A Simple and Robust Protocol for Distributed Counting

Edith Cohen
Google Research, Mountain View, CA, USA
Tel Aviv University, Israel

Moshe Shechner
Tel Aviv University, Israel

Uri Stemmer
Tel Aviv University, Israel
Google Research, Tel Aviv, Israel

Abstract

We revisit the *distributed counting* problem, where a server must continuously approximate the total number of events occurring across k sites while minimizing communication. The communication complexity of this problem is known to be $\Theta(\frac{k}{\varepsilon}\log N)$ for deterministic protocols. Huang, Yi, and Zhang (2012) showed that randomization can reduce this to $\Theta(\frac{\sqrt{k}}{\varepsilon}\log N)$, but their analysis is restricted to the *oblivious setting*, where the stream of events is independent of the protocol's outputs.

Xiong, Zhu, and Huang (2023) presented a *robust* protocol for distributed counting that removes the oblivious assumption. However, their communication complexity is suboptimal by a polylog(k) factor and their protocol is substantially more complex than the oblivious protocol of Huang et al. (2012). This left open a natural question: could it be that the simple protocol of Huang et al. (2012) is already robust?

We resolve this question with two main contributions. First, we show that the protocol of Huang et al. (2012) is itself *not* robust by constructing an explicit adaptive attack that forces it to lose its accuracy. Second, we present a new, surprisingly simple, robust protocol for distributed counting that achieves the optimal communication complexity of $O(\frac{\sqrt{k}}{\varepsilon}\log N)$. Our protocol is simpler than that of Xiong et al. (2023), perhaps even simpler than that of Huang et al. (2012), and is the first to match the optimal oblivious complexity in the adaptive setting.

2012 ACM Subject Classification Theory of computation; Theory of computation → Distributed algorithms

Keywords and phrases Distributed Streaming, Adversarial Streaming

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.40

Related Version *Full Version*: `https://arxiv.org/abs/2509.05870` [28]

Funding *Edith Cohen*: Partially supported by Israel Science Foundation (grant 1156/23).
Moshe Shechner: Partially supported by the Israel Science Foundation (grant 1419/24).
Uri Stemmer: Partially supported by the Israel Science Foundation (grant 1419/24), and the Blavatnik Research Foundation.

1 Introduction

In the *distributed counting problem*, there is a server (a.k.a. the *coordinator*) and k sites. Throughout the execution, each site observes *events* occurring at different times. The goal of the server is to maintain an ongoing ε-approximation of the number of events observed by all sites together, while minimizing the communication cost.[1]

[1] For consistency with prior work, throughout the introduction we measure communication complexity by the total number of messages sent during the execution. See Section 4.2.2 for communication in bits.

Keralapura, Cormode, and Ramamirtham [51] presented a simple *deterministic* protocol for this problem with a communication cost of $O\left(\frac{k}{\varepsilon}\log N\right)$, where N is the total number of events. In their protocol, each site tracks its local event count and notifies the server whenever this count increases by a factor of $(1+\varepsilon)$. The number of notifications per site is bounded by $O(\frac{1}{\varepsilon}\log N)$, and the total communication is bounded by $O(\frac{k}{\varepsilon}\log N)$. Notably, this protocol requires only one-way communication from the sites to the server. Yi and Zhang [69] later established a lower bound of $\Omega\left(\frac{k}{\varepsilon}\log N\right)$ on the communication complexity of every deterministic protocol, even when two-way communication is allowed. This shows that the protocol of Keralapura et al. [51] is optimal among deterministic protocols.

For *randomized* protocols, the picture is more nuanced. Huang, Yi, and Zhang [49] showed that with only one-way communication, the complexity remains $\Theta(\frac{k}{\varepsilon}\log N)$, offering no asymptotic improvement over deterministic protocols. However, for the two-way setting, they presented a protocol achieving a lower cost of $O(\frac{\sqrt{k}}{\varepsilon}\log N)$, provided that $\varepsilon \leq 1/\sqrt{k}$.[2] Furthermore, they proved a matching lower bound, showing that their two-way randomized protocol is asymptotically optimal among all randomized protocols for this problem.

Oblivious vs. robust protocols. Huang et al. [49] analyzed their randomized protocol under the assumption that the entire input sequence is fixed in advance. That is, for any fixed input sequence, their protocol must succeed with high probability over its internal randomness. This model is known as the *oblivious setting* because the entity generating the inputs is "oblivious" to the protocol's outputs. Protocols designed to succeed in this setting are called *oblivious* protocols. In contrast, protocols that provably maintain utility even when the input sequence is chosen adaptively, as a function of the protocol's previous outputs, potentially in an adversarial manner, are called *robust* protocols (a.k.a. *adaptive* protocols). It can be easily seen that any deterministic protocol that guarantees correctness in the oblivious setting is automatically robust.[3] However, this is not the case for randomized protocols. Intuitively, the difficulty is that when inputs are chosen adaptively, these inputs can become dependent on the internal randomness of the protocol, thereby breaking the analysis (and correctness) of many oblivious protocols. Designing (randomized) robust algorithms and data structures that outperform deterministic ones is a very active research area in several sub-fields of theoretical computer science.[4]

The work of Huang et al. [49] gave rise to the question of understanding the communication complexity of *robust* protocols for the distributed counting problem. This question was addressed by Xiong, Zhu, and Huang [68] who presented a robust variant of the protocol of Huang et al. [49]. Their protocol is based on a generic "robustification technique" from Hassidim et al. [44]. This technique uses differential privacy (DP) to protect not the input data, but rather the algorithm's internal randomness. As Hassidim et al. [44] showed, this can be used to limit (in a precise way) the dependencies between the internal randomness of the algorithm and its inputs, thereby making it easier to argue about the adaptive setting.[5] While this "robustification technique" is quite generic, it often comes at the cost of increased algorithmic complexity and reduced performance compared to the "base" oblivious algorithm.

---

[2] Throughout the introduction, when stating the communication complexity of randomized protocols we assume that the answers are pointwise accurate with constant probability. For example, for every time step t, the released answer is ε-correct with probability at least 0.99. The dependency on the failure probability will be made precise in the technical sections that follow.

[3] Let $\mathcal{P}$ be a deterministic oblivious protocol and suppose towards contradiction that there is an adaptive adversary $\mathcal{A}$ that causes $\mathcal{P}$ to fail on inputs generated (adaptively) by $\mathcal{A}$. Let $\vec{x}$ denote the input sequence generated by $\mathcal{A}$ when interacting with $\mathcal{P}$. Now, as $\mathcal{P}$ is deterministic, it must also fail on this input sequence $\vec{x}$ even when it is fixed ahead of time, which contradicts the utility guarantees of $\mathcal{P}$.

[4] See, e.g., [1, 4, 5, 7–16, 19, 20, 23–26, 35, 39–45, 50, 53, 54, 57, 61, 65, 67].

[5] Following Hassidim et al. [44], differential privacy was used as a tool to "robustify" many oblivious algorithms in several settings. See, e.g., [4, 6, 7, 22, 27, 39, 56, 59].

At a high level, the robust protocol of Xiong et al. [68] has two components: (1) An oblivious protocol, similar to that of Huang et al. [49] but more suited to serve as the "base" protocol in the robustification-via-DP framework; and (2) a DP layer that adds appropriately calibrated noise in order to "hide" the internal randomness used by the base protocol from anyone who observes the released outputs. This approach has three shortcomings:

1. The resulting protocol is significantly more complex compared to the oblivious protocol of Huang et al. [49], mainly due to the DP layer.
2. The use of differential privacy as a robustification method inflates the communication complexity by a factor of $\log^{0.75}(k)$, from $O\left(\frac{\sqrt{k}}{\varepsilon}\log N\right)$ to $O\left(\frac{\sqrt{k}\cdot\log^{0.75}(k)}{\varepsilon}\log N\right)$.
3. The resulting protocol is robust only in the *black-box setting*, where the inputs might depend on the outputs of the server, but not directly on the internal state of the protocol or the messages transmitted between the server and the sites.

These shortcomings raise the following question:

▶ **Question 1.** *Could it be that the randomized protocol of Huang et al. [49] is itself robust, thereby avoiding these shortcomings?*

This question was left open by Xiong et al. [68]. That is, even though they presented a robust variant of the protocol of Huang et al. [49], they did not present an attack or argue about the robustness of the original (randomized) protocol.

1.1 Our results

We give two answers to Question 1.

1. First, we show that the original (randomized) protocol of Huang et al. [49] is itself *not* robust. Specifically, we design an adaptive attack that generates the inputs (sequences of events) in a way that forces the protocol of Huang et al. [49] to fail.
2. Nevertheless, we show that small modifications to the protocol of Huang et al. [49] do make it robust (without using any external "robustification techniques"). This allows us to avoid all three shortcomings listed above. Specifically, our protocol is as simple as that of Huang et al. [49] (perhaps even simpler), our communication complexity is optimal, and our protocol is robust even in the face of a *white-box* attacker where the inputs might be generated both as a function of the released outputs and as a function of the internal state and messages transmitted between the server and the sites.

These results are specified in the following two theorems.

▶ **Theorem 2** (Adaptive attack, informal version of Theorem 4)**.** *There exists an attack that generates adaptive inputs to the protocol of Huang et al. [49] that forces it to fail (have error greater than $(1+\varepsilon)$) with high probability.*

▶ **Theorem 3** (Simple robust protocol, informal version of Theorem 17)**.** *There is a robust protocol in the white-box setting for the distributed counting problem with k sites that guarantees ε-accuracy over N events while achieving optimal communication complexity of $O\left(\frac{\sqrt{k}}{\varepsilon}\log N\right)$.*

Before giving a technical overview of our results, we first describe the oblivious protocol of Huang et al. [49], so that we could illustrate our adaptive attack on it. We then describe the robust protocol of Xiong et al. [68], so that we could convey the simplicity of our new protocol.

1.2 Informal overview of the oblivious protocol of Huang et al. [49]

At a high level, the protocol of Huang et al. [49] operates in *rounds*, where each round processes roughly twice the number of events as the previous one. Hence, the total number of rounds is $\Theta(\log N)$. The breakdown into rounds is determined by a deterministic background protocol, similar to that of Keralapura et al. [51]: Each site i transmits a message to the coordinator each time its local event count n_i doubles. This allows the coordinator to maintain a factor 2 approximation $n' \in [n/2, n]$ of the total event count $n = \sum_{i\in[k]} n_i$. When n' roughly doubles, the server ends the current round and broadcasts to all sites an updated *transmission probability* $p \propto \frac{\sqrt{k}}{\varepsilon n'} = \Theta\left(\frac{\sqrt{k}}{\varepsilon n}\right)$. This transmission probability remains fixed during the round and is roughly halved from round to round.

