then there exists*

1. *an item-level ϵ -differentially private algorithm for COUNTDISTINCT in the general model with error at most $O(\min(d, K, \sqrt{\epsilon^{-1}K \ln(T/\beta)}, \epsilon^{-1}T \log(T/\beta)))$ at all time steps with probability at least $1 - \beta$ for $\epsilon > 0$;*
2. *an item-level (ϵ, δ) -differentially private algorithm for COUNTDISTINCT in the general model with error $O\left(\min(d, K, (\epsilon^{-2}K \ln(1/\delta) \ln^2(T/\beta))^{1/3}, \epsilon^{-1}\sqrt{T \ln(1/\delta) \log(T/\beta)})\right)$ at all time steps with probability at least $1 - \beta$, for any $0 < \delta < 1$ and $0 < \epsilon < 1$.*

Proof. The $O(\min(d, K))$ bound follows from the fact that the algorithm that outputs 0 at every time step is ϵ -differentially private and has error at most $\min(d, K)$ for any ϵ . The third error bounds in the minimum for Theorem 11 are achieved by Algorithm 1, as shown below. Since we assume here all parameters are known, one can compute the minimum of the three bounds and choose the algorithm accordingly. The fourth bound in Theorem 11 follow by a direct application of the Laplace mechanism Fact 1 with $\Delta_1 = T$ resp. Gaussian mechanism Fact 3 with $\Delta_2 = \sqrt{T}$.

■ **Algorithm 1** COUNTDISTINCT, known K .

```

1: Input: Data Stream  $x = x^1, x^2, \dots$ , initial counts  $c_1, \dots, c_d$  (default 0), parameters  $\epsilon, \delta$ 
   and  $\beta$ , stream length bound  $T$ , stopping parameter  $S_K \geq 1$ 
2: if  $\delta = 0$  then  $\epsilon_1 = \epsilon/(2S_K)$ 
3: if  $\delta > 0$  then  $\epsilon_1 = \epsilon/(4\sqrt{2S_K \ln(1/\delta)})$ 
4: count = 1
5:  $\tau_1 = \text{Lap}(2/\epsilon_1)$ 
6:  $\nu_1 = \text{Lap}(1/\epsilon_1)$ 
7:  $Q = 0$ 
8: out =  $Q + \nu_1$ 
9: Thresh =  $16\epsilon_1^{-1}(\ln(2T/\beta))$ 
10: for  $t = 1, \dots$ , do
11:    $c_i = c_i + x_i^t$  for all  $i \in [d]$ 
12:    $Q = |\{i \in [d] \mid c_i > 0\}|$ 
13:    $\mu_t = \text{Lap}(4/\epsilon_1)$ 
14:   if  $|\text{out} - Q| + \mu_t > \text{Thresh} + \tau_{\text{count}}$  then
15:     count = count + 1
16:      $\tau_{\text{count}} = \text{Lap}(2/\epsilon_1)$ 
17:      $\nu_{\text{count}} = \text{Lap}(1/\epsilon_1)$ 
18:     out =  $Q + \nu_{\text{count}}$ 
19:   end if
20:   output out
21:   if count  $\geq S_K$  then Abort
22: end for

```

The algorithm for our third bound, given in Algorithm 1, is based on the sparse vector technique, where S_K is a parameter dependent on K that we choose suitably below. We omit the proof of the following lemma, since it follows from well-known techniques (Sparse Vector Technique [7, 8], Laplace mechanism (Fact 1) and composition theorems (Facts 5 and 6)).

► **Lemma 12.** *For $\delta = 0$ and any $\epsilon > 0$, Algorithm 1 is ϵ -differentially private. For $0 < \epsilon < 1$ and $0 < \delta < 1$, Algorithm 1 is (ϵ, δ) -differentially private.*

We show the claimed accuracy bound using the following lemma.

► **Lemma 13.** *For $\delta = 0$, for any time step t before the algorithm aborts, we have that the maximum error up to time t is at most $O(\epsilon^{-1}S_K \ln(T/\beta))$. Setting $S_K = \sqrt{K\epsilon/(18 \ln(T/\beta))} + 1$, with probability at least $1 - \beta$, Algorithm 1 does not abort before having seen the entire stream, and has error at most $O(\sqrt{\epsilon^{-1}K \ln(T/\beta)} + \epsilon^{-1} \ln(T/\beta))$. For $\delta > 0$, for any time step t before the algorithm aborts, we have that the maximum error up to time t is $O(\epsilon^{-1}\sqrt{S_K \ln(1/\delta)} \ln(T/\beta))$. Setting $S_K = \left(\frac{K\epsilon}{36\sqrt{\ln(1/\delta)} \ln(T/\beta)}\right)^{2/3} + 1$, with probability at least $1 - \beta$, Algorithm 1 does not abort before having seen the entire stream, and has error at most $O\left(\left(\epsilon^{-2}K \ln(1/\delta) \ln^2(T/\beta)\right)^{1/3} + \epsilon^{-1}\sqrt{\ln(1/\delta)} \ln(T/\beta)\right)$.*

Proof. Note that at every time step t in Algorithm 1, we set $Q = \sum_{i=1}^d f^t(x_i)$. Let $\alpha = (8/\epsilon_1) \ln(2T/\beta) = (1/2) \cdot \text{Thresh}$. By Laplace tailbounds (Fact 2), at every time step t :
(a) $|\tau_\ell| \leq (2/\epsilon_1) \ln(2T/\beta) = \alpha/4$ with probability at least $1 - \beta/(2T)$, where ℓ is the value of variable count at time step t , and

(b) $|\mu_\ell| \leq (4/\epsilon_1) \ln(2T/\beta) = \alpha/2$ with probability at least $1 - \beta/(2T)$.

Thus, with probability $\geq 1 - \beta$, we have at all time steps t simultaneously:

- (i) Whenever the condition in line 14 is true at time t , then $|\text{out} - \sum_{i \in [d]} f^t(x_i)| > \text{Thresh} - 3\alpha/4 = 5\alpha/4$, and
- (ii) Whenever the condition in line 14 is false at time t , then $|\text{out} - \sum_{i \in [d]} f^t(x_i)| \leq \text{Thresh} + 3\alpha/4 < 3\alpha$.

