Scalable Approximate Query Tracking over Highly Distributed Data Streams with Tunable Accuracy Guarantees

Nikos Giatrakos[a,b,*], Antonios Deligiannakis[a,b], Minos Garofalakis[a,b], Daniel Keren[c], Vasilis Samoladas[a,b]

[a]ATHENA Research and Innovation Centre, Artemidos 6 & Epidavrou, 15125 Athens, Greece
[b]School of Electrical and Computer Engineering, Technical University of Crete, University Campus., 73100 Chania, Greece
[c]Department of Computer Science, Haifa University, 31905 Haifa, Israel

Abstract

The recently-proposed Geometric Monitoring (GM) method has provided a general tool for the distributed monitoring of arbitrary non-linear queries over streaming data observed by a collection of remote sites, with numerous practical applications. Unfortunately, GM-based techniques can suffer from serious scalability issues with increasing numbers of remote sites. In this paper, we propose novel techniques that effectively tackle the aforementioned scalability problems by exploiting a carefully designed sample of the remote sites for efficient approximate query tracking. Our novel sampling-based scheme utilizes a sample of cardinality proportional to $\sqrt{N}$ (compared to N for the original GM and its variants), where N is the number of sites in the network, to perform the monitoring process. Our extensive experimental evaluation and comparative analysis over a variety of real-life data streams demonstrates that our sampling-based techniques can significantly reduce the communication cost during distributed monitoring with controllable, predefined accuracy guarantees. In that, we manage to scale the monitoring of any given non-linear function on much higher network scales which had not been reached by any GM related method or variant so far.

Keywords: distributed function tracking, data streams, sampling

1. Introduction

Efficient data stream processing algorithms have become an integral part of real-time monitoring applications, from network traffic monitoring to financial or stock data analysis and sensor data querying. Streaming tuples are rapidly produced in a number of geographically dispersed sites (routers, ATMs, sensor nodes etc) and are *continuously* processed online to provide continuous up-to-date query answers destined to support decision making procedures such as DDoS attacks, fraudulent transactions, market trend predictions, and tsunami wave detection, in a timely manner. In such distributed settings, it is imperative to design efficient algorithms that reduce the communication burden during the continuous monitoring process [1, 2], since either the available bandwidth is limited, or data transmission is a crucial factor that reduces network lifetime (e.g., for battery-powered sensor nodes [3]).

The problem of efficiently tracking the value of a function (often compared to some predefined threshold) over the union of local streams in a large-scale distributed system, lies at the core of several recent research efforts [4, 5, 2, 6, 7, 8]. Monitoring tasks may involve functions that are simple linear aggregates, such as checking whether the sum of a distributed set of variables exceeds a predetermined threshold [9, 8], thresholded counts of items [10] or frequently occurring items in a set of distributed streams [11]. More complicated function monitoring may involve holistic aggregates [4, 8], self-join as well as stream-join operations [6, 12], or general, *non-linear* function tracking [5].

The original work of [5] is the first to propose a generic, Geometric Monitoring (GM) method for monitoring *any non-linear function f* over the global average of vectors maintained at the distributed sites, with respect to some

---

Figure 1: Illustration of GM at a given time point t. The monitored convex hull is depicted in gray, while the position of e and the current $v(t)$ are shown as well. Black spheres refer to the local constraints constructed by sites. Since none of them crosses the threshold surface, no synchronization is needed.

threshold T, i.e., monitoring whether $f(\cdot) \lessgtr T$. The GM method is a very powerful technique that has already been exploited in a wide range of applications, including: (i) outlier detection in sensor networks [13], where the monitored function is any of the L_p norms, cosine similarity, extended Jaccard coefficient, or correlation coefficient; (ii) tracking range, norm-aggregate and join-aggregate queries [12, 14] over distributed data streams; (iii) monitoring fragmented skyline queries [15]; (iv) detecting machines that are about to become faulty in data centers [16]; and, (v) distributed online prediction [17] by dynamically monitoring the accuracy of distributed local models. In a nutshell, the GM method can offer a general solution to any non-linear function tracking task expressed over a combination (sum, average, convex combination) of vector data collected by distributed data sources, in which continuous data communication to a central site is not feasible, due to either bandwidth or energy constraints.