The finer approximation is achieved via the following randomized protocol. For each event arrival at site i, the site samples a Bernoulli random variable with parameter p. If the sample is 1, the site transmits its exact local counter $\bar{n}_i = n_i$ to the server. Throughout the execution, the server estimates each local count n_i as $\hat{n}_i = 0$ if no transmission was received from site i and as $\hat{n}_i = \bar{n}_i - 1 + 1/p$, where $\bar{n}_i$ is the *last* value received from site i. The correction term $(-1 + 1/p)$ accounts for the expected value of $n_i - \bar{n}_i \sim \text{Geom}[p] - 1$, which is the number of events that occurred since the last report by the site. The server then estimates the total event count as $\hat{n} = \sum_{i\in[k]} \hat{n}_i$.

Huang et al. [49] showed that for any fixed input sequence, the estimator for each local count, $\hat{n}_i$, is unbiased, i.e., $\mathrm{E}[\hat{n}_i] = n_i$, and that it has a bounded variance of $\mathrm{Var}(\hat{n}_i) \le \frac{1}{p^2}$. Thus, the variance of the final estimator satisfies $\mathrm{Var}(\hat{n}) = \mathrm{Var}\left(\sum_{i\in[k]} \hat{n}_i\right) \le \frac{k}{p^2}$, which is at most $(\varepsilon n)^2$ as $p = \Theta\left(\frac{\sqrt{k}}{\varepsilon n}\right)$. So the standard deviation is of the order of εn, as required.

When a round ends and p is updated to p', the server performs a random correction on its stored count $\bar{n}_i$ for each site i. This ensures that the corrected value has the exact same distribution as if the protocol had been run all along with transmission probability p'. Consequently, the variance bound ($\mathrm{Var}(\hat{n}) \le \varepsilon n$) remains valid throughout the execution.[6]

The expected communication cost per round includes $O(k)$ messages for broadcasting the transmission probability p, and $p \cdot O(n)$ messages from the sites to the server, since each of the n events is transmitted with probability p. Overall, the communication cost is

$$O\left((k + pn) \cdot \log N\right) = O\left(\frac{\sqrt{k}}{\varepsilon} \log N\right),$$

where the last equality follows by plugging in p and by the assumption that $k \le 1/\varepsilon^2$.

1.3 Our adaptive attack on the protocol of Huang et al. [49]

We construct an adaptive adversary that observes the server's output and selectively generates events that cause the protocol of Huang et al. [49], which we denote as $\mathcal{A}$, to fail. The attack operates iteratively in a round-robin fashion over the sites as follows: it continues sending events to a given site until it observes a change in the output of $\mathcal{A}$. Once a change is detected, the attack proceeds to the next site. The intuition is that when the output of $\mathcal{A}$ changes, it means that the current site i has just transmitted a message to the server, with $\bar{n}_i = n_i$. At that point the estimator $\hat{n}_i$ is larger than the actual local count n_i by $(-1 + 1/p)$. Hence,

[6] Specifically, when p is updated to p', the server updates every $\bar{n}_i$ to $\bar{n}'_i = \bar{n}_i - Z$ for an appropriate random variable Z whose distribution depends on p, p'.

by stopping sending events to that site, the attacker introduces a positive estimation bias. Furthermore, this bias persists (in expectation) even when the server changes to the next round, as the random corrections to $\bar{n}_i$ do not change the expectation of the estimate $\hat{n}_i$. By repeating this process across multiple sites, the adversary accumulates a global bias in the total estimate $\hat{n}$, thereby causing the server to lose accuracy. After sufficiently many events, the expected bias at any given time is $\approx (k-1)/p \approx \sqrt{k}\varepsilon n$. This can be formalized to obtain Theorem 2. See Section 2 for more details.

1.4 Informal overview of the robust protocol of Xiong et al. [68]

We now elaborate on the robust protocol of Xiong et al. [68]. As we mentioned, this protocol has two components: (1) An oblivious "base protocol", similar to that of Huang et al. [49]; and (2) a DP-layer for robustifying this oblivious protocol.

We first describe the oblivious "base protocol" used by Xiong et al. [68]. At a high level, like Huang et al. [49], this oblivious protocol operates in rounds. In the beginning of each round, the server notifies the sites of the new round, collects exact counts from all the sites, and broadcasts the exact total count till now, denoted as n_0. Given n_0, every site i divides its stream of events into blocks of size $\Delta = O(\varepsilon \cdot n_0/\sqrt{k})$. For each block j, site i samples a random threshold $r_{i,j} \in [\Delta]$. Throughout the round, site i sends a message to the server whenever its local count crosses any of its internal thresholds $r_{i,j}$. On the server side, we estimate the total event count from the beginning of the round as $B \cdot \Delta$, where B denotes the number of messages received from all sites together. This protocol has similar performance to Huang et al. [49], and similarly, its analysis assumes an oblivious input.

We now elaborate on how Xiong et al. [68] robustified this protocol using DP. To motivate this, observe that if the attacker knows all the internal thresholds $r_{i,j}$, then it can easily attack this protocol, similarly to the way we attacked the protocol of Huang et al. [49]. Specifically, the attacker sends events to site i till its local count crosses a "large" threshold, say at least $3\Delta/4$. This accumulates a bias of $\Omega(\varepsilon n_0/\sqrt{k})$ in the estimation for the local count of site i. The attacker then continues to the next site. After $\Omega(\sqrt{k})$ sites, the attacker achieves a significant bias of more than εn_0, thereby causing the protocol to fail. So we want to hide the internal thresholds from the attacker.

However, note that even if the attacker does not know the thresholds then it can still attack the protocol in a similar manner. Specifically, the attacker sends events to site i till it sees that the output of the protocol was modified. At that moment the attacker knows that site i has just transmitted a message, and therefore it learns its current threshold exactly (as the attacker knows how many events site i has received), and it can hence conduct the same attack as before. So we also want to hide the times at which the output of the protocol changes.

Xiong et al. [68] used two generic techniques from differential privacy in order to hide these two aspects of their oblivious protocol: the `AboveThreshold` algorithm of Dwork et al. [37] and the binary tree mechanism of Dwork et al. [36]. They then proved, building on the work of Hassidim et al. [44], that the resulting protocol is robust in the *black-box* model. On the other hand, a *white-box* attacker, who knows the internal thresholds or sees the communication between the server and the sites, can still conduct the attack described above. Thus, the protocol of Xiong et al. [68] is not robust in the white-box model.

1.5 Informal overview of our simple robust protocol

Our protocol also operates in rounds. Similarly to the oblivious "base protocol" of Xiong et al. [68], in the beginning of each round, the server notifies the sites of the new round, collects exact counts from all the sites, and broadcasts the exact total count so far, denoted as n_0. Given n_0, every site determines this round's *transmission probability* $p \approx \frac{\sqrt{k}}{\varepsilon n_0}$. Then, throughout the round, whenever a site receives an event, it samples a bit b from Bernoulli(p) and if $b = 1$ then the site sends the message "1" to the server. At any moment throughout the round, the server estimates the current total number of events as $n_0 + B/p$, where B denotes the number of "1" messages the server has received from the beginning of the round. The round ends when $B = k$, which happens after $\frac{k}{p} \approx \varepsilon\sqrt{k}n_0$ events in expectation.

We show that this absurdly simple protocol is both optimal and adversarially robust (in the white-box model). We sketch the arguments here; see Sections 3, 4, 5 for the formal details.

Communication. Note that n_0 increases (in expectation) by a factor of $(1 + \varepsilon\sqrt{k})$ from round to round. Therefore, the total number of rounds is $\approx \log_{1+\varepsilon\sqrt{k}} N$, which for $\varepsilon < 1/\sqrt{k}$ is $\approx \frac{1}{\sqrt{k}\varepsilon} \log N$. The communication cost of each round is $O(k)$: In the beginning of the round the server broadcasts the start of the new round and obtains an exact local count from each of the sites, and throughout the round the server receives exactly k "1" messages from the sites. Therefore, the expected overall communication cost is $O\left(\frac{\sqrt{k}}{\varepsilon} \log N\right)$, matching the communication cost of the oblivious protocol of Huang et al. [49].

Robustness. The robustness of our protocol follows from the fact that it is symmetric w.r.t. which site receives an event. Specifically, at any given moment throughout the execution, and for every conditioning on the transcript of the interaction so far, the distribution on the outcome of the server is *identical* whether site i receives an event or site j receives an event. Thus, the attacker might as well send all events to site $i = 1$, as this has no effect on the outcome distribution of the server. As sending all events to site $i = 1$ is an *oblivious* stream of inputs, it suffices to analyze the error of our protocol in the oblivious setting.

Accuracy. So let us consider the oblivious stream where all events arrive at site $i = 1$. For consistency with prior works, we analyze our protocol in terms of *"per-time"* accuracy (a.k.a. *pointwise* accuracy): For every time step n, with probability 0.99 the returned estimate at this time, $\hat{n}$, satisfies $|\hat{n} - n| \leq \varepsilon n$. Here we only sketch the argument; see Sections 4 and 5 for the formal details, along with a *"for-all"* accuracy guarantee (uniform across all time steps). At a high level, our utility analysis has two parts:

(1) ***"Per-round"* accuracy.** For every r we show that with probability at least 0.99, the maximum relative error of the count at all events that fall in the rth round is at most ε. To see why this holds, recall that the number of "1" messages we receive during the round is exactly k, and that the number of events witnessed during this round is distributed as the sum of k geometric random variables $X_1, \ldots, X_k \sim \text{Geom}(p)$. That is, after the ℓth "1" message, our estimate is $\left(n_0 + \frac{\ell}{p}\right)$, where n_0 is the exact count at the beginning of the round, while the actual event count is $\left(n_0 + \sum_{j=1}^{\ell} X_j\right)$. We can thus bound our estimation error using standard (partial-sum) tail bounds on the geometric distribution. Specifically, with probability at least 0.99 it holds that $\max_{1 \leq \ell \leq k} \left|\sum_{j=1}^{\ell} X_j - \frac{\ell}{p}\right| \lesssim \frac{\sqrt{k}}{p} \approx \varepsilon n_0 \leq \varepsilon n$, yielding our desired relative error throughout the round.