Further, the random variable ν_ℓ for $\ell \in [S_K]$ is distributed as $\text{Lap}(1/\epsilon_1)$ and is added to $\sum_{i \in [d]} f^t(x_i)$ at every time step t where out is updated. By the Laplace tail bound (Fact 2), ν_ℓ is bounded for all $\ell \in [S_K]$ by $\epsilon_1^{-1} \ln(S_K/\beta) \leq \alpha/8$ with probability at least $1 - \beta$. Altogether, all of these bounds hold simultaneously with probability at least $1 - 2\beta$. We condition on all these bounds being true.

Assume the algorithm has not terminated yet at time t and let out be the value of variable out at the beginning of time t . Let p_ℓ be the last time step at which the value of out was updated. It holds that $|\text{out} - \sum_{i \in [d]} f_i^{p_\ell}(x)| = |\nu_\ell| \leq \alpha/8$. If the condition in line 14 is true at time t , then

$$\left| \sum_{i \in [d]} f_i^{p_\ell}(x) - \sum_{i \in [d]} f^t(x_i) \right| \geq \left| \sum_{i \in [d]} f^t(x_i) - \text{out} \right| - \left| \text{out} - \sum_{i \in [d]} f_i^{p_\ell}(x) \right| \geq 5\alpha/4 - \alpha/8 = 9\alpha/8.$$

Thus, between two time steps where the value of out is updated, there is a change of at least $9\alpha/8$ in the sum value, i.e., the value of $f^t(x_i)$ has changed at least once for $\geq 9\alpha/8$ different items i . Since $K = \sum_{i=1}^d \sum_{t=2}^T \mathbb{1}(f^t(x_i) \neq f^{t-1}(x_i))$, to guarantee (under the noise conditions), that the algorithm does not terminate before we have seen the entire stream, it suffices to choose S_K where $S_K > K/(9\alpha/8)$.

For $\delta = 0$, we have $\alpha = (8/\epsilon_1) \ln(2T/\beta) = (16S_K/\epsilon) \ln(2T/\beta)$, thus we have to choose $S_K > K\epsilon/(18S_K \ln(2T/\beta))$. Choosing $S_K = \lfloor \sqrt{K\epsilon/(18 \ln(2T/\beta))} \rfloor + 1$ fulfills this condition.

Similarly, for $\delta > 0$, choosing $S_K = \left(\frac{K\epsilon}{36\sqrt{\ln(1/\delta)} \ln(T/\beta)} \right)^{2/3} + 1$ fulfills this condition.

Now consider any time step t and let out be the output at time t , i.e., the value after processing time step t . If the condition in line 14 is false, we showed above that $|\text{out} - \sum_{i \in [d]} f^t(x_i)| < 3\alpha$. If the condition is true at time t , we have $\text{out} = \sum_{i \in [d]} f^t(x_i) + \nu_\ell$ for some $\ell \in [S_K]$, and, thus, $|\text{out} - \sum_{i \in [d]} f^t(x_i)| \leq \alpha/8 < \alpha$.

For $\delta = 0$, we have $\alpha = (8/\epsilon_1) \ln(2T/\beta) = O(\sqrt{\epsilon^{-1}K \ln(T/\beta)} \ln(T/\beta) + \epsilon^{-1} \ln(T/\beta))$.

Plugging in $S_K = \left(\frac{K\epsilon}{36\sqrt{\ln(1/\delta)} \ln(T/\beta)} \right)^{2/3} + 1$ yields the final bound for $\delta > 0$. ◀

To finish the proof of Theorem 11, note that if $\epsilon^{-1} \ln(T/\beta) > \sqrt{\epsilon^{-1}K \ln(T/\beta)}$, then $\sqrt{\epsilon^{-1}K \ln(T/\beta)} > K$, which can be seen by multiplying both sides of the inequality with $\sqrt{K}/\sqrt{\epsilon^{-1} \ln(T/\beta)}$. Thus the upper bound $\min(d, K, \sqrt{\epsilon^{-1}K \ln(T/\beta)})$ holds for $\delta = 0$.

Also, if $\epsilon^{-1} \sqrt{\ln(1/\delta)} \ln(T/\beta) > (\epsilon^{-2}K \ln(1/\delta) \ln^2(T/\beta))^{1/3}$, then $\epsilon^{-1} \sqrt{\ln(1/\delta)} \ln(T/\beta) > K$, which can be seen by first cubing the inequality and then dividing by $\epsilon^{-2} \ln(1/\delta) \ln^2(T/\beta)$. Thus, for $\delta > 0$, the upper bound of $\min(d, K, (\epsilon^{-2}K \ln(1/\delta) \ln^2(T/\beta))^{1/3})$ holds. ◀

3.2 Generalizations

We now argue about Theorem 6. Let Q be a real-valued function on input streams from $\{-1, 0, 1\}$ and let $Q^t = Q(x_1, \dots, x_t)$. Further, let Q be such that 1.) for any x and y which are neighboring, we have $|Q^t(x) - Q^t(y)| \leq 1$ for all time steps t , and 2.) $\sum_{t=1}^T |Q^t(x) - Q^{t-1}(x)| \leq$

K . The first bound from Theorem 6 is achieved by an algorithm that never updates the output, and the third bounds for ϵ and (ϵ, δ) -differential privacy are obtained by the Laplace and Gaussian mechanisms, respectively. The second bound for both ϵ and (ϵ, δ) -differential privacy is obtained by Algorithm 1 by setting $Q = Q^t(x)$ at every time step t . The proofs follow by exchanging $\sum_{i \in [d]} f^t(x_i)$ by $Q^t(x)$ in the proofs of Lemma 12 and 13.

4 A Connection between the General Model under Event-Level Privacy and the “Likes”-Model under Item-Level Privacy

Our bounds from Theorems 2, 3, and 4 as well as the bounds from [16] imply that under *item-level privacy*, the “likes”-model and the general model are roughly equally hard: all upper bounds hold for the general model and all lower bounds hold for the “likes”-model, and the bounds are tight up to a $\log T$ factor. However, under *event-level privacy*, the “likes”-model is significantly easier than the general model: It can be solved via continual counting on the difference sequence of the true output, which gives error polylogarithmic in $\log T$. This is possible because for event-level privacy in the “likes”-model, the difference sequence of the output (i.e., the difference between the true output value of the current and the preceding time step) has ℓ_∞ -sensitivity 1 for event-level privacy, but for item-level privacy, the sensitivity can be as large as T .