In this work, we demonstrate that, despite the generic nature of the GM method as a distributed tracking scheme, GM together with its recently extended versions [18, 19, 14] face significant scalability issues as the number of remote sites increases. We then develop scalable approximate monitoring techniques for general, non-linear function tracking with tunable accuracy guarantees. We start by presenting the GM method and the problems it faces when the number of sites increases.

1.1. Geometric Monitoring Basics

As in previous works [4, 5, 20, 6, 21], we assume a distributed, two-tiered setting, where data arrives continuously at N geographically dispersed sites. At the top tier, a central coordinator exists that is capable of communicating with every site, while pairwise site communication is only allowed via the coordinator. Each site S_i, $i \in [1..N]$ participating at the bottom tier periodically receives updates on its local stream and maintains a d-dimensional *local measurements vector* $v_i(t)$, capturing the current state of its local stream. The *global measurements vector (i.e., stream)* $v(t)$ at any given timestamp t, is defined as the average of the $v_i(t)$ vectors, $v(t) = \frac{\sum_{i=1}^{N} v_i(t)}{N}$. The coordinator aims to continuously monitor whether the value of a function $f(v(t))$, parameterized by the global average $v(t)$, lies above/below a given threshold T. We term the part of the input domain where $f(v(t)) = T$ as the *threshold surface* and, for simplicity, we henceforth use T to refer to either the threshold surface or the function value on the threshold surface.

Assume that at a previous time instant t_s, the coordinator has collected the local $v_i(t_s)$ vectors. Using $e(t_s)$ to distinguish the global average $v(t_s)$ at t_s, the coordinator computes $e(t_s) = \frac{\sum_{i=1}^{N} v_i(t_s)}{N}$ at that time, subsequently broadcasting $e(t_s)$ to the sites in the bottom tier. The previous process is referred to as a *synchronization* step. Note that, until the next synchronization, the coordinator's view of the global vector is fixed at $e(t) = e(t_s)$. Following [5], upon receiving $e(t)$, sites keep receiving updates of their local streams and accordingly maintain their $v_i(t)$ vectors. At any given timestamp, each site S_i individually computes a *deviation vector* $\Delta v_i(t) = v_i(t) - v_i(t_s)$, which depicts the change that the local vector has undergone since t_s. By adding the deviation vector to $e(t)$, sites compute their *drift vectors* as $e(t) + \Delta v_i(t)$. Since $v(t) = \frac{\sum_{i=1}^{N} v_i(t)}{N} = e(t) + \frac{\sum_{i=1}^{N} (v_i(t) - v_i(t_s))}{N} = \frac{\sum_{i=1}^{N} (e(t) + \Delta v_i(t))}{N}$, $v(t)$ is a convex combination of the drift vectors.

2

Consequently, $v(t)$ will always lie in the convex hull formed by the $\Delta v_i(t)$ vectors translated by $e(t)$, as depicted in Figure 1 for $d = 2$, $N = 5$: $v(t) \in Conv\ (e(t) + \Delta v_1(t), \dots, e(t) + \Delta v_N(t))$. If the convex hull does not intersect the *inadmissible* part of the input domain (on the right of Figure 1), where the monitored inequality is reversed (from $f(v(t)) > T$ to $f(v(t)) < T$ or vice versa), it is assured that $v(t)$ cannot lie in that part either. Hence, our monitoring problem is transformed to the question of how to decide in a distributed manner whether the convex hull intersects the threshold surface.

It has been proven [5] that if the sites locally construct hyperspheres $B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$, centered at $e(t) + \frac{1}{2}\Delta v_i(t)$ with radius $\frac{1}{2}\|\Delta v_i(t)\|$, then:

$$Conv(e(t) + \Delta v_1(t), \dots, e(t) + \Delta v_N(t)) \subset \bigcup_{i=1}^{N} B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$$

That is, the union of these hyperspheres is always guaranteed to cover the convex hull of the translated local drifts (this holds for vectors of any dimension). Thus, having constructed $B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$, each site individually checks for an intersection of its local sphere with the threshold surface. In case the intersection is non-empty in at least one S_i, a *local violation* occurs at S_i, indicating that the convex hull and, thus, $v(t)$ **may** have crossed the threshold surface. Hence, a synchronization takes place where the coordinator collects the $v_i(t)$ vectors and assesses whether $f(v(t))$ truly switched sides ($\lessgtr$) with T. It then computes the new $e(t)$ and communicates it back to the sites. From this point forward, the tracking process can proceed as described above. If no local violation occurs (as in the example of Figure 1), then no communication is necessary.