(2) **From per-round to per-time accuracy.** Converting a per-round guarantee into a per-time guarantee is subtle, because for a fixed time step n it is not clear a priori which round r the time step n will belong to. Furthermore, conditioning on n belonging to a specific round r changes the distribution of our geometric RVs and breaks our per-round analysis. A naive solution here would be to union bound over all rounds, but this would yield a "for-all" guarantee that would not match the optimal per-time complexity. Informally, we overcome this by showing that it suffices to union bound over *constantly* many rounds rather than all rounds. More specifically, for a fixed time step t we let $t_0 < t$ denote the largest time step such that with high probability there is at least one round that starts between time t_0 and t. This allows us to ignore the execution before time t_0, as the error in step t is independent of past rounds (due to the sync in the beginning of the round starting between t_0 and t). Furthermore, we show that in expectation there is at most a constant number of rounds that begin between t_0 and t, and that it suffices to argue only about these $O(1)$ rounds.

These arguments can be formalized to obtain Theorem 3, showing that our simple protocol is both optimal and adversarially robust in the white-box model.

1.6 Empirical demonstration

In the full version of the paper (See [28]), we present simulation results comparing our robust protocol with the oblivious protocol of Huang et al. [49]. We test both protocols on two types of streams: our adaptive attack stream and a non-adaptive input stream. We did not include an evaluation of the robust protocol of Xiong et al. [68], as its source code was not available, its implementation relies on external DP libraries, and its associated constants appear to be large. We observe that:

(1) As predicted by our analysis, the protocol of Huang [49] is vulnerable to the adaptive attack, whereas our protocol is robust. The difference is quite striking, even more so than guaranteed by our theoretical analysis of the attack, which suggests the constants in our analysis were not tight.

(2) On the non-adaptive stream, both protocols exhibit similar performance. This shows that robustness does not come at the expense of practicality. (Recall that both protocols have the same asymptotic guarantees on oblivious streams.)

1.7 Related works

The distributed counting problem is a specific instance of the more general framework of *distributed functional monitoring*, where the goal is to continuously track a function over data streams distributed across multiple sites. This framework has been extensively studied for various functions beyond simple sums. Significant research has focused on estimating frequency moments, identifying heavy hitters, approximating quantiles, and more [2, 21, 29–33, 47, 51, 58, 62, 66].

The distributed functional monitoring model is related to other multi-player communication models. A notable example is the classic *coordinator model*, introduced by Dolev et al. [34], where k players, each holding a static input, communicate with a central referee to compute a function of their joint inputs in a single shot. A significant body of work has analyzed the communication and round complexity of core problems within this model, including [3, 18, 38, 46, 48, 55, 60, 63, 64].

2 An Adaptive Attack on the Protocol of Huang et al. [49]

In this section we show that the protocol proposed in Huang et al. [49], is not robust by presenting an adaptive attack. The attack consistently induces a relative estimation error larger by a factor of $\Omega(\sqrt{k})$ compared to the target accuracy ε.

▶ **Theorem 4** (Attack on the protocol of Huang et al. [49]). *There exist universal constants $c_1, c_2, c_3 > 0$ and an adaptive attack on the protocol of [49] such that for every event count $n > c_3\sqrt{k}/\varepsilon$ we have* $\Pr\left[\hat{n} - n > c_1\sqrt{k}\,\varepsilon n\right] \geq 1 - e^{-c_2 k}$.

2.1 Description of the protocol of Huang et al. [49]

The protocol of Huang et al. [49] is described in Algorithms 1 and 2. We briefly review the protocol and its analysis, which assumes that the *event stream is fixed in advance*.

Algorithm 1 Doubling Protocol Keralapura et al. [51].

Output: A running estimate $n' \in [n/2, n]$ of $n = \sum_i n_i$ and BoundaryReached when n' doubles.

Server state: for each site i, $n'_i \leftarrow 0$; total $n' \leftarrow 0$; doubling threshold $\tau \leftarrow 1$.
Site i state: local count $n_i \leftarrow 0$;

Event arrival at site i: // runs independently at each site
  $n_i \leftarrow n_i + 1$
  if n_i is a power of 2 **then** send DoublingNotify(i, n_i) to server

Upon server receiving DoublingNotify(i, n_i):
  $n' \leftarrow n' + n_i - n'_i$; $n'_i \leftarrow n_i$
  if $n' \geq 2\tau$ **then**
    BoundaryReached(n'); $\tau \leftarrow n'$
    // n' doubled

The Doubling protocol (Algorithm 1, Keralapura et al. [51]) is deterministic and runs concurrently on the same input. It maintains a constant-factor approximation of the total event count: each site notifies the server whenever its local count doubles, and the server triggers a BoundaryReached alert when its global estimate doubles. On N events, this protocol incurs a total communication cost of $k \log N$.

The site-side of Huang et al. [49] protocol is very simple: Each site i keeps a local copy of a transmission probability p that is updated via server broadcasts and its local event count n_i. On each event, it reports its local count to the server with probability p.

On the server side, the protocol progresses in rounds. The server stores for each site a value $\bar{n}_i$ (the last reported count, adjusted as described below) and computes the per-site estimate $\hat{n}_i = \bar{n}_i + 1/p - 1$ (when $\bar{n}_i > 0$). The global estimate is $\hat{n} = \sum_i \hat{n}_i$. A new round starts upon a BoundaryReached alert, at which point the server broadcasts an updated probability p and adjusts each $\bar{n}_i$ so that the resulting state matches the distribution that would have arisen had the new p been in effect from the beginning.

The analysis of Huang et al. [49] holds under the assumption that the event stream is fixed in advance. First, note that with a fixed transmission probability p, the most recent report from a site with current local count n_i satisfies $\bar{n}_i \sim n_i + 1 - \text{Geom}[p]$. Hence the per-site estimate $\hat{n}_i := \bar{n}_i + 1/p - 1$ is an unbiased estimator of n_i.

Algorithm 2 Huang et al. [49].

Input: parameters k, ε; BoundaryReached from protocol Algorithm 1.
Output: Running estimate $\hat{n} = \sum_i \hat{n}_i$.

// Site protocol for sites $i \in [k]$:
Site i state: $p \leftarrow 1$ (updated via broadcasts by server), $n_i \leftarrow 0$ event counter.

Event arrival at site i: // runs independently at each site
  $n_i \leftarrow n_i + 1$
  Sample $X \sim \text{Bernoulli}(p)$.
  if $X = 1$ **then** send $\text{Report}(i, n_i)$ to server // report exact local n_i

Upon site i receiving Broadcast(p): update local p.

// Server protocol:
Server state: for each site i: last report $\bar{n}_i \leftarrow 0$, per-site estimate $\hat{n}_i \leftarrow 0$; current transmit prob $p \leftarrow 1$.

Upon BoundaryReached(n') OR server receiving Report($i, \bar{n}$): // If an event triggers both, server processes the report first.
  if *Report*$(i, \bar{n})$ *is received* **then**
    $\bar{n}_i \leftarrow \bar{n}$; $\hat{n}_i \leftarrow \bar{n}_i - 1 + \frac{1}{p}$.
    Publish $\hat{n} \leftarrow \sum_j \hat{n}_j$.
  if *BoundaryReached*(n') **then** // close old round, open new round
    $p_{\text{old}} \leftarrow p$; $p \leftarrow 2^{\min\{0, \lfloor \log_2 \frac{\sqrt{k}}{\varepsilon n'} \rfloor\}}$ // power of 2 that is $\Theta\left(\frac{\sqrt{k}}{\varepsilon n'}\right)$.
    if $p < 1$ **then** // When $p = 1$ server maintains exact counts
      for $i \in [k]$ **do** // Adjust $\bar{n}_i$ to updated p
        Sample $Z_i \sim Z_{p, p_{\text{old}}}$ // as in Definition 7; $B \sim \text{Bernoulli}[1 - p/p_{\text{old}}]$; $G \sim \text{Geom}[p]$; $Z_i \leftarrow B \cdot G$
        $\bar{n}_i \leftarrow \max\{0,\ \bar{n}_i - Z_i\}$.
        if $\bar{n}_i = 0$ **then** $\hat{n}_i \leftarrow 0$ **else** $\hat{n}_i \leftarrow \bar{n}_i - 1 + \frac{1}{p}$
      Broadcast(p) to all sites.

When the protocol decreases p at the start of a new round, the server updates each $\bar{n}_i$ by subtracting a random variable Z_i with the property[7] that $\text{Geom}[p_{\text{old}}] + Z_i \overset{d}{=} \text{Geom}[p]$. This ensures that the distribution of the server state after the update is identical to the distribution that would have been obtained had the protocol run with the new probability p from the start. Moreover, the expected value of $\hat{n}_i$ is unchanged, since $\mathrm{E}[Z_i] = 1/p - 1/p_{\text{old}}$.

Therefore, under a fixed input stream, the global estimate $\hat{n} = \sum_i \hat{n}_i$ remains unbiased for all time. The error distribution at any event count n is stochastically dominated by the deviation of a sum of k independent $\text{Geom}[p]$ random variables from its expectation k/p. By standard sub-exponential tail bounds (see Equation (4)), for any fixed $\delta > 0$, the deviation is $O(\varepsilon n)$ with probability at least $1 - \delta$.

This completes our overview of the analysis of Huang et al. [49] in the non-adaptive setting. We demonstrate that these guarantees fail under adaptively chosen event streams.

[7] These are *Zero-inflated geometrics*, see Definition 7.

2.2 Attack description

Our attack on the protocol of Huang et al. (Algorithm 2) is given in Algorithm 3. The attack continuously monitors the estimates $\hat{n}$ published by the server. It generates the input stream by injecting events exclusively at the current site i until the server updates its estimate. At that moment the attacker switches to the next site in a round-robin order.

Algorithm 3 Attack on the protocol of Huang et al. [49].

Input: Access to protocol with k sites $i \in [k]$: InjectEvent(i), Observe() returning current $\hat{n}$.

Output: A sequence of adversarial event injections.