In the general model, there are no better upper bounds known for event-level differential privacy than for item-level differential privacy, and the upper and lower bounds from [16] for (ϵ, δ) -differential privacy for the event-level setting in the general model leave a polynomial (in T) gap, in the case where the maximum flippancy $w_x \in (T^{1/2}, T^{2/3})$: In that case, ignoring polynomial factors in ϵ^{-1} , $\log(1/\delta)$, and $\log T$, the lower bound of [16] is $\Omega(T^{1/4})$, while their algorithm gives an additive error of $O(T^{1/3})$. Specifically, finding the best achievable error for *event-level privacy* in the general model is explicitly posed as an open question in [16].

We resolve this question for a large class of algorithms, called γ -*output-determined* algorithms. All known algorithms for this problem in *any* model are 0-output-determined. Specifically, we show that for γ -output-determined algorithms our lower bounds and the lower bounds from [16] for *item-level privacy* in the “likes”-model basically carry over to *event-level privacy* in the *general model*. It follows that our algorithm and the algorithm from [16] for event-level privacy in the general model are tight up to a factor that is linear in $\log T$ *within the class of output-determined algorithms*. Note that our reduction works both for the ϵ -differential privacy as well as for (ϵ, δ) -differential privacy and we give the corresponding lower bounds in Theorems 16 and 19. In the following, we denote by $\text{COUNTDISTINCT}(x)$ the stream of true answers to the COUNTDISTINCT problem on stream x .

► **Definition 14.** Let $\gamma \geq 0$. An algorithm \mathcal{A} for the COUNTDISTINCT problem is said to be γ -output-determined, if for all inputs x and y such that $\text{COUNTDISTINCT}(x) = \text{COUNTDISTINCT}(y)$ and any $S \in \text{range}(\mathcal{A})$ we have:

$$\Pr(\mathcal{A}(x) \in S) \leq \Pr(\mathcal{A}(y) \in S) + \gamma$$

► **Theorem 15.** Let $\epsilon > 0, \delta \geq 0$ and $\gamma \geq 0$. Let \mathcal{A}_1 be an event-level, (ϵ, δ) -differentially private, γ -output-determined algorithm for COUNTDISTINCT that works in the general model and has error at most α for streams of length $T+1$ with probability $1-\beta$. Then there exists an item-level, $(2\epsilon, (1+e^\epsilon)\delta + e^\epsilon\gamma)$ -differentially private algorithm \mathcal{A}_2 for COUNTDISTINCT that works in the “likes”-model, and has error at most α for streams of length T with probability $1-\beta$.

Proof. We describe algorithm \mathcal{A}_2 , that is item-level $(2\epsilon, (1 + e^\epsilon)\delta + e^\epsilon\gamma)$ -dp in the “likes”-model, derived from a γ -output-determined algorithm \mathcal{A}_1 which is event-level, (ϵ, δ) -dp in the general model: Let x be an input for COUNTDISTINCT in the “likes”-model of length T , i.e., x is such that $\sum_{t' \leq t} x_i^{t'}$ can only take the values 0 or 1, for any $i \in [d]$ and $t \in [T]$. Let $x_0 = 0^d x$, i.e., we attach a d -dimensional all-zero vector before x , and define $(\mathcal{A}_2(x))^t = (\mathcal{A}_1(x_0))^{t+1}$ for all $t \in [T]$ (note that \mathcal{A}_1 can take inputs from the “likes”-model). We now show that \mathcal{A}_2 is item-level $(2\epsilon, (1 + e^\epsilon)\delta + e^\epsilon\gamma)$ -differentially private. Let x and y be two item-level neighbouring inputs in the “likes”-model. That is, there exists an item i such that the streams x_i and y_i may be completely different, while $x_j = y_j$ for all $j \neq i$. Additionally, since we are in the “likes”-model, for any time step t , $\sum_{t' \leq t} x_i^{t'} \in \{0, 1\}$ and $\sum_{t' \leq t} y_i^{t'} \in \{0, 1\}$.

Next, we define input streams z and w in the general model where $\text{COUNTDISTINCT}(z) = \text{COUNTDISTINCT}(w)$, z is event-level neighbouring to x_0 , and w is event-level neighbouring to y_0 . Since \mathcal{A}_1 is event-level (ϵ, δ) -dp and works for the general model, we then have for any $S \in \text{range}(\mathcal{A}_2)$

$$\begin{aligned} \Pr[\mathcal{A}_2(x) \in S] &= \Pr[(\mathcal{A}_1(x_0))_{t=2}^{T+1} \in S] \leq e^\epsilon \Pr[(\mathcal{A}_1(z))_{t=2}^{T+1} \in S] + \delta \\ &\leq e^\epsilon \Pr[(\mathcal{A}_1(w))_{t=2}^{T+1} \in S] + \delta + \gamma \\ &\leq e^{2\epsilon} \Pr[(\mathcal{A}_1(y_0))_{t=2}^{T+1} \in S] + (1 + e^\epsilon)\delta + e^\epsilon\gamma \\ &= e^{2\epsilon} \Pr[\mathcal{A}_2(y) \in S] + (1 + e^\epsilon)\delta + e^\epsilon\gamma, \end{aligned}$$

where the second inequality holds as \mathcal{A}_1 is γ -output-determined.

To define such z and w , let $-e_i$ be the vector such that $-e_i(j) = 0$ for all $j \neq i$ and $-e_i(i) = -1$. Then $z = -e_i x$ and $w = -e_i y$. Note that z and w are valid input streams for the general model, while they are not valid for the “likes”-model. Clearly, z is event-level neighbouring to x_0 , and w is event level neighbouring to y . Recall that $\text{COUNTDISTINCT}(z)^t = \sum_{j=1}^d \mathbb{1}(\sum_{t' \leq t} z_j^{t'} > 0)$. Since $\sum_{t' \leq t} x_i^{t'} \in \{0, 1\}$ for all $t \in [T]$ we have $\sum_{t' \leq t} z_i^{t'} \leq 0$ for all $t \in [T + 1]$. By the same argument, we have $\sum_{t' \leq t} w_i^{t'} \leq 0$ for all $t \in [T + 1]$. Since z and w only differ in the i th coordinate, which never contributes to the COUNTDISTINCT value as it is never 1, we have $\text{COUNTDISTINCT}(z) = \text{COUNTDISTINCT}(w)$.