Communication savings are achieved by postponing a synchronization until some site's local sphere intersects the threshold surface. Note that the convex hull or its superset, the union of local spheres, may cross the threshold surface, while the actual position of $v(t)$ may not be in the intersection. As a result, *the framework may cause synchronizations when $f(v(t))$ has not crossed the threshold, termed False Positives (FPs)*.

Example 1. *Consider a news monitoring scenario as the ones used in [5, 21, 18, 19]. The focus of the tracking process is to monitor the coherence of a specific term with a given document category within a predefined window of w observations per site. Individual sites receive news stories tagged as belonging to a particular category and also mark those stories including the term of interest. The Mutual Information (MI) [22] function is used to measure the relevance of a specific (term, category) pair. In our running example, we initially assume a setup composed of $N = 5$ sites and we are interested in monitoring (the threshold is used to facilitate - keep less dependent on N - the illustrations)*

$$log(\frac{v^1(t) \cdot w \cdot N}{(v^1(t) + v^3(t)) \cdot (v^1(t) + v^2(t))}) > log(N) + 0.01$$

using $w = 20$, i.e. each site considers the $w = 20$ most recent <term,category> pairs it received and globally 100 observations are included in the computation. The global vector $v(t) = [v^1(t), v^2(t), v^3(t)]$ is an average vector over all sites and is composed of $d = 3$ dimensions as shown in the formula: the first dimension $v^1(t)$ is the average of local counts $v_i^1(t)$s (i.e., $v^1(t)$ at site S_i) of the number of <term,category> co-occurrences, $v^2(t)$ averages the local counts $v_i^2(t)$s of news stories including the term without being tagged by the tracked category and $v^3(t)$ averages the local counts $v_i^3(t)$s of news stories including the category occurrences without the tracked term.

Assume a site S_i receives an update of the form $[1, 1, 1]$ (term, category co-occurrence), immediately after the last synchronization. Further assume that the oldest update in the window that is removed upon the new arrival is $[0, 0, 0]$. Then, $\Delta v_i = [1, 1, 1]$ and $\|\Delta v_i\| = \sqrt{3}$. Now, a second update $[0, 1, 0]$ (the tracked termed occurred without the category) is received at S_i, while the oldest, expired update is now $[0, 0, 1]$. Then $\Delta v_i = [1, 2, 0]$ and $\|\Delta v_i\| = \sqrt{5}$.

Figure 6(a)[1] shows the threshold surface corresponding to $T = log(N) + 0.01$ for our running example and instantiates a convex hull along with the inscribed hyperspheres. All details about the participating site vectors are included in the caption of the figure. □

---

[1] All illustrations in Figure 6 were created using the Mathematica Software: Wolfram Research, Inc., Mathematica, Version 10.0, Champaign, IL (2014).

| (a) | (b) | (c) |
| $Conv(e+\Delta v_1, \ldots, e+\Delta v_5)$ | $Conv(e+\Delta v_1, \ldots, e+\Delta v_{10})$ | $Conv(e+\Delta v_1, \ldots, e+\Delta v_{100})$ |
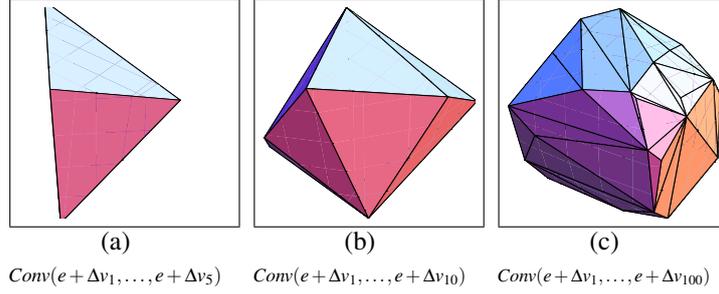
Figure 2: The effect of network scale on the monitored area ($d = 3$). All Δv_i vectors are randomly chosen from the unit cube, the front view of which is the box included in each figure. As the network scale increases, the length of more Δv_i vectors increases. Inevitably, the volume of the convex hull that needs to be monitored increases, resulting in more FP alerts.