Initialize $i \leftarrow 1$; $\hat{n}_{\text{last}} \leftarrow$ Observe().

while *horizon not reached* **do**

```
// Focus on site i only; all other sites are idle
repeat                                          // Wait for a change in running estimate
  | InjectEvent(i)                              // Generate event at site i
until n̂ ← Observe() ≠ n̂_last

n̂_last ← n̂                                     // Record current estimate
i ← 1 + (i mod k)                               // Move to the next site in round robin
```

2.3 Preliminaries for the Analysis of the Attack

For brevity, we use the same symbol to denote a random variable and its distribution. We write $\text{Geom}(q)$ for a geometric random variable with success probability q, supported on $\{1, 2, \ldots\}$ with $\Pr(G = k) = (1-q)^{k-1}q$. We also use the standard notation $\text{Bin}(n, p)$ and $\text{Bernoulli}(p)$ for the binomial and Bernoulli distributions, respectively.

▶ **Lemma 5** (Chernoff bound, multiplicative lower tail [52]). *Let $x_1, \ldots, x_n$ be independent $\{0,1\}$ random variables, and set $X = \sum_{i=1}^{n} x_i$ with mean $\mu = \mathrm{E}[X]$. Then for all $0 \le \alpha \le 1$ we have $\Pr[X \le (1-\alpha)\mu] \le \exp\left(-\frac{\mu\alpha^2}{2}\right)$.*

▶ **Corollary 6** (Binomial lower tail). *Let $B \sim \text{Binomial}(n, p)$ with $p < \frac{1}{2}$. Then for all $0 \le \alpha \le 1$ we have $\Pr[B \le (1-\alpha)pn] \le \exp\left(-\frac{pn\alpha^2}{2}\right)$.*

▶ **Definition 7** (Zero-inflated geometric random variables). *For $0 \le q \le p \le 1$, define*

$$Z_{q,p} := B \cdot G, \qquad B \sim \text{Bernoulli}\left(1 - \frac{q}{p}\right), \quad G \sim \text{Geom}(q),$$

with B and G independent.

The proofs for the following two Lemmas are available at the full version of the paper (See [28]).

▶ **Lemma 8** (Zero-inflated geometric telescopes). *Fix parameters $1 \ge p_1 \ge p_2 \ge \cdots \ge p_r > 0$. If Z_{p_{i+1},p_i}, $i = 1, \ldots, r-1$, are independent, then*

$$\sum_{i=1}^{r-1} Z_{p_{i+1},p_i} \overset{d}{=} Z_{p_r,p_1}.$$

▶ **Lemma 9** (Bernstein tail for a sum of zero-inflated geometrics). *Let $p \in (0,1]$, $\alpha_1, \ldots, \alpha_r \in [0,1]$, and for each i let $X_i = B_i\, G_i$ with $B_i \sim \mathrm{Bernoulli}(\alpha_i)$ and $G_i \sim \mathrm{Geom}(p)$, supported on $\{1,2,\ldots\}$ with $\Pr(G_i = k) = (1-p)^{k-1}p$. Assume $\{(B_i, G_i)\}_{i=1}^r$ are independent. Put $A := \sum_{i=1}^r \alpha_i$ and $\mu := \mathrm{E}[\sum_i X_i] = A/p$. Then for all $t \geq 0$,*

$$\Pr\Big(\sum_{i=1}^r X_i - \mu \geq t\Big) \;\leq\; \exp\left(-\frac{t^2}{2\left(\frac{A(1-p)}{p^2} + \frac{t}{p}\right)}\right) \;\leq\; \exp\left(-\frac{1}{2}\min\left\{\frac{p^2t^2}{A(1-p)},\, pt\right\}\right).$$

We bound the configured transmission probability and the round length in terms of the event count n_0 at the start of a round. These bounds hold for every input stream, including adaptively chosen inputs.

▶ **Observation 10** (transmission probability at round start). *Let a new round be initiated when the total event count is n_0. Then the configured transmission probability p satisfies*

$$\min\left\{1,\, \frac{\sqrt{k}}{2\varepsilon n_0}\right\} \;\leq\; p \;\leq\; \min\left\{1,\, \frac{2\sqrt{k}}{\varepsilon n_0}\right\}.$$

Proof. The server counter n' in Algorithm 1 satisfies $n' \in [n/2, n]$. Write $x = \frac{\sqrt{k}}{\varepsilon n'}$. If $x \geq 1$ then $\lfloor \log_2 x \rfloor \geq 0$ and $p = 1$. If $x < 1$ then $p = 2^{\lfloor \log_2 x \rfloor} \in [x/2, x]$, so $\frac{\sqrt{k}}{2\varepsilon n_0} \leq p \leq \frac{2\sqrt{k}}{\varepsilon n_0}$. Both cases give the stated bounds. ◀

▶ **Observation 11** (Relative round length). *Let $n_0 < n_1$ be the event count at the start of two consecutive rounds of Algorithm 1. Then $\frac{n_1 - n_0}{n_0} \leq 7$.*

Proof. Let n'_i be the respective values of the server counter n' at the start of the rounds. It holds that $n'_1 \in [2n'_0, 4n'_0]$. Furthermore, at any given time, the server's counter n' satisfies $n' \in [n/2, n]$, where n is the actual event count. Therefore $n_0 \geq n'_0$ and $n_1 \leq 2n'_1 \leq 8n'_0 \leq 8n_0$. ◀

2.4 Analysis of the Attack

Throughout this section we assume $\varepsilon \leq 1/\sqrt{k}$. We analyze the distribution of the protocol execution under the attack. A *transcript* denotes the full specification of the states of the attack, sites, and server up to some stopping point.

▶ **Lemma 12** (Many reporting sites). *Assume $\varepsilon\sqrt{k} < 1$. Fix a transcript up to the initiation of a round at event count $n_0 \geq 4\sqrt{k}/\varepsilon$, and consider the execution from event $n_0 + 1$ onward. Let R denote the number of distinct sites that send a Report message to the server during the next n_0 events. Then*

$$\Pr\left[R > \tfrac{k}{8}\right] \;\geq\; 1 - \exp\left(-\tfrac{\sqrt{k}}{32\varepsilon}\right).$$

Proof. Each update to the published estimate is caused by a Report message from the site currently injected by the attack. Whenever a site reports, the attack observes the change in $\hat{n}$ and moves to the next site.

The transmission probability values during the next n_0 events are configured at a round that starts at some event count $n \in [n_0, 2n_0]$ (that is, either at the round that starts at n_0 or at a later round). By Observation 10, using our assumption that $n_0 \geq 4\sqrt{k}/\varepsilon$, the

transmission probability values are in the interval $\left[\frac{\sqrt{k}}{4\varepsilon n_0}, \frac{2\sqrt{k}}{\varepsilon n_0}\right]$. Therefore, $\Pr\left[R < \frac{k}{8}\right] \le \Pr\left[\mathrm{Bin}\left[n_0, \frac{\sqrt{k}}{4\varepsilon n_0}\right] < \frac{k}{8}\right]$. Applying the Chernoff bound (Corollary 6) with $\alpha = 1 - \frac{\varepsilon\sqrt{k}}{2}$ gives

$$\Pr\left[\mathrm{Bin}\left[n_0, \tfrac{\sqrt{k}}{4\varepsilon n_0}\right] < \tfrac{k}{8}\right] \le \exp\left(-\tfrac{1}{2}\tfrac{\sqrt{k}}{4\varepsilon}\left(1 - \tfrac{\varepsilon\sqrt{k}}{2}\right)^2\right) \le \exp\left(-\tfrac{\sqrt{k}}{32\varepsilon}\right),$$

where the last inequality uses $\varepsilon\sqrt{k} \le 1$.

Finally, by the round-robin structure of the attack, once $k/8$ sites have reported, they are necessarily distinct. Hence with probability at least $1 - \exp(-\sqrt{k}/(32\varepsilon))$, the number of distinct reporting sites in the round exceeds $k/8$. ◀

▶ **Corollary 13** (Many sites report once $n/14$ events have occurred). *Let $n \ge 56\sqrt{k}/\varepsilon$, and let $n_0 \ge n/14$ be the first event index that is greater or equal to $n/14$ at which a new round starts. Let R_n^* be the number of distinct sites that report at least once during $[n_0, n]$. Then*

$$\Pr\left[R_n^* > \tfrac{k}{2}\right] \ge 1 - \exp\left(-\tfrac{\sqrt{k}}{8\varepsilon}\right),$$

where the probability is over the randomness of the protocol execution from $n_0 + 1$ onward, conditioned on the transcript up to n_0.

Proof. By Observation 11, some round must start between $n/14$ and $n/2$; let n_0 be the first such start. Applying Lemma 12, with probability at least $1 - \exp(-\sqrt{k}/(8\varepsilon))$, more than $k/2$ sites report in the interval $[n_0, 2n_0] \subset [n_0, n]$. ◀

▶ **Definition 14** (Reporting transcript). *Consider the randomness in the execution of the protocol under attack. Observe that the attack algorithm is deterministic. The protocol, however, uses two types of randomness:*

- *site-side randomness, determining when sites send reports, and*
- *server-side randomness, used to adjust the counters $\bar{n}_i$ at the beginning of each round.*

We call a transcript that fixes all site-side randomness but leaves the server-side randomness unspecified a reporting transcript.

▶ **Observation 15.** *A reporting transcript uniquely determines the sequence of round start indices, the configured transmission probabilities, and the states of the attack algorithm and the sites. In particular, the value of R_n^* is fixed by the reporting transcript. The only remaining randomness lies in the server-side adjustments of $\bar{n}_i$ and, consequently, in the estimates $\hat{n}$ published by the server.*

▶ **Lemma 16** (Attack efficacy conditioned on reporting transcript). *Fix an event index $n \ge 500\sqrt{k}/\varepsilon$ and the reporting transcript up to that point. Let m^* denote the number of events received by the active site at event n since its most recent report. Assume $R_n^* \ge k/2$, where R_n^* is as in Corollary 13. Then there exist absolute constants $c_1, c_2 > 0$ such that*

$$\mathrm{E}[\hat{n} - n] = \mathrm{E}\left[\sum_{i=1}^{k}(\hat{n}_i - n_i)\right] \ge c_1\sqrt{k}\varepsilon n - m^*, \tag{1}$$

$$\Pr\left[\hat{n} - n \ge \tfrac{c_1}{2}\sqrt{k}\,\varepsilon n - m^*\right] \ge 1 - e^{-c_2 k}. \tag{2}$$

Proof. Let p be the transmission probability at event n. For each site i, let $p'_i \geq p$ be the transmission probability in the round in which i most recently reported. By Observation 11, any event time $t \in [n/2, n]$ lies in a round that started at some $n_0(t) \geq t/7 \geq n/14$. Applying Observation 10 at that start time gives

$$p'_i \;\leq\; \frac{2\sqrt{k}}{\varepsilon\, n_0(t)} \;\leq\; \frac{28\sqrt{k}}{\varepsilon\, n},$$

for every site that reports at some $t \in [n/2, n]$. Since $R^*_n \geq k/8$, at least $k/8$ sites satisfy $p'_i \leq \frac{28\sqrt{k}}{\varepsilon\, n}$.