We are left with analyzing the error of the two algorithms. For this, note that by definition of x_0 , we have $\text{COUNTDISTINCT}(x_0)^{t+1} = \text{COUNTDISTINCT}(x)^t$. Thus, running \mathcal{A}_2 on x gives the same error as running \mathcal{A}_1 on x_0 . \blacktriangleleft

In particular, for any output-determined algorithm, Theorem 15 implies that all lower bounds on the error for the COUNTDISTINCT problem under *item-level* differential privacy which hold for the “likes”-model (and thus, all lower bounds for COUNTDISTINCT under item-level differential privacy shown in this paper in Theorem 19 and in [16]), carry over to *event-level* differential privacy in the *general model*. This means that if there is an algorithm achieving a better error than the bounds stated in Theorem 19 and in [16] for event-level differential privacy in the general model, it cannot be γ -output-determined for $\gamma = O(\delta)$, i.e., it must be such that it does not *only* depend on the number of distinct elements at any given time step.

5 Item-Level Lower Bounds in the “Likes”-Model

In the following we show lower bounds for solving COUNTDISTINCT under item-level differential privacy, and in the “likes”-model. The lower bounds also apply to the general model. In Section 3, we showed a complementing upper bound which holds in the general model, even if K is unknown to the algorithm.

► **Theorem 16.** *Let d and $T > 4$ be non-negative integers and let $\epsilon > 0$.*

1. *Let $L \geq 8$ be a non-negative integer such that $L \leq dT$. There exists an input stream x of d -dimensional vectors from $\{-1, 0, 1\}^d$, which is valid in the “likes”-model with multiple updates per time step, with length T and flippancy K with $\min(3L/8, T/4 - 1) \leq K \leq \min(L, dT/4)$ such that any ϵ -differentially private algorithm to the COUNTDISTINCT problem with item-level privacy with error at most α at all time steps with probability at least $2/3$ must satisfy*

$$\begin{aligned} \alpha &= \Omega(\min(d, L, \epsilon^{-1}T, \sqrt{\epsilon^{-1}L \max(\ln(T/L), 1)})) \\ &= \Omega(\min(d, K, \epsilon^{-1}T, \sqrt{\epsilon^{-1}K \max(\ln(T/K), 1)}). \end{aligned}$$

2. *Let $L \geq 8$ be a non-negative integer such that $L \leq T$. There exists an input stream x of d -dimensional vectors from $\{-1, 0, 1\}^d$, which is valid in the “likes”-model with multiple updates per time step, with length T , flippancy K with $L/16 \leq K \leq \min(L, T/4)$, and with $\|x^t\|_1 = 1$ for all t (i.e., each update modifies at most one item) such that any ϵ -differentially private algorithm to the COUNTDISTINCT problem with item-level privacy with error at most α at all time steps with probability at least $2/3$ must satisfy*

$$\alpha = \Omega(\min(d, K, \sqrt{\epsilon^{-1}K \ln(T/K)}).$$

Proof. Let d , T , and L be as given in the theorem statement. Assume there is an ϵ -differentially private algorithm \mathcal{A} for the COUNTDISTINCT problem with error at most α at all time steps with probability at least $2/3$. If $\alpha > d/2$, then the error is $\Omega(d)$. Also, if $\alpha > L/8$, then $\alpha = \Omega(L)$. Thus, in the following, we consider the case $\alpha \leq d/2$ and $\alpha \leq L/8$. Defining $m = \lfloor 2\alpha \rfloor$, it follows that $m \leq \min(d, L/8)$.

Singleton updates. We first find $T' \leq T$ and $L' \leq L$ such that $4m$ divides T' and m divides L' . If this is not the case for T and L , then pick parameters T' and L' such that (i) $4m$ divides T' and m divides L' , (ii) $\Delta = T - T' \leq 4m < L/2 \leq T/2$ (i.e. $T' = \Theta(T)$) and (iii) $0 \leq L - \Delta - L' \leq m$. This implies that $L' \geq 7L/8 - \Delta \geq 3L/8$. Thus, as $L \leq T$, then $0 \leq L - \Delta - L' = L - (T - T') - L' = T' - L' - (T - L) \leq T' - L'$, i.e., $L' \leq T'$.

We use T' and L' in the proof below to construct a sequence of length T' fulfilling the statements of the theorem. To complete the proof of the theorem, we append to the sequence $T - T'$ many all-zero vectors to guarantee that the stream has length T . Note that appending to the sequence “blank” operation will not invalidate the statements of the theorem.

We now construct a set of input sequences of length T' with flippancy $K := \min(L', T'/4)$ and use them to prove a lower bound for α of $\Omega(\min(K \ln(T'/K), \sqrt{\epsilon^{-1}K \ln(T'/K)}))$. Combined with the above case distinctions giving lower bounds on α of $\Omega(d)$, and $\Omega(L)$, the fact that $K = \Theta(L)$ and that $T' = \Theta(T)$, this implies that $\alpha = \Omega(\min(d, K, \sqrt{\epsilon^{-1}K(\ln(T'/K) + 1)}))$.

Let $k := \min(L', T'/4)/m$ be a positive integer. Partition the timeline into T'/m blocks of length m , namely $B_1 = [1, m]$, $B_2 = [m + 1, 2m]$, \dots . Now, for any subset of blocks $J = (j_1, \dots, j_k)$ with $1 \leq j_1 < j_2 < \dots < j_k \leq T'/m$, define an input sequence $x(J)$ such that for any item $i \in [m]$ we insert element i in the i th time step of every odd block of J (i.e. the first, third, \dots block in J), and delete it again at the i th position of every even block of J (i.e. the second, fourth, \dots block in J). More formally, for any item $i \in [m]$, set $x(J)_i^t = 1$ for all $t = B_{j_{2p-1}}[i] = (j_{2p-1} - 1)m + i$, $p = 1 \dots, \lceil k/2 \rceil$, and set $x(J)_i^t = -1$ for all $t = B_{j_{2p}} = (j_{2p} - 1)m + i$, $p = 1 \dots, \lceil k/2 \rceil$. In all other time steps t , no updates are performed, i.e., $x(J)^t$ is an all-zero vector. Thus, for every $i \in [m]$, we have $f^t(x_i) = 1$