1.2. Existing Scalability Issues

We now explain why the GM framework may result in increased communication in either highly distributed networks, or in cases where the monitored function is parameterized with the sum (as opposed to the average) of the local measurements vectors.

High N values $\Rightarrow$ proneness to FP synchronizations. As already described, the set that GM tracks is the convex hull $Conv(e(t)+\Delta v_1(t), \ldots, e(t)+\Delta v_N(t))$. It is not difficult to see that the more sites participate in the distributed monitoring process (and, thus, contribute their $e(t)+\Delta v_i(t)$ in the formation of this convex hull), the larger the tracked region will be. Moreover, the hyperspheres maintained by each site in order to include the expanded convex hull will cover an even larger, compared to the expanded convex hull, area of the input domain because $Conv(e(t)+\Delta v_1(t)$,

$\ldots, e(t)+\Delta v_N(t))$ is a subset of $\bigcup\limits_{i=1}^{N} B(e(t)+\frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$.

This raises the potential for a larger number of FP synchronization decisions as the number of sites increases. In Figure 1, sites S_1, S_2 require a simple triangular area to be monitored, which is continuously expanded as $S_3 - S_5$ contribute vectors and inscribe respective spheres. In higher dimensions this trend is even more evident as more site vectors append convex polytopes and facets to the overall convex hull. For instance, Figure 2 schematically exhibits the effect of progressively increasing the network scale from $N = 5$ sites to $N = 10$ and $N = 100$ in a 3-d space by randomly picking additional Δv_is from the unit cube. Larger values of N yield a convex hull that tends to cover the entire unit cube (boxed area in Fig. 2).

We stress that the cost of an FP synchronization is equivalent to $N + 1$ messages, assuming the coordinator is equipped with broadcast capabilities, or $2N$ otherwise, and that all sites are required to participate in this process. This not only increases the total communication cost per FP as the number of sites N increases, but also increases the cost per site, since a site transmits messages each time at least one network site exhibits a local violation.

The challenge of monitoring sum-parameterized functions. The scalability issues that arise in case of monitoring sum-parameterized functions, i.e., $f(v_{sum}(t)) \lessgtr T$, with $v_{sum}(t) = N \cdot v(t) = \sum\limits_{i=1}^{N} v_i(t)$, are much more pronounced compared to the average case $f(v(t))$. As we show in our detailed study (Section 7), this is because, apart from having more sites contributing vectors to form the convex hull (as happens with average input), sum-parameterization requires all site drift vectors to be scaled proportionally to N. Hence, the size of the tracked convex hull and of the covering hyperspheres increases with the network scale, which makes it much more prone to false positives.

1.3. Contributions

Our contributions can be summarized as follows:
• As in our analysis above, we point out the limitations of the GM approach in highly distributed environments, that result in excessive false positive data centralization decisions and prohibitive communication cost.
• Having identified the degree of distribution (N) as a key limitation of GM based techniques, we introduce an algorithmic framework that exploits a small sample (proportional to only $O(\sqrt{N})$) of the sites to perform the monitoring process, and we formally study the properties of our sampling-based geometric scheme, as well as how it addresses

the scalability issues of GM. In a nutshell, our scalable approximate monitoring algorithm exploits *Horvitz-Thompson sampling estimators* [23, 24] over a carefully built sample of the sites in order to construct controllable-error approximations of the average vector; furthermore, it employs multidimensional tail-probability bounds and thorough geometric analysis to control the effect of these approximations on the accuracy of GM. Our approach can considerably decrease the amount of false positive data centralizations and the communication burden on the network at the cost of potentially causing a few, controllable False Negatives (FNs), i.e., missing a true threshold crossing of the monitored function. As we theoretically prove and practically demonstrate, such missed *threshold violations* are quickly corrected (by appropriate detection) in the immediate aftermath, and their rate can be tuned, thus being acceptable in monitoring applications both in broader classes of data stream monitoring algorithms [25, 26, 4, 2, 6, 20] and GM related applications [13, 12, 14, 15, 16, 17].