Observe that $\bar{n}_i$ equals the last count reported by site i, minus the cumulative randomized adjustments applied so far. By Lemma 8, the distribution of this cumulative adjustment is Z_{p,p'_i}.

For $i \neq i^*$, the last reported count is equal to the current count n_i and therefore

$$\bar{n}_i = n_i - Z_{p,p'_i}, \qquad \mathrm{E}[\bar{n}_i] = n_i - \tfrac{1}{p} + \tfrac{1}{p'_i}.$$

For the active site i^*, the last reported count is $n_i - m^*$ and therefore

$$\bar{n}_{i^*} = n_{i^*} - m^* - Z_{p,p'_{i^*}}, \qquad \mathrm{E}[\bar{n}_{i^*}] = n_{i^*} - m^* - \tfrac{1}{p} + \tfrac{1}{p'_{i^*}}.$$

With the definition $\hat{n}_i := \bar{n}_i + \frac{1}{p} - 1$, we have

$$\mathrm{E}[\hat{n}_i] = \begin{cases} n_i + \frac{1}{p'_i} - 1, & i \neq i^*, \\ n_{i^*} - m^* + \frac{1}{p'_{i^*}} - 1, & i = i^*. \end{cases}$$

Hence

$$\mathrm{E}\left[\sum_{i=1}^{k}(\hat{n}_i - n_i)\right] = \sum_{i=1}^{k}\frac{1}{p'_i} - k - m^* \;\geq\; \frac{k}{8}\cdot\frac{\varepsilon n}{28\sqrt{k}} - k - m^* \;=\; \frac{\sqrt{k}\varepsilon n}{224} - k - m^*.$$

Since by our choice of large enough n, taking $c_1 = 1/500$ we obtain Equation (1).

For deviations, note that

$$\sum_{i=1}^{k}\hat{n}_i = \sum_{i=1}^{k} n_i - m^* + \sum_{i=1}^{k}\left(\frac{1}{p'_i} - 1\right) - \sum_{i=1}^{k} Z_{p,p'_i}.$$

Therefore, the distribution of the fluctuations of $\sum_i \hat{n}_i - n$ around its mean is exactly the distribution of the fluctuations of $\sum_i Z_{p,p'_i}$ around (its mean) $\sum_i(1/p - 1/p'_i)$. By Observation 10, $p \geq \sqrt{k}/(2\varepsilon n)$.

Applying Lemma 9 with deviation $t = \frac{c_1}{2}\sqrt{k}\,\varepsilon n$, using $p \geq \sqrt{k}/(2\varepsilon n)$ and $A \leq k$, yields

$$\Pr\left(\left|\sum_{i=1}^{k} Z_{p,p'_i} - \sum_{i=1}^{k}\left(\frac{1}{p} - \frac{1}{p'_i}\right)\right| > \tfrac{c_1}{2}\sqrt{k}\,\varepsilon n\right) \;\leq\; e^{-\frac{c_1^2}{8}k}.$$

Rewriting in terms of $\hat{n} - n + m^*$ and choosing the constants appropriately gives the claimed high-probability bound (2). ◀

Proof of Theorem 4. Fix an event index n and consider a transcript of the execution under the attack up to that point. We call a transcript *bad* if either $R_n^* < k/2$ or $m^* > \frac{1}{3}c_1\sqrt{k}\varepsilon n$.

From Corollary 13, the first bad event has probability at most $\exp(-\Omega(\sqrt{k}/\varepsilon))$. For the second bad event, recall that $m^* \sim \mathrm{Geom}(p)$ with $p \geq \sqrt{k}/(\varepsilon n)$. By standard geometric tail bound: $\Pr[m^* > t] = (1-p)^t \leq \exp(-pt)$. Substituting $t = \frac{1}{3}c_1\sqrt{k}\varepsilon n$ and $p \geq \sqrt{k}/(\varepsilon n)$ yields

$$\Pr\left[m^* > \tfrac{1}{3}c_1\sqrt{k}\varepsilon n\right] \;\leq\; \exp\left(-\tfrac{1}{3}c_1\sqrt{k}\varepsilon n \cdot \tfrac{\sqrt{k}}{\varepsilon n}\right) \;=\; \exp\left(-\tfrac{1}{3}c_1 k\right).$$

Thus the probability of a bad transcript is at most $e^{-\Omega(k)}$. Conditioned on a good transcript, we apply Lemma 16. From (2) we obtain $\Pr\left[\hat{n} - n \;\geq\; \frac{2}{3}c_1\sqrt{k}\varepsilon n\right] \;\geq\; 1 - e^{-c_2 k}$. Since good transcripts occur with probability $1 - e^{-\Omega(k)}$, the same lower bound holds unconditionally (with adjusted constants $c_1, c_2 > 0$). This proves the theorem. ◀

3 Robust Distributed Counting Protocol

Robust, our robust distributed counting protocol, is described in Algorithm 4. The site-side protocol of Robust is very simple and similar to that of HYZ12: The site stores a local copy of a transmission probability p that is updated by server broadcasts. The site also tracks its local event count as n_i. When an event arrives, n_i is incremented. With probability p, the site then sends a ReportSample() message to the server. When a CountRequest() request arrives from the server, the site sends the server its current count n_i.

Server-side, the protocol operates in rounds: In the beginning of each round the server broadcasts to all sites a CountRequest(), which requests their exact counts, and each site i reports n_i, which the server stores as $\bar{n}_i$. Therefore, in the beginning of the round the server has the exact event count $\bar{n} \leftarrow \sum_{i\in[k]} \bar{n}_i$. The server then updates the transmission probability p, based on the updated $\bar{n}$, and broadcasts it to the sites (which update their local copy). The transmission probability remains fixed throughout the round. The server also maintains a counter B of the number of ReportSample() messages received from sites during the round. The published estimate for the total number of events is $\hat{n} := \bar{n} + B/p$, and it is updated when either B or $\bar{n}$ were updated. The round ends when the kth report is received and a new round is started.

Properties of the Robust Protocol

The properties of the protocol are stated with respect to an adversary (event stream) that is *white-box adaptive*. Let $\hat{n}(n)$ be the server's estimate after the n-th event and let M_N denote the total number of *messages* sent by the time the N-th event occurs. All probabilities are over the protocol's internal randomness.

▶ **Theorem 17** (Robust: per-event (ε, δ) accuracy)**.** *Fix $\delta \in (0,1)$. There exists a constant $c = c(\delta) > 0$ such that, for any event stream and every $n \geq 1$,*

$$\Pr\left[\,|\hat{n}(n) - n| > \varepsilon n\,\right] \leq \delta.$$

Moreover,

$$\mathbb{E}[M_N] \;=\; O\left(\frac{k\log N}{\log\left(1 + \frac{\sqrt{k}\,\varepsilon}{c}\right)}\right) \;=\; \begin{cases} O\left(\dfrac{\sqrt{k}}{\varepsilon}\log N\right), & \varepsilon \leq c/\sqrt{k}, \\ O\big(k\log N\big), & \varepsilon > c/\sqrt{k}. \end{cases}$$

Algorithm 4 Robust Distributed Counting Protocol.

Input: number of sites k, accuracy ε, factor $c \geq 1$
Output: Running estimate $\hat{n}$ of the number of events n.

// Site protocol for sites $i \in [k]$:
Site i state: $p \leftarrow 1$ (updated via broadcasts by server), $n_i \leftarrow 0$ (local event counter).

Event arrival at site i: // runs independently at each site
  $n_i \leftarrow n_i + 1$
  Sample $X \sim \text{Bernoulli}(p)$.
  if $X = 1$ **then** send ReportSample() to server // report that an event was sampled

Upon site i receiving Broadcast(p): update local p.
Upon site i receiving a CountRequest(): send ReportCount(i, n_i) to server.

// Server protocol:
Server state: Counter of messages in current round $B \leftarrow 0$; Number of messages per round $L \leftarrow k$; for each site $i \in [k]$: $\bar{n}_i \leftarrow 0$ (reported n_i at the beginning of current round); current transmit prob $p \leftarrow 1$.
Running estimate: $\hat{n} \leftarrow B/p + \bar{n}$; where $\bar{n} := \sum_{i \in [k]} \bar{n}_i$.

Upon server receiving ReportSample():
  $B \leftarrow B + 1$
  if $B = k$ **then** // close round, open new round
    Broadcast CountRequest().
    Collect $\bar{n}_i \leftarrow$ ReportCount(i, n_i) from sites $i \in [k]$
    Update $p \leftarrow \min\left\{1, c\frac{\sqrt{k}}{\varepsilon \bar{n}}\right\}$; Broadcast$(p)$
    $B \leftarrow 0$

▶ **Theorem 18** (Robust: uniform (ε, δ) accuracy). *For any $\varepsilon > 0$, $\delta \in (0,1)$, and N, set*

$$c = O\left(\max\left\{\sqrt{A}, \frac{A}{\sqrt{k}}\right\}\right), \qquad A := \log\frac{1}{\delta} + \log\log N + \max\left\{0, \log\frac{1}{\sqrt{k}\,\varepsilon}\right\}.$$

Then, for any event stream of length at most N, with probability at least $1 - \delta$,

$$\max_{n \in [N]} \frac{|\hat{n}(n) - n|}{n} \leq \varepsilon, \qquad \text{and} \qquad M_N = O\left(\frac{k \log N}{\log\left(1 + \frac{\sqrt{k}\,\varepsilon}{c}\right)} + k \log\frac{1}{\delta}\right).$$

4 Analysis of the Robust Protocol

In this section we prove Theorem 17 and Theorem 18. In Section 4.1 we argue that our protocol is robust, in Section 4.2 we analyze the communication cost, and in Section 4.3 we analyze the accuracy.