for all time steps $t \in [j_{2p-1}m, (j_{2p} - 1)m]$, for all $p \leq \lceil k/2 \rceil$, and $f^t(x_i) = 0$ for all time steps $t \in [j_{2p}m, (j_{2p+1} - 1)m]$. For any item $m < i \leq d$, we have $f^t(x_i) = 0$ for all $t \in [T']$. Furthermore, items i with $i > m$ (if they exist) are never inserted or deleted. In total, there are $k = \min(L', T'/4)/m$ updates per item $i \in [m]$, thus exactly K updates in total, and, hence, the total flippancy is $K = \min(L', T'/4)$. If $K = L'$, then $L \geq K \geq 3L/8$. If $K = T'/4$, then $L' \leq T'$ implies that $L \geq L' \geq K = T'/4 \geq L'/4 \geq 3L/32 \geq L/16$. Thus in either case $K = \Theta(L')$. Furthermore $K \leq T'/4 \leq T/4$.

Now let E_J , for $J = (j_1, \dots, j_k)$ with $1 \leq j_1 < j_2 < \dots < j_k \leq T'/m$, be the set of output sequences where \mathcal{A} outputs (i) a value of $m/2$ or larger for all time steps $t \in [j_{2p-1}m, (j_{2p} - 1)m]$ with $1 \leq p \leq \lceil k/2 \rceil$, and (ii) smaller than $m/2$ for all time steps t such that (a) $t < j_1m$ or (b) $t \in [j_{2p}m, (j_{2p+1} - 1)m]$ for some $0 \leq p < \lceil k/2 \rceil$ or (c) $t \geq j_km$. Note that for an input sequence $x(J)$ every output sequence where \mathcal{A} has additive error smaller than $\alpha = m/2$ must belong to E_J . As the algorithm is correct with probability at least $2/3$, $\Pr[\mathcal{A}(x(J)) \in E_J] \geq 2/3$.

Two input sequences are neighboring if they differ in the data of at most one item for item-level differential privacy. As two input sequences $x(I)$ and $x(J)$ with $I \neq J$ differ in the data of at most m items, it follows by group privacy that $\Pr[\mathcal{A}(x(J)) \in E_I] \geq e^{-m\epsilon}2/3$ for any $J = (j_1, \dots, j_k)$ with $1 \leq j_1 < j_2 < \dots < j_k \leq T'/m$ and $I = (i_1, \dots, i_k)$ with $1 \leq i_1 < i_2 < \dots < i_k \leq T'/m$. Also note that the set of output sequences E_J for distinct $J = (j_1, \dots, j_k)$ are disjoint, since for each multiple of m (i.e., the end of a block), it is clearly defined whether the output is at least $m/2$ or smaller than $m/2$, and as such the values j_1, \dots, j_k can be uniquely recovered. Thus, there are $\binom{T'/m}{k}$ disjoint events E_J and the sum over all J of the probabilities that the algorithm with input $x(I)$ outputs an event E_J is at most 1. More formally, we have:

$$1 \geq \binom{T'/m}{k} e^{-m\epsilon}2/3 \geq \frac{(T'/m)^k}{(k)^k} e^{-m\epsilon}2/3 = \frac{T'^{(K/m)}}{K^{K/m}} e^{-m\epsilon}2/3$$

where the last equality is since $k = K/m$. This gives

$$m^2 + \epsilon^{-1}m \ln(3/2) \geq \epsilon^{-1}K \ln(T'/K)$$

which implies

$$m = \Omega(\min(K \ln(T'/K), \sqrt{\epsilon^{-1}K \ln(T'/K)}).$$

Note that since $T' \geq 4K$, $\ln(T'/K) \geq \ln(4) > 1$. This completes the proof.

Multiple updates. We first find $T' \leq T$ and $L' \leq L$ such that 4 divides T' and m divides L' . If this is not the case for T and L , then pick parameters T' and L' such that (i) 4 divides T' and m divides L' , (ii) $\Delta = T - T' \leq 4$ (i.e. $T' = \Theta(T)$) and (iii) $\Delta m \leq L - L' \leq (\Delta + 1)m$. This implies that $L' \geq L - (\Delta + 1)m \geq L - 5m \geq 3L/8$.

We use T' and L' in the proof below to construct a sequence of length T' fulfilling the statements of the theorem. To complete the proof of the theorem, we append to the sequence $T - T'$ many all-zero vectors to guarantee that the stream has length T . Note that appending to the sequence “blank” operation will not invalidate the statements of the theorem.

The idea is similar to above, only we do not define blocks, but directly choose $k := \min(L'/m, T'/4)$ time steps in which all items in $[m]$ are updated. Thus the flippancy K will equal mk . More precisely, we construct the following set of input sequences. For any $I = (t_1, \dots, t_k)$ with $1 \leq t_1 < t_2 < \dots < t_k \leq T'$, we define an input sequence $x(I)$ as follows: For any item $i \in [m]$, set $x(I)_i^{t_j} = 1$ for all odd j , and $x(I)_i^{t_j} = -1$ for all even j .

40:16 Private Counting of Distinct Elements in the Turnstile Model and Extensions

All other coordinates are set to 0. In total, there are k updates per item in $[m]$, thus, exactly K updates in total, i.e., the total flippancy equals $K = \min(L', mT'/4)$. This implies that $\min(3L/8, T/4 - 1) \leq K \leq \min(L, dT/4)$.

Now, let E_I , for $I = (t_1, \dots, t_k)$ with $1 \leq t_1 < t_2 < \dots < t_k \leq T'$, be the set of output sequences with a value of $m/2$ or larger at all time steps $t \in [t_{2p-1}, t_{2p})$ for some $1 \leq p \leq \lceil k/2 \rceil$, and a value smaller than $m/2$ at all time steps t where (a) $t \leq t_1$ or (b) $t \in [t_{2p}, t_{2p+1})$ for some $0 \leq p < \lceil k/2 \rceil$. Note that for input sequence $x(I)$ every output sequence where \mathcal{A} has an additive error smaller than $m/2$ must be in E_I . As the algorithm is correct with probability at least $2/3$, $\Pr[\mathcal{A}(x(I)) \in E_I] \geq 2/3$. As two input sequences $x(I)$ and $x(J)$ with $I \neq J = (j_1, \dots, j_k)$ with $1 \leq j_1 < j_2 < \dots < j_k \leq T'$ differ in the data of at most m items, it follows by group privacy that $\Pr[\mathcal{A}(x(I)) \in E_J] \geq e^{-m\epsilon} 2/3$ for any such J .