• We exhibit the applicability of our sampling-based techniques on the latest advancements regarding GM-based monitoring, the convex safe zone concept introduced in [14, 27], which still faces scalability issues. However, our contributions advance beyond that:

  • We present a novel lemma, tailored for the convex safe zone concept, enabling an exact mapping of the monitoring task from a multidimensional input domain of arbitrary dimension to a unidimensional one. Our mapping can practically reduce the size of communicated data during a false positive centralization proportionally to the dimensionality of the initial monitored space.

  • To achieve scalability while exploiting and retaining the advantages of our unidimensional mapping, we revisit our sampling-based GM scheme, in the unidimensional setting. The revised $(1-d)$ version of our scheme, tailored for the convex safe zone concepts, enables additional optimizations as it reduces the approximation error of the monitored quantity and can both in expectation and also in practice (Section 6.6) achieve to even further reduce the number of false (FP,FN) decisions throughout the tracking process.

• We present a thorough experimental study using two real datasets, a variety of different functions, threshold values and network sizes that no previous GM variant has reached. Our performance comparisons are against not only the vanilla GM approach [5], but also against approaches orthogonal to ours, including the balancing optimization proposed in [5], the recently proposed prediction-based GM [18, 19] and the convex safe zone ideas of [14, 27]. In highly distributed environments our algorithms can ensure, in most cases, *one and up to two orders of magnitude fewer transmitted messages* compared to the rest of the candidate algorithms, while also significantly reducing the *per site* communication cost, even without exploiting any of these methods' orthogonal optimizations within our proposed schemes.

• We provide a thorough, comprehensive study on sum- (instead of average- commonly handled by GM) parameterized function monitoring which has not been explicitly discussed in GM related works. We describe alternative methods for handling such functions and discuss their effect from the scalable monitoring perspective.

Table 1 summarizes the main notations used in this paper.

2. Scalable Query Tracking

Having shown that a high degree of distribution N renders GM inefficient, we design a sampling-based framework that overcomes scalability issues. In Section 2.1 we formally present a set of requirements that any candidate sampling-based scheme for GM monitoring should abide by, to both provide communication efficiency and compliance with application-defined accuracy requirements. In Section 2.2, we detail our generic sampling-based GM scheme.

The intuition behind our approach is that, instead of monitoring the entire convex hull formed by the N sites as GM does (Sections 1.1 and 1.2), we choose to track a narrower convex hull composed of a carefully-crafted random sample of sites. The latter sample of sites constructs a subset of $Conv(\{e(t) + \Delta v_i\} : \forall S_i \in \{S_1, \ldots, S_N\})$ reducing the tracked space and warding off FP synchronizations.

2.1. Efficiency and Accuracy Requirements

In addition to choosing fewer sites to reduce the monitored region, our sampling-based geometric scheme should have the potential to guarantee improved communication efficiency at time t by ensuring that, if sampling-based

Table 1: Frequently Used Symbols

Symbol	Description
N	The number of sites of the bottom tier
M	The number of sampling trials per update cycle
$K_\mu(t), K(t)$	$K_\mu(t)$: Sample of sites participating in the μ-th sampling trial at time t, $K(t) \equiv K_1(t)$, $1 \le \mu \le M$
$\|y\|$	L_2 norm of vector y
d	Dimensionality of the input domain
S_i	The i-th site, $S_i \in \{S_1, \dots, S_N\}$
t_s	Time-point of the last synchronization
$v_i(t)$	Local measurements vector at S_i at time t
$\Delta v_i(t)$	Deviation vector at S_i at time t $(= v_i(t) - v_i(t_s))$
$v(t), \hat{v}(t)$	Global average & its statistic estimator using $M = 1$ at time t
$\hat{v_\mu}(t)$	Estimator of $v(t)$ provided by the μ-th sampling trial at time t, $\hat{v}(t) = \hat{v_1}(t)$
$e(t) = e(t_s) = v(t_s)$	Global average vector at t_s, unchanged at time t unless a synchronization occurs.
$B(c, \rho)$	Hypersphere centered at c with radius equal to ρ
(ε, δ)	Approximation parameters denoting that $\hat{v}(t)$ lies within ε distance from $v(t)$ with probability at least $1 - \delta$
$g_i(t)$	Sampling function $0 \le g_i \le 1$ for site S_i at time t
ε_T	Minimum distance of $e(t)$ from threshold surface
$d_C(y)$	Signed distance of vector y from convex subset C of the admissible region

monitoring is applied instead of the basic GM method at a given timestamp, its inscribed local constraints are fully contained within the local constraints of the original GM $\bigcup_{i=1}^{N} B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$ (Section 1.1). This guarantees that, at time t, the region tracked by the sampled sites does not cross the threshold surface before the union of balls of the conventional GM does and, thus, additional FP synchronizations **cannot** be caused.