4.1 Robustness to an adaptive adversary

Robustness of the protocol is immediate. An adaptive adversary in a white-box setting has the freedom to select which site to inject the next event to, based on the server and stations (historical and current) state and the transcript so far. Observe, however, that the actions

and effective state of the server at any given time are oblivious to the particulars of how the events are assigned to sites, and only depend on the total number of events that occurred. In particular, when syncing event counts at the beginning of each round, the server only uses $\bar{n}$ which is the total exact number of events. Likewise, the number and time of ReportSample() messages, counted by the server in B, do not depend on the site at which the event had occurred. The computed probability p, the termination and restart time of rounds, and the running estimate only depends on these two parameters $\bar{n}$ and B.

4.2 Communication Analysis

We bound the number of rounds that the protocol performs on the first N events.

▶ **Lemma 19** (Number of rounds for N events). *Let R_N be the number of rounds the protocol performs to process the first N events. Then*

$$\mathbb{E}[R_N] \;=\; \Theta\left(\frac{\log N}{\log\left(1+\frac{\sqrt{k}\,\varepsilon}{c}\right)}\right).$$

Moreover, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$R_N \;=\; O\left(\frac{\log N}{\log\left(1+\frac{\sqrt{k}\,\varepsilon}{c}\right)} \;+\; \log\frac{1}{\delta}\right),$$

where the implicit constant in the $O(\cdot)$ notation is absolute and independent of $\delta, \varepsilon, k, N$.

Proof. The number of events in one round is

$$S \;=\; \sum_{i=1}^{k} X_i, \qquad X_i \overset{\text{i.i.d.}}{\sim} \mathrm{Geom}(p), \quad p \;=\; \min\left\{1, \frac{c\sqrt{k}}{\varepsilon\,\bar{n}}\right\},$$

where $\mathrm{Geom}(p)$ is the geometric distribution on $\{1,2,\ldots\}$. Hence $\mathrm{E}[S] = k/p \;\geq \max\{k, (\varepsilon\sqrt{k}/c)\,\bar{n}\}$.

Call a round *big* if

$$S \;\geq\; \tfrac{1}{2}\,\mathbb{E}[S] \;\geq\; \frac{\varepsilon\sqrt{k}}{2c}\,\bar{n}.$$

For $X \sim \mathrm{Geom}(p)$ we have $\Pr(X \geq 1/(2p)) \geq 1/2$; by standard properties of sums of independent geometrics (negative binomial concentration), this implies

$$\Pr\big(S \;\geq\; \tfrac{1}{2}\,\mathbb{E}[S]\big) \;\geq\; \tfrac{1}{2}.$$

Thus each round is big with probability at least $1/2$, independently of other rounds.

Let

$$s^\star \;=\; \left\lceil \log_{1+\frac{\varepsilon\sqrt{k}}{2c}} N \right\rceil.$$

After $s^\star$ big rounds we must have processed at least N events. Let T be the number of rounds needed to accrue $s^\star$ big rounds. Then T is stochastically dominated by the number of Bernoulli(1/2) trials needed to see $s^\star$ successes, so $\mathbb{E}[T] \leq 2s^\star$, and by a standard Chernoff lower-tail bound, for all $\delta \in (0,1)$,

$$T \;\leq\; 2\left(s^\star + 2\sqrt{s^\star \ln\tfrac{1}{\delta}} + 4\ln\tfrac{1}{\delta}\right) \quad \text{with probability at least } 1-\delta.$$

Since $R_N \le T$, the high-probability bound follows. Moreover,

$$\mathbb{E}[R_N] \;\le\; \mathbb{E}[T] \;\le\; 2s^\star \;=\; \Theta\left(\frac{\log N}{\log\left(1+\frac{\sqrt{k}\,\varepsilon}{c}\right)}\right),$$

as replacing the base $1+\varepsilon\sqrt{k}/(2c)$ by $1+\varepsilon\sqrt{k}/c$ only changes the expression by an absolute constant factor. This completes the proof. ◀

4.2.1 Communication on the first N events

In each round the protocol sends $O(k)$ messages. We combine with Lemma 19 to obtain

$$\mathbb{E}[M_N] \;=\; O\left(k\,\frac{\log N}{\log\left(1+\frac{\sqrt{k}\,\varepsilon}{c}\right)}\right).$$

To establish the message bound in Theorem 17, take $c = \Theta(1)$; then $\log\bigl(1+\frac{\sqrt{k}\varepsilon}{c}\bigr) = \Theta(\min\{1,\sqrt{k}\varepsilon\})$.

Similarly from Lemma 19 we obtain that for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$M_N \;=\; O\left(k\,\frac{\log N}{\log\left(1+\frac{\sqrt{k}\,\varepsilon}{c}\right)} \;+\; k\log\frac{1}{\delta}\right).$$

To establish the message bound in Theorem 18, we substitute $c = O\bigl(\max\{\sqrt{A},\, A/\sqrt{k}\}\bigr)$ (with A as defined there).

4.2.2 Communication bounds in bits

Following prior work, we measure communication in terms of *messages*. We present a lightly modified version of our protocol (Algorithm 4) in which each message has size $O(\log(k/\varepsilon))$ bits, improving over the $O(\log N)$-bit messages required in a vanilla implementation.

▶ **Lemma 20** (Communication bound in bits). *There is a (lightly modified) version of Algorithm 4 with the same accuracy guarantees whose per-round communication is*

$$O\left(k \;+\; \min\left\{k,\; \frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right\}\cdot\log\frac{k}{\varepsilon}\right) \quad \textit{bits.}$$

Proof. The vanilla description of the protocol (Algorithm 4) uses messages that are event-triggered from sites to the server. These messages are of size $O(1)$. Additionally, messages of potentially larger sizes are used for (i) syncing the updated transmission probability and (ii) event counts in the beginning of each round. We describe the modifications needed to condense this communication.

Transmission probability syncing. Our analysis goes through when the server quantizes the probabilities to powers of 2 (that is, sets $p \leftarrow 2^{\min\{0,\lceil\log_2\frac{c\sqrt{k}}{\varepsilon n}\rceil\}}$. In each broadcast, the server then only sends the increase to the previous value of the negated exponent. Therefore, the total bits broadcasted, over all rounds, are

$$O(\log_2(\varepsilon N/(c\sqrt{k}))) = O(\log_2 N).$$

Since each broadcast is to k sites, this amounts to $k\log_2 N$ total bits communicated to facilitate the updates of p. Since $\mathrm{E}[R_N] = \Omega(\log N)$, syncing p does not add asymptotically to the total communication in bits.

Total event count syncing. At the end of each round, the server collects the event counts $\bar{n}_i$ from the sites $i \in [k]$. Observe that the server only needs to obtain an *estimate* of the total count $\bar{n}$ that is within a relative error of $\varepsilon/2$ (and then provide $(\varepsilon/2)$ accuracy guarantees). Obtaining an estimated count is an instance of *distributed approximate counting* (without tracking), with the added benefit that the sites (and the server) share a high probability constant-factor proxy $\bar{n}'$ for $\bar{n}$ (e.g., $\bar{n}' = \Theta(c\sqrt{k}/(\varepsilon p) + k/p)$).

It therefore suffices that each site broadcasts its value to within an additive error of $O(\max\{1, \frac{\varepsilon \bar{n}'}{k}\})$. When $n_i = O(\bar{n}')$, this requires $O(\log(k/\varepsilon)$ bits per message and $O(k \log(k/\varepsilon))$ bits per round.[8]

In the regime where $\varepsilon > 1/\sqrt{k}$, we can use weighted sampling, where sites only share their approximate count probabilistically. In this case, only $\Theta\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$ sites send messages to obtain (ε, δ)-accurate $\bar{n}$. ◀

4.3 Accuracy Analysis

We first bound the probability that the maximum relative error over all the event times in a single round exceeds ε:

▶ **Lemma 21** (Tail bound on the maximum relative error in a round)**.** *Consider the random variable that is a single round of the protocol that is started at some event count n_0. Let R be the number of events in the round and $\hat{n}$ the estimate of the event count when the event count is n. Then*

$$\Pr\left[\max_{n \in [t, t+R-1]} \frac{|n - \hat{n}|}{n} > \varepsilon\right] \le 2\exp\left(-\min\left\{\frac{c^2}{8}, \frac{c\sqrt{k}}{4}\right\}\right).$$

Proof. We use the following fact (see proof at the full version [28]).

▷ Claim 22 (Bound on the maximal partial-sum deviation [17]). Let $X_1, \ldots, X_r \overset{\text{iid}}{\sim} \text{Geom}(p)$. Define $S_i = \sum_{j=1}^{i} X_j$. Then for all $t > 0$,

$$\Pr\left(\max_{1 \le i \le r} \left|S_i - \tfrac{i}{p}\right| \ge t\right) \;\le\; 2\exp\left(-\min\left\{\frac{t^2 p^2}{8r}, \frac{tp}{4}\right\}\right).$$

We apply the claim with $t = \varepsilon\bar{n}$, $r = k$ and $p = c\sqrt{k}/(\varepsilon\bar{n})$ and thus $pt = c\sqrt{k}$ and obtain that the probability of a deviation that exceeds $\varepsilon\bar{n}$ at any of the message times (update points) in the round is at most $2\exp\left(-\min\left\{\frac{c^2}{8}, \frac{c\sqrt{k}}{4}\right\}\right)$.

Finally, observe that this bound on the maximum deviation at *message times* translates to a bound of $\varepsilon\bar{n} + 1/p = (1 + 1/(c\sqrt{k}))\varepsilon\bar{n}$ on the maximum deviation between the actual and estimated counts over *all events* in the round, since the estimate is updated in increments of $1/p$.[9] ◀

[8] This can be converted to a high probability or in-expectation bound to cover the case where the coarse estimate $\bar{n}'$ is off.