Let $J = (j_1, \dots, j_k)$ with $1 \leq j_1 < j_2 < \dots < j_k \leq T'$. Note that the events E_I and E_J for any $I \neq J$ are disjoint, since in the event E_I it is clearly defined for every time step whether the output is at least $m/2$ or smaller than $m/2$, and from that the set I can be uniquely recovered. Thus, there are $\binom{T'}{k}$ disjoint events E_J and the probability that with input $x(I)$ the algorithm outputs any one of them is at most 1. Thus we have

$$1 \geq \binom{T'}{k} e^{-m\epsilon} 2/3 \geq \frac{T'^k}{k^k} e^{-m\epsilon} 2/3 = \frac{T'^{K/m}}{(K/m)^{K/m}} e^{-m\epsilon} 2/3 \quad (1)$$

where the last equality is since $k = K/m$.

Next we consider two cases, the first one resulting in two different lower bounds on m and the second one giving a third lower bound on m . The combination of these three lower bounds then gives the claimed bound above of

$$\alpha = m/2 = \Omega(\min(\epsilon^{-1}T', \sqrt{\epsilon^{-1}K \max(\ln(T'/K), 1)}, K \max(\ln(T'/K), 1)))$$

Case 1. $L' < mT'/4$. In this case $K = L'$ and we have

$$m^2\epsilon + m \ln(3/2) \geq K \ln(T'm/K) \geq K \max(\ln(T'/K), 1)$$

where the last inequality holds since $K \leq mT'/4$, i.e., $\ln(T'm/K) \geq \ln(4) > 1$. Hence

$$m = \Omega(\min(\sqrt{\epsilon^{-1}K \max(\ln(T'/K), 1)}, K \max(\ln(T'/K), 1))).$$

As $K = L' = \Theta(L)$ it follows that

$$m = \Omega(\min(\sqrt{\epsilon^{-1}L \max(\ln(T'/L), 1)}, L \max(\ln(T'/L), 1))).$$

Case 2. $L' \geq mT'/4$. This implies that $K = mT'/4$ and, thus, that there are updates in at least $T'/4$ many time steps. In this case Inequality 1 can be reformulated as follows:

$$1 \geq \frac{T'^{K/m}}{K/m^{K/m}} e^{-m\epsilon} 2/3 = 4^{T'/4} e^{-m\epsilon} 2/3 = e^{\ln(4)T'/4 - m\epsilon} 2/3,$$

which implies that Inequality 1 is satisfied for $m = \Omega(\epsilon^{-1}T')$.

These two cases show $\alpha = \Omega(\min(\epsilon^{-1}T', \sqrt{\epsilon^{-1}L \max(\ln(T'/L), 1)}, L \max(\ln(T'/L), 1)))$ for the above input sequence. Combined with the above lower bounds on α of $\Omega(\min(d, L))$ and the fact that $T' = \Theta(T)$, it follows that $\alpha = \Omega(\min(d, L, \epsilon^{-1}T, \sqrt{\epsilon^{-1}L \max(\ln(T/L), 1)}))$. ◀

■ **Algorithm 2** COUNTDISTINCT, unknown K .

1: **Input:** Data Set $D = x^1, x^2, \dots$, initial counts c_1, \dots, c_d (default 0), parameters ϵ, δ and β, T
2: $t = 1$
3: $K_1 = 2$
4: **for** $j = 1, \dots$, **do**
5: $\epsilon_j = 6\epsilon/(\pi^2 j^2)$
6: $\delta_j = 6\delta/(\pi^2 j^2)$
7: $\beta_j = 6\beta/(\pi^2 j^2)$
8: **if** $\delta = 0$ **then**
9: $S_{K_j} = \sqrt{K_j \epsilon_j / (18 \ln(T/\beta_j))} + 1$
10: **end if**
11: **if** $\delta > 0$ **then**
12: $S_{K_j} = \left(\frac{K_j \epsilon_j}{36 \sqrt{\ln(1/\delta_j)} \ln(T/\beta_j)} \right)^{2/3} + 1$
13: **end if**
14: Run Algorithm 1 on $x^t, x^{t+1}, \dots, c_1, \dots, c_d, \epsilon_j, \delta_j, \beta_j, T, S_{K_j}$ until it aborts
15: Let t' be the last time step processed by Algorithm 1
16: $t = t' + 1$
17: $j = j + 1$
18: $K_j = 2^j$
19: **end for**

6 Unknown Total Flippancy

The algorithms from Section 3 can be easily extended to the case where the total flippancy K is not known beforehand, at the cost of $\text{polylog}(K)$ factors in the error bound, as shown by Algorithm 2 and the lemmata below. The fact that K is not known causes no serious problem, as the algorithm repeatedly “guesses” K and then runs the algorithm from earlier with the current guess.

► **Lemma 17.** *For any $0 < \epsilon < 1$ and $0 \leq \delta < 1$, Algorithm 2 is (ϵ, δ) -differentially private.*

Proof. By Lemma 12, the j th instance of Algorithm 1 is (ϵ_j, δ_j) -differentially private. Since $\sum_{j=1}^{\infty} \epsilon_j = \epsilon$ and $\sum_{j=1}^{\infty} \delta_j = \delta$, by Fact 5, Algorithm 2 is (ϵ, δ) -differentially private. ◀

► **Lemma 18.** *For $\delta = 0$, the error of Algorithm 2 is at most*

$$O(\ln K \sqrt{\epsilon^{-1} K \ln(T \ln K/\beta)} + \epsilon^{-1} \ln^2 K \ln(T \ln K/\beta)).$$

For $\delta > 0$, the error of Algorithm 2 is at most

$$O((\epsilon^{-1} K \ln^2 K \ln(\ln K/\delta) \ln^2(T \ln K/\beta))^{1/3} + \epsilon^{-1} \ln^2 K \sqrt{\ln(\ln K/\delta)} \ln(T \ln K/\beta)).$$

Proof. Let j_l be the value of variable j after the last element in the stream is processed. For any $j < j_l$, note that by Lemma 13, with probability at least $1 - \beta_j$, by the choice of S_{K_j} , the algorithm does not abort before having seen the entire stream if the total flippancy is at most K_j . Thus, when the algorithm aborts for some $j < j_l$, we know that the flippancy is at least K_j , and the bound from Lemma 13 holds for the j th instance of Algorithm 1 with S_{K_j} .