Requirement 1. **[Efficiency]** *The area of the input domain monitored by applying a candidate sampling-based tracking scheme at a given time point should be included in* $\bigcup_{i=1}^{N} B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$.

Since the monitoring process utilizes a small sample (subset) of the sites available in the network, it will be approximate in nature. At any given time t, our sampling-based geometric techniques monitor an *unbiased* (i.e., $E(\hat{v}(t)) = v(t)$) estimation $\hat{v}(t)$ of the true global average $v(t)$ originating from only a sample $K(t) \subseteq \{S_1, \dots S_N\}$ of sites' local vectors. We wish to keep the estimation error controllable and tunable based on *a priori* defined accuracy requirements. To control the approximation error, we employ an (ε, δ) approximation scheme. More precisely, for a priori given $0 < \delta \le 1$, $\varepsilon > 0$ we shall require $v \in B(\hat{v}(t), \varepsilon)$ with high probability, at least $1 - \delta$.

Requirement 2. **[Approximation Quality]** *At any given time, the estimation $\hat{v}(t)$ monitored by the sampled sites should not exceed an $\varepsilon-$ distance from the true $v(t)$, with high probability $1 - \delta$; that is, for application defined $0 < \delta \le 1$, $\varepsilon > 0$: $P(v(t) \notin B(\hat{v}(t), \varepsilon)) \le \delta$.*

Due to the fact that $v(t)$ is monitored in an approximate manner, tuned according to (ε, δ), it is possible that a synchronization is prevented while $v(t)$ truly switched side with respect to the threshold surface. Our sampling-based scheme should enable applications to explicitly tune the probability P_{FN} of such False Negative (FN) events. This requires an additional (application-defined) input parameter apart from (ε, δ).

Nonetheless, specifying a triplet of parameters, two of which refer to the input domain rather than the function value, should be carefully considered from an application's viewpoint. Application accuracy requirements should be

expressed in a simple way that abstracts the actual details of the input domain of the tracking process. While it is more natural for the end user to directly specify P_{FN}, for ease of presentation we assume that the parameter specified is the δ parameter of Requirement 2. We demonstrate in Section 3 that P_{FN} is directly linked to δ. Then, our scheme can accordingly tune not only the probability of a FN, i.e., P_{FN}, but also the approximation quality in the (ε, δ) scheme. To do so, ε should be expressed as a function of δ, i.e., $\varepsilon = \varepsilon(\delta)$.

Requirement 3. [Tunable Accuracy] *At any given time, the proposed sampling-based monitoring algorithm should possess the ability to receive a single tolerance value $0 < \delta \leq 1$ and self-tune its Approximation Quality i.e., (ε, δ) and FN rate i.e., P_{FN}.*

Hence, we assume that the application expresses its monitoring needs in the form: $f(v(t)) \gtrless T, \delta : 0 < \delta \leq 1$. We then proceed in describing the generic operation of our sampling-based framework.

2.2. Our Generic Sampling-Based Scheme

We now present our sampling-based GM algorithm and demonstrate how it satisfies Requirements 1-3. A key idea in our scheme is to independently sample each site S_i with a different probability $g_i(t)$ that depends on various factors. Our discussion in this section assumes that these sampling probabilities $g_i(t)$ have been determined, deferring the analysis of the $g_i(t)$ computation to Section 3.

Algorithmic Sketch. Our proposed framework divides its operation into three phases, namely (a) initialization, (b) monitoring (or tracking) and (c) synchronization phases. The initialization phase is executed only once upon the reception of the function tracking query. The tracking phase and (if necessary) synchronization phases of the generic sampling-based algorithm we describe below are executed in regular data *update cycles* in the network.