[9] technically we should apply this with the slightly smaller $\varepsilon' = \varepsilon/(1 + 1/(c\sqrt{k}))$ but we skip that to reduce clutter

4.3.1 (ε, δ)-accuracy on the first N events

Proof of Theorem 18; accuracy. From Lemma 21, using

$$c = O\left(\max\left\{\sqrt{\log\left(\frac{R}{\delta}\right)}, \frac{\log\left(\frac{R}{\delta}\right)}{\sqrt{k}}\right\}\right),$$

with probability at least $1-\delta$, the maximum relative error at all events in the first R rounds is at most ε. This follows from a per-round bound on the error probability of δ/R and a union bound on the R rounds. To obtain the bound for the first N events, we substitute for R the upper tail bound on R_N, the number of rounds in N events, given in Lemma 19. ◀

4.3.2 Per-event-index (ε, δ)-accuracy

From Lemma 21, using

$$c = O\left(\max\left\{\sqrt{\log\left(\frac{1}{\delta}\right)}, \frac{\log\left(\frac{1}{\delta}\right)}{\sqrt{k}}\right\}\right), \tag{3}$$

we obtain that the protocol provides *per-round* (ε, δ) *accuracy*, that is, for each round (independently of history), with probability at least $1-\delta$, the maximum relative error of the count at all events that fall in the round is at most ε.

To establish per-event-index (ε, δ) accuracy, we relate it to the per-round accuracy via the following (proof in Section 5):

▶ **Lemma 23** (Per-round to Per-event success bound). *For any $\pi > 0$, with $\delta(\pi) = O(\pi/\log(1/\pi))$, the protocol with per-round $(\varepsilon, \delta(\pi))$ accuracy provides (ε, π) per-event-index accuracy.*

Proof of Theorem 17; accuracy. We apply the protocol with

$$c = O\left(\max\left\{\sqrt{\log(1/\delta(\pi))}, \frac{\log(1/\delta(\pi))}{\sqrt{k}}\right\}\right).$$

Per Equation (3) this gives per-round $(\varepsilon, \delta(\pi))$ accuracy and per Lemma 23 this translates to (ε, π) per-event-index accuracy.

Now note that

$$\max\left\{\sqrt{\log(1/\delta(\pi))}, \frac{\log(1/\delta(\pi))}{\sqrt{k}}\right\} = O\left(\max\left\{\sqrt{\log(1/\pi)}, \frac{\log(1/\pi)}{\sqrt{k}}\right\}\right).$$

Therefore we can express c as stated in terms of π (with a different hidden constant). ◀

5 Per-round to per-event-index accuracy

In this section we prove Lemma 23. We derive a bound on the maximum per-event-index failure probability from the per-round failure probability δ. For the purpose of this analysis we introduce a simple stochastic renewal process which captures the relevant components of the protocol's execution.

Renewal process. Fix integers $k \geq 1$ and a constant $C > 0$. For $n \geq 1$ let

$$S_n \;=\; \sum_{i=1}^{k} X_i, \qquad X_i \overset{\text{iid}}{\sim} \text{Geom}\big(p(n)\big), \quad p(n) := \min\{1, C/n\}.$$

Write

$$\mu_n := \mathrm{E}[S_n] = \frac{k}{p(n)} = \frac{k}{C}\, n, \qquad \nu_n^2 := \mathrm{Var}(S_n) = \frac{k\,(1-p(n))}{p(n)^2}.$$

Partition the outcomes of S_n into A_n (good) and B_n (bad) with

$$\Pr(B_n) \leq \delta \quad \text{and} \quad \Pr(A_n) = 1 - \Pr(B_n) \geq 1 - \delta,$$

where the partition depends only on n.

Our renewal process starts with $N_0 = 1$ and uses $N_{j+1} = N_j + L_j$, where $L_j \sim S_{N_j} \mid A_{N_j}$ if the outcome falls in A_{N_j}, and $L_j \sim S_{N_j} \mid B_{N_j}$ otherwise. Let $T_j \in \{A, B\}$ be the label of L_j. For a time $t \geq 1$, let $J(t) = \max\{j : N_j \leq t\}$. We say that t *is bad* if $T_{J(t)} = B$.

▶ Remark 24 (Mapping to our protocol). S_n samples the length of a round that starts at event-index n. The partitioning to label B_n corresponds to a round being bad (not meeting the maximum accuracy requirement) and A_n to the round meeting the requirement.

A time t in the process corresponds to an event-index t in the protocol. When t is good, the round is good and the accuracy requirement is met. When t is bad, the round is bad and the accuracy requirement may not have been met. We aim to bound the probability that t is bad.

Tail bounds. We use the standard sub-exponential tail (Bernstein/Bennett form): for all $x \geq 0$,

$$\Pr\big(S_n - \mu_n \geq x\big) \;\leq\; \exp\!\left(-\tfrac{1}{2}\,\min\!\left\{\frac{x^2}{\nu_n^2},\; x\,p(n)\right\}\right). \tag{4}$$

Inverting (4) yields, for every n, the quantile bound

$$q_\delta(n) \;\leq\; \mu_n \;+\; 2\max\!\left\{\nu_n\sqrt{\log\frac{1}{\delta}},\; \frac{\log\frac{1}{\delta}}{p(n)}\right\}, \tag{5}$$

where $q_\delta(n)$ denotes the $(1-\delta)$-quantile of S_n.

▷ Claim 25 (Quantile-to-gap bound for t_0). Fix $t \in \mathbb{N}$ and define $t_0 < t$ to be the largest $t_0 \geq 0$ such that $\Pr\big[S_{t_0} > t - t_0\big] \;\leq\; \delta$. If t_0 is not defined we take $t_0 = 0$. Then

$$\frac{t - t_0}{\mu_{t_0}} \;\leq\; 1 \;+\; 4\,\max\!\left\{\sqrt{\tfrac{\log\frac{1}{\delta}}{k}},\; \tfrac{\log\frac{1}{\delta}}{k}\right\} \;+\; \frac{2}{t_0} = O(1 + \tfrac{\log\frac{1}{\delta}}{k}).$$

Proof. Let $s := t - t_0$ and note that by its definition $s \;\leq\; 1 + q_\delta(t_0)$. Substitute (5) and divide by μ_{t_0} to obtain the claim. ◁

We try to provide an informal overview of our approach to bounding the probability that t is bad: Observe that the probability that time t is bad is bounded by δ times the expected number of round starts before t. The issue in using this, however, is that for a large t, this expected number of round starts in $[0, t]$ can be large. In this case, our strategy is to look at a more restricted window $[t_0, t]$. We choose $t_0(t)$ so that the probability that no round starts in $[t_0, t]$ is at most δ. We then pay this δ towards our bound and only consider rounds starting in this window.

▶ **Lemma 26** (Per-t bad-probability via the window $[t_0, t]$)**.** *With t_0 as in Claim 25, let $M_t := \sum_j \mathbf{1}\{N_j \in [t_0, t]\}$ be the number of round starts in $[t_0, t]$, where (N_j) are the renewal start positions generated by the process. Then*

$$\Pr(t \text{ lies in a bad round}) \;\le\; \delta \;+\; \delta\,\mathrm{E}[M_t] \;\le\; \delta\Big(2 + \frac{t - t_0}{\mu_{t_0}}\Big).$$

Proof.

Show that the probability that no round starts in $[t_0, t]$ is at most δ. Define $f(n) := \Pr(S_n > t - n)$. Since $p(n) = \min\{1, C/n\}$ is non increasing in n, the law of S_n is stochastically increasing in n, while $t - n$ decreases; hence f is non-decreasing. Let $W := N_{J(t)}$ be the start of the round that covers t. The event "no start in $[t_0, t]$" is exactly $\{W \le t_0 \;\&\; S_W > t - W\}$, so

$$\Pr(\text{no start in } [t_0, t]) = \mathrm{E}\left[f(W)\right] \;\le\; f(t_0) \;\le\; \delta,$$

where the expectation is over the randomness of the process generating W.

Bound bad probability of t by union over starts in $[t_0, t]$. On $\{M_t \ge 1\}$, exactly one start $n \in [t_0, t]$ has its interval covering t. Let $w(n) := \Pr(\exists j :\; N_j = n)$ denote the (unconditional) probability of a start at n. By the Markov property at n and because $\Pr(B_n) \le \delta$,

$$\Pr(t \text{ bad},\, M_t \ge 1) \le \sum_{n=t_0}^{t} w(n)\,\Pr\left(B_n\right) \le \delta \sum_{n=t_0}^{t} w(n) = \delta\,\mathrm{E}[M_t].$$

Bound $\mathrm{E}[M_t]$ by coupling to an i.i.d. baseline and Wald. We show that $\mathrm{E}[M_t]$ is bounded from above by a similar process where round lengths are all S_{t_0}. Let $j^\star := \min\{j :\; N_j \ge t_0\}$ and set $H := t - N_{j^\star} \in [0, t - t_0]$. For $n \ge t_0$ we have $S_n \succeq S_{t_0}$ (stochastic dominance since $p(n)$ is non-increasing). Couple the post-t_0 interarrivals $L_{j^\star+i} \sim S_{N_{j^\star+i}}$ with i.i.d. $Y_i \sim S_{t_0}$ via common uniforms so that $L_{j^\star+i} \ge Y_i$ a.s. Let $N'(h) := \max\{m \ge 0 :\; Y_1 + \cdots + Y_m \le h\}$. Then pathwise $\#\{j : N_j \in (N_{j^\star}, t]\} \le N'(H)$, hence

$$M_t \;\le\; 1 + N'(H).$$

Conditioning on H and using Wald's bound for nonnegative i.i.d. steps, $\mathrm{E}[N'(H) \mid H] \le H/\mu_{t_0}$, so

$$\mathrm{E}[M_t] \;\le\; 1 + \mathrm{E}\left[\frac{H}{\mu_{t_0}}\right] \;\le\; 1 + \frac{t - t_0}{\mu_{t_0}}.$$

Combining. The probability that t is bad is decomposed to the events of "no round started" in the interval $[t_0, t]$ and "at least one started." From the above we obtain

$$\Pr(t \text{ bad}) \;\le\; \delta \;+\; \delta\,\mathrm{E}[M_t] \;\le\; \delta \;+\; \delta\Big(1 + \frac{t - t_0}{\mu_{t_0}}\Big) \;=\; \delta\Big(2 + \frac{t - t_0}{\mu_{t_0}}\Big).$$ ◀

Proof of Lemma 23. The claim follows by combining Claim 25 and Lemma 26: for every fixed t, we have $\Pr(t \text{ is bad}) \;\le\; O\left(\delta(1 + \frac{\log \frac{1}{\delta}}{k})\right)$. ◀

References

1 Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In *stoc21*, pages 447–455, 2021. doi:10.1145/3406325.3451041.

2 Chrisil Arackaparambil, Joshua Brody, and Amit Chakrabarti. Functional monitoring without monotonicity. In *International Colloquium on Automata, Languages, and Programming*, pages 95–106. Springer, 2009. doi:10.1007/978-3-642-02927-1_10.