Since the algorithm aborts for all $j < j_l$, we can conclude that the total flippancy of the stream processed by the j th run of Algorithm 1 is at least K_j . Since $\sum_j \beta_j = \beta$, with probability at least $1 - \beta$, (1) the total flippancy K is at least $\sum_{j < j_l} K_j = 2^{j_l} - 1$, and

(2) the bound from Lemma 13 holds for all instances of Algorithm 1 (with their respective parameters). It follows (a) that $K \geq K_{j_i} - 1 \geq K_j$ for all $j < j_i$ and (b) $j_i = O(\ln K)$. The maximum error over the stream is the maximum error of any instance of Algorithm 1. Since $K_j = O(K)$, $\epsilon_{j_i} \leq \epsilon_j$ and $\delta_{j_i} \leq \delta_j$ for all $j \leq j_i$, the final bound is now obtained by plugging K , $\epsilon_{j_i} = \Theta(\epsilon/j^2)$ for ϵ , $\delta_{j_i} = \Theta(\delta/j^2)$ for δ , and $\beta_{j_i} = \Theta(\beta/j^2)$ for β into the bound from Lemma 13, and upper bounding j^2 by $\log^2 K$. ◀

One can also obtain the minimum of the bound from Lemma 18 and $\min(K, T, d)$ at the cost of an additive $\epsilon \ln^2 K \ln(\ln K/\beta)$ factor with a slightly more involved algorithm, which involves choosing to either not update the output or abort if there is a trivial algorithm which performs better for the current estimate of K . If we knew the value of K beforehand, we could choose the best algorithm upfront. Not knowing the value of K makes it slightly more complicated. However, the algorithm and analysis are fairly straightforward, and we defer it to the full version.

7 Lower Bounds for Approximate Differential Privacy

In this section, we adapt the lower bounds from [16] for item-level differential privacy to our parameter scheme.

► **Theorem 19.** *Let $\epsilon, \delta \in (0, 1]$.*

1. *Let K, T be sufficiently large parameters. There exists a dimension d and an input stream x of d -dimensional vectors from $\{-1, 0, 1\}^d$ of length T and with flippancy at most K which is valid in the “likes”-model, such that any item-level, (ϵ, δ) -differentially private algorithm to the COUNTDISTINCT problem with error at most α at all time steps with probability at least 0.99 must satisfy $\alpha = \Omega\left(\min\left(\frac{\sqrt{T}}{\epsilon \log T}, \frac{(K\epsilon)^{1/3}}{\epsilon \log(K\epsilon)}\right)\right)$.*
2. *Let K and T be sufficiently large parameters satisfying $K \leq T$. There exists a dimension d and an input stream x of d -dimensional vectors from $\{-1, 0, 1\}^d$ of length T and with flippancy at most K which is valid in the “likes”-model and satisfies $\|x^t\|_1 = 1$ for all t , such that any item-level, (ϵ, δ) -differentially private algorithm to the COUNTDISTINCT problem with error at most α at all time steps with probability at least 0.99 must satisfy $\alpha = \Omega\left(\frac{K^{1/3}}{\epsilon \log K}\right)$.*

The reduction in [16] is based on a lower bound for the 1-way marginals problem. In that problem, the data set y is an table consisting of n rows and m columns, where every entry is in $\{0, 1\}$. Two data sets y and y' are neighbouring if they differ in at most one row. The goal is to estimate the average column sums, i.e., the vector $(\sum_{i=1}^n y[i, j])_{j \in [m]}$. The following lower bound holds for estimating 1-way marginals under (ϵ, δ) -differential privacy:

► **Lemma 20** (Bun, Ullman, and Vadhan [4]). *Let $\epsilon \in (0, 1]$, $\gamma \in (0, 1)$, and $m, n \in \mathbb{N}$, and $\delta = o(1/n)$. Any algorithm which is (ϵ, δ) -differential private and has error at most γ with probability at least 0.99 satisfies $n = \Omega\left(\frac{\sqrt{m}}{\gamma \epsilon \log m}\right)$.*

Proof Sketch of Theorem 19. We start by arguing about Item 2. For this case, our example stream is exactly the same as in [16], given in Algorithm 5 in [16] (for a formulation using our slightly different notation see Algorithm 3). They give a reduction from the 1-way marginals problem: For any instance \mathcal{I} of the 1-way marginals problem with n rows and m columns, there is an instance $C(\mathcal{I})$ of COUNTDISTINCT with $T = 2mn$, such that if \mathcal{I} and \mathcal{I}' are neighbouring, then $C(\mathcal{I})$ and $C(\mathcal{I}')$ are item-neighbouring. Further, if we can solve $C(\mathcal{I})$ within error α , we can solve \mathcal{I} within error α/n . It follows by Lemma 20 that $\alpha = \Omega\left(\min\left(\frac{\sqrt{m}}{\epsilon \log m}, n\right)\right)$. In the instance they constructed, $d = n$, i.e. each row in the 1-way

marginals problem gives an item in the COUNTDISTINCT problem. Further, the total flippancy K can be as large as $2mn$ for worst case inputs. Thus, in order to apply the reduction, we need $2mn \leq K \leq T$. Given parameters $K \leq T$, we choose $m = K/(2n)$. The lower bound $\Omega\left(\min\left(\frac{\sqrt{m}}{\epsilon \log m}, n\right)\right)$ translates to $\Omega\left(\min\left(\frac{\sqrt{K/(2n)}}{\epsilon \log(K/(2n))}, n\right)\right)$. For $n = \frac{K^{1/3}}{2(\epsilon \log K)^{2/3}}$, we have

$$\frac{\sqrt{K/(2n)}}{\epsilon \log(K/(2n))} \geq \frac{K^{1/3} \epsilon^{1/3} \log^{1/3} K}{\epsilon \log(K^{1/2})} = \Omega\left(\frac{K^{1/3} \log^{1/3} K}{\epsilon^{2/3} \log K}\right) = \Omega(n).$$

Thus, we get $\alpha = \Omega\left(\frac{K^{1/3} \log^{1/3} K}{\epsilon^{2/3} \log K}\right)$.