3 Sepehr Assadi, Gillat Kol, and Zhijun Zhang. Rounds vs communication tradeoffs for maximal independent sets. In *FOCS*, 2022.

4 Idan Attias, Edith Cohen, Moshe Shechner, and Uri Stemmer. A framework for adversarial streaming via differential privacy and difference estimators. *CoRR*, abs/2107.14527, 2021. arXiv:2107.14527.

5 Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan R. Ullman. Algorithmic stability for adaptive data analysis. In *STOC*, 2016.

6 Amos Beimel, Haim Kaplan, Yishay Mansour, Kobbi Nissim, Thatchaphol Saranurak, and Uri Stemmer. Dynamic algorithms against an adaptive adversary: generic constructions and lower bounds. In *STOC*, 2022.

7 Omri Ben-Eliezer, Talya Eden, and Krzysztof Onak. Adversarially robust streaming via dense-sparse trade-offs. In *SOSA*, 2022.

8 Omri Ben-Eliezer, Rajesh Jayaram, David P Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. In *PODS*, 2020.

9 Aaron Bernstein. Deterministic partially dynamic single source shortest paths in weighted graphs. In *ICALP*, 2017.

10 Aaron Bernstein and Shiri Chechik. Deterministic decremental single source shortest paths: beyond the $O(mn)$ bound. In *STOC*, 2016.

11 Aaron Bernstein and Shiri Chechik. Deterministic partially dynamic single source shortest paths for sparse graphs. In *SODA*, 2017.

12 Sayan Bhattacharya, Monika Henzinger, and Giuseppe F. Italiano. Deterministic fully dynamic data structures for vertex cover and matching. In *SODA*, 2015.

13 Sayan Bhattacharya, Monika Henzinger, and Danupon Nanongkai. New deterministic approximation algorithms for fully dynamic matching. In *STOC*, 2016.

14 Sayan Bhattacharya, Monika Henzinger, and Danupon Nanongkai. Fully dynamic approximate maximum matching and minimum vertex cover in $O(\log^3 n)$ worst case update time. In *SODA*, 2017.

15 Sayan Bhattacharya and Peter Kiss. Deterministic rounding of dynamic fractional matchings. In *ICALP*, 2021.

16 Sayan Bhattacharya and Janardhan Kulkarni. Deterministically maintaining a $(2+\epsilon)$-approximate minimum vertex cover in $O(1/\epsilon^2)$ amortized update time. In *SODA*, 2019.

17 Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. doi:10.1093/ACPROF:OSO/9780199535255.001.0001.

18 Mark Braverman and Rotem Oshman. A rounds vs. communication tradeoff for multi-party set disjointness. In *FOCS*, 2017.

19 Vladimir Braverman, Avinatan Hassidim, Yossi Matias, Mariano Schain, Sandeep Silwal, and Samson Zhou. Adversarial robustness of streaming algorithms through importance sampling. In *NeurIPS*, 2021.

20 Amit Chakrabarti, Prantar Ghosh, and Manuel Stoeckl. Adversarially robust coloring for graph streams. In *ITCS*, 2022.

21 Jiecao Chen and Qin Zhang. Improved algorithms for distributed entropy monitoring. *Algorithmica*, 2017. doi:10.1007/S00453-016-0194-Z.

22 Yeshwanth Cherapanamjeri, Sandeep Silwal, David P. Woodruff, Fred Zhang, Qiuyi Zhang, and Samson Zhou. Robust algorithms on adaptive inputs from bounded adversaries. In *ICLR*, 2023.

23 Julia Chuzhoy. Decremental all-pairs shortest paths in deterministic near-linear time. In *STOC*, 2021.

24 Julia Chuzhoy, Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, and Thatchaphol Saranurak. A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In *FOCS*, 2020.

25 Julia Chuzhoy and Sanjeev Khanna. A new algorithm for decremental single-source shortest paths with applications to vertex-capacitated flow and cut problems. In *STOC*, 2019.

26 Julia Chuzhoy and Thatchaphol Saranurak. Deterministic algorithms for decremental shortest paths via layered core decomposition. In *SODA*, 2021.

27 Edith Cohen, Xin Lyu, Jelani Nelson, Tamás Sarlós, Moshe Shechner, and Uri Stemmer. On the robustness of countsketch to adaptive inputs. In *ICML*, 2022.

28 Edith Cohen, Moshe Shechner, and Uri Stemmer. A simple and robust protocol for distributed counting. *arXiv preprint arXiv:2509.05870*, 2025. doi:10.48550/arXiv.2509.05870.

29 Graham Cormode. The continuous distributed monitoring model. *ACM SIGMOD Record*, 2013.

30 Graham Cormode, Minos Garofalakis, Shanmugavelayutham Muthukrishnan, and Rajeev Rastogi. Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In *SIGMOD*, 2005.

31 Graham Cormode, S. Muthukrishnan, and Ke Yi. Algorithms for distributed functional monitoring. In *SODA*, 2008.

32 Graham Cormode, Shanmugavelayutham Muthukrishnan, Ke Yi, and Qin Zhang. Continuous sampling from distributed streams. *Journal of the ACM (JACM)*, 2012.

33 Mark Dilman and Danny Raz. Efficient reactive monitoring. *IEEE journal on selected areas in communications*, 20(4):668–676, 2002. doi:10.1109/JSAC.2002.1003034.

34 Danny Dolev and Tomás Feder. *Multiparty communication complexity*. IBM Thomas J. Watson Research Division Yorktown Heights, NY, USA, 1989.

35 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *STOC*, 2015.

36 Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *STOC*, 2010.

37 Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, 2009.

38 Hossein Esfandiari, Praneeth Kacham, Vahab Mirrokni, David P Woodruff, and Peilin Zhong. Optimal communication bounds for classic functions in the coordinator model and beyond. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1911–1922, 2024. doi:10.1145/3618260.3649742.

39 Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *NeurIPS*, 2021.

40 Maximilian Probst Gutenberg, Virginia Vassilevska Williams, and Nicole Wein. New algorithms and hardness for incremental single-source shortest paths in directed graphs. In *STOC*, 2020.

41 Maximilian Probst Gutenberg and Christian Wulff-Nilsen. Decremental SSSP in weighted digraphs: Faster and against an adaptive adversary. In *SODA*, 2020.

42 Maximilian Probst Gutenberg and Christian Wulff-Nilsen. Deterministic algorithms for decremental approximate shortest paths: Faster and simpler. In *SODA*, 2020.

43 Moritz Hardt and Jonathan R. Ullman. Preventing false discovery in interactive data analysis is hard. In *FOCS*, 2014.

44 Avinatan Hassidim, Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Adversarially robust streaming algorithms via differential privacy. In *NeurIPS*, 2020.

45 Jacob Holm, Kristian De Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM (JACM)*, 2001.

46 Zengfeng Huang, Xuemin Lin, Wenjie Zhang, and Ying Zhang. Communication-efficient distributed covariance sketch, with application to distributed pca. *Journal of Machine Learning Research*, 22(80):1–38, 2021. URL: `https://jmlr.org/papers/v22/20-705.html`.

47 Zengfeng Huang, Zhongzheng Xiong, Xiaoyi Zhu, and Zhewei Wei. Simple and optimal algorithms for heavy hitters and frequency moments in distributed models. In *STOC*, 2025.

48 Zengfeng Huang and Ke Yi. The communication complexity of distributed epsilon-approximations. *SIAM Journal on Computing*, 46(4):1370–1394, 2017. `doi:10.1137/16M1093604`.

49 Zengfeng Huang, Ke Yi, and Qin Zhang. Randomized algorithms for tracking distributed count, frequencies, and ranks. In *PODS*, 2012.

50 Haim Kaplan, Yishay Mansour, Kobbi Nissim, and Uri Stemmer. Separating adaptive streaming from oblivious streaming using the bounded storage model. In *CRYPTO*, 2021.

51 Ram Keralapura, Graham Cormode, and Jeyashankher Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In *SIGMOD*, 2006.

52 Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

53 Danupon Nanongkai and Thatchaphol Saranurak. Dynamic spanning forest with worst-case update time: adaptive, Las vegas, and $O(n^{1/2-\epsilon})$-time. In *STOC*, 2017.

54 Danupon Nanongkai, Thatchaphol Saranurak, and Christian Wulff-Nilsen. Dynamic minimum spanning forest with subpolynomial worst-case update time. In *FOCS*, 2017.

55 Jeff M Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *SODA*, 2012.

56 Menachem Sadigurschi, Moshe Shechner, and Uri Stemmer. Relaxed models for adversarial streaming: The bounded interruptions model and the advice model. In *ESA*, 2023.

57 Thomas Steinke and Jonathan Ullman. Tight lower bounds for differentially private selection. In *FOCS*, 2017.

58 Srikanta Tirthapura and David P Woodruff. Optimal random sampling from distributed streams revisited. In *DISC*, 2011.

59 Jan van den Brand, Yu Gao, Arun Jambulapati, Yin Tat Lee, Yang P. Liu, Richard Peng, and Aaron Sidford. Faster maxflow via improved dynamic spectral vertex sparsifiers. In *STOC*, 2022.

60 Emanuele Viola. The communication complexity of addition. *Combinatorica*, 35(6):703–747, 2015. `doi:10.1007/S00493-014-3078-3`.

61 David Wajc. Rounding dynamic matchings against an adaptive adversary. In *STOC*, 2020.

62 David P Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *STOC*, 2012.

63 David P Woodruff and Qin Zhang. When distributed computation does not help. *CoRR, abs/1304.4636*, 5, 2013.

64 David P Woodruff and Qin Zhang. An optimal lower bound for distinct elements in the message passing model. In *SODA*, 2014.

65 David P. Woodruff and Samson Zhou. Tight bounds for adversarially robust streams and sliding windows via difference estimators. In *FOCS*, 2021.

66 Hao Wu, Junhao Gan, and Rui Zhang. Learning based distributed tracking. In *KDD*, 2020.

67 Christian Wulff-Nilsen. Fully-dynamic minimum spanning forest with improved worst-case update time. In *STOC*, 2017.

68 Zhongzheng Xiong, Xiaoyi Zhu, and Zengfeng Huang. Adversarially robust distributed count tracking via partial differential privacy. In *NeurIPS*, 2023.

69 Ke Yi and Qin Zhang. Optimal tracking of distributed heavy hitters and quantiles. In *PODS*, 2009.