For Item 1., where we allow general updates, we have to slightly modify the example in [16]: namely, in their Algorithm 5, we collapse every one of their vectors $z^{(j)}$, $j = 1, \dots, m$, into vectors of length 2, one time step for all insertions corresponding to column j , and one time step for all deletions corresponding to column j . See Algorithm 4. We then again get a reduction with the same properties as before, except that $T = 2n$ and K can be as large as $2mn$. Now, the analysis from [16] can be repeated with our T taking the role of w_x in [16], and our K taking the role of T in [16]. ◀

■ **Algorithm 3** Algorithm 5 from [16]: Reduction from 1-way marginals to COUNTDISTINCT.

```

1: Input: Data Set  $y[1], \dots, y[n] \in \{0, 1\}^{n \times m}$  and blackbox access to a mechanism  $M$  for
   COUNTDISTINCT
2: Output: Estimates of marginals  $b = (b[1], \dots, b[m])$ 
3: for  $j = 1, \dots, m$  do
4:   for  $i = 1, \dots, n$  do
5:     Set  $z^{(j)}[i] = e_i$ 
6:     Set  $z^{(j)}[i + n] = -e_i$ 
7:   end for
8: end for
9: Run  $M$  on  $x \rightarrow z^{(1)} \circ z^{(2)} \circ \dots \circ z^{(m)}$  and record answer vector  $r$ 
10: for  $j \in [m]$  do do
11:    $b[j] = r[(2j - 1)n]/n$ 
12: end for
13: output  $b$ 

```

■ **Algorithm 4** Reduction from 1-way marginals to COUNTDISTINCT for arbitrarily many updates per round.

```

1: Input: Data Set  $y[1], \dots, y[n] \in \{0, 1\}^{n \times m}$  and blackbox access to a mechanism  $M$  for
   COUNTDISTINCT
2: Output: Estimates of marginals  $b = (b[1], \dots, b[m])$ 
3: for  $j = 1, \dots, m$  do
4:   Set  $z^{(j)}[1] = y^T[j]$ 
5:   Set  $z^{(j)}[2] = -y^T[j]$ 
6: end for
7: Run  $M$  on  $x \rightarrow z^{(1)} \circ z^{(2)} \circ \dots \circ z^{(m)}$  and record answer vector  $r$ 
8: for  $j \in [m]$  do do
9:    $b[j] = r[(2j - 1)]/n$ 
10: end for
11: output  $b$ 

```

References

- 1 Aditya Akella, Ashwin Barambe, Mike Reiter, and Srinivasan Seshan. Detecting ddos attacks on isp networks. In *Proceedings of the Workshop on Management and Processing of Data Streams*, pages 1–2, 2003.
- 2 Daniel N Baker and Ben Langmead. Dashing: fast and accurate genomic distances with hyperloglog. *Genome biology*, 20:1–12, 2019.
- 3 Jean Bolot, Nadia Fawaz, S. Muthukrishnan, Aleksandar Nikolov, and Nina Taft. Private decayed predicate sums on streams. In *Proc. 16th ICDT*, pages 284–295, 2013. doi:10.1145/2448496.2448530.
- 4 Mark Bun, Jonathan R. Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. *SIAM J. Comput.*, 47(5):1888–1938, 2018. doi:10.1137/15M1033587.
- 5 Vera Clemens, Lars-Christian Schulz, Marten Gartner, and David Hausheer. Ddos detection in P4 using HYPERLOGLOG and COUNTMIN sketches. In *Proc. NOMS 2023*, pages 1–6, 2023.
- 6 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi:10.1007/11681878_14.
- 7 Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proc. 41st STOC*, pages 381–390, 2009. doi:10.1145/1536414.1536467.
- 8 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.
- 9 Alessandro Epasto, Jieming Mao, Andres Muñoz Medina, Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong. Differentially private continual releases of streaming frequency moment estimations. In Yael Tauman Kalai, editor, *Proc. 14th ITCS*, pages 48:1–48:24, 2023. doi:10.4230/LIPIcs.ITCS.2023.48.
- 10 Cristian Estan, George Varghese, and Michael E. Fisk. Bitmap algorithms for counting active flows on high-speed links. *IEEE/ACM Trans. Netw.*, 14(5):925–937, 2006.
- 11 Hendrik Fichtenberger, Monika Henzinger, and Lara Ost. Differentially private algorithms for graphs under continual observation. In *Proc. 29th ESA*, pages 42:1–42:16, 2021.
- 12 Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.
- 13 Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Proc. 2007 Conference on Analysis of Algorithms*, pages 127–146, 2007.
- 14 Badih Ghazi, Ravi Kumar, Jelani Nelson, and Pasin Manurangsi. Private counting of distinct and k-occurring items in time windows. In *Proc. 14th ITCS*, pages 55:1–55:24, 2023. doi:10.4230/LIPIcs.ITCS.2023.55.
- 15 Monika Henzinger, A. R. Sricharan, and Teresa Anna Steiner. Differentially private histogram, predecessor, and set cardinality under continual observation, 2023. arXiv:2306.10428.
- 16 Palak Jain, Iden Kalemaj, Sofya Raskhodnikova, Satchit Sivakumar, and Adam Smith. Counting distinct elements in the turnstile model with differential privacy under continual observation, 2023. arXiv:2306.06723.
- 17 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. An optimal algorithm for the distinct elements problem. In *Proc. 29th PODS*, pages 41–52, 2010.
- 18 Matti Karppa and Rasmus Pagh. Hyperlogloglog: Cardinality estimation with one log more. In *Proc. 28th KDD*, pages 753–761, 2022.

- 19 Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. Why go logarithmic if we can go linear?: Towards effective distinct counting of search traffic. In *Proc. 11th EDBT*, pages 618–629, 2008.
- 20 Dingyu Wang and Seth Pettie. Better cardinality estimators for hyperloglog, pcsa, and beyond. In *Proc. 42nd PODS*, pages 317–327, 2023.
- 21 Lotte Weedage, Nelly Litvak, and Clara Stegehuis. Locating highly connected clusters in large networks with hyperloglog counters. *J. Complex Networks*, 9(2), 2021.