These update cycles may correspond to slides of sliding windows declared by the application queries as is common in streaming applications [28, 29], may involve epochs in sensor network settings [3] or may be defined based on a global Poisson parameter expressing the data arrival rate across the network i.e., such that our algorithm is executed when a data tuple has arrived in at least one S_i. These update models also account for the application (experimental) results that have been presented in all works related to the GM approach [5, 21, 18, 19, 14, 27, 13]. Hence, the temporal reference t expresses the current execution of the monitoring and (if necessary) the synchronization phase, after an update cycle. Formally:

- *Initialization Phase:* This is a first, full synchronization phase (see below) that is triggered upon the reception of the application query, rather than being caused by some local violations. In the end of this phase, $t_s, v_i(t_s), e(t)$ have been appropriately set across the network.

- *Monitoring (or Tracking) Phase:* In this phase, initially each S_i receives updates of its local vector $v_i(t)$ (after a window slide, epoch expiration or data arrival rate-based time interval) and computes $\Delta v_i(t) = v_i(t) - v_i(t_s)$. At the beginning of each monitoring phase, the sample of sites participating in the tracking process is empty, i.e., $K(t) = \emptyset$ (for all M trials mentioned below). Then, in order to determine if $S_i \in K(t)$, each site independently flips a biased coin with success probability of $g_i(t)$, where $g_i(t) \in [0, 1]$ is a sampling function independently computed by each site. In practice this corresponds to having each S_i generate a random number $\rho_i(t)$ and check whether $\rho_i(t) \leq g_i(t)$ (in which case $S_i \in K(t)$) or not (so that $S_i \notin K(t)$). In our scheme S_i constructs a local constraint $B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$ only if it finds itself included in the sample $K(t)$ of sites participating in the monitoring process, i.e., $S_i \in K(t)$, and this sphere is checked for threshold crossing. In case at least one $S_i \in K(t)$ detects a threshold crossing of its local constraint (hypersphere), it calls for a synchronization. This process may be repeated $1 \leq M \leq \left\lceil \frac{log(0.01)}{log(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N})} \right\rceil$ times (independent trials where $\rho_{i1}(t), \ldots, \rho_{iM}(t)$ random numbers are checked against $g_i(t)$), independently at each site $S_i \in \{S_1, \ldots, S_N\}$.

  Remarkably, the $g_i(t)$ we propose in Section 3 takes into account $\|\Delta v_i(t) = v_i(t) - v_i(t_s)\|$ as built since the last synchronization at t_s. Therefore, even if S_i receives no update at the current execution of the monitoring phase (thus $v_i(t)$ lately remains steady, but can still deviate from $v_i(t_s)$), it still has a chance (proportional to $\|\Delta v_i(t)\|$ in Section 3) to enter $K(t)$.

- *Synchronization (Partial or Full) Phase:* Apart from performing the monitoring only with the sampled sites, our technique may prevent the cost of a full synchronization by applying a *partial* one, using $K(t)$.

  - *Partial* Synchronization: the coordinator initially broadcasts a message requiring only the sampled sites of one, say the first, out of M trials to contribute their $\Delta v_i(t)$ vectors. Using these vectors, it derives an unbiased estimate $\hat{v}(t)$ (using Estimator 1 discussed shortly) of $v(t)$ and checks (Requirement 2) whether $B(\hat{v}(t), \varepsilon)$ crosses T. If it does not, the tracking continues unaffected and a full synchronization is prevented. Otherwise, a full synchronization follows.

  - *Full* Synchronization: also sites that did not include themselves in the first sampling trial contribute their local vectors, for a new $e(t)$ to be computed. The coordinator sets $t_s = t$, computes $e(t) = e(t_s)$ and broadcasts $e(t)$ to the sites. Upon receiving $e(t)$, sites set $t_s = t$, $v_i(t_s) = v_i(t)$. Thus, $e(t)$ constitutes the last known global average data vector and $t_s, v_i(t_s)$ have been appropriately set across the network. $t_s, v_i(t_s), e(t)$ remain unchanged until another full synchronization takes place.

According to our above description, if $B(\hat{v}(t), \varepsilon)$ does not cross T, only a partial synchronization is performed. This is because the coordinator deduces that the local violation at some sites caused an FP alarm with high probability $(1 - \delta)$ (Requirement 2). On the contrary, if $B(\hat{v}(t), \varepsilon)$ crosses T, a full synchronization is required because the coordinator believes (again based on Requirement 2) that a true threshold violation may have taken place and probes the rest of the sites so as to compute the exact value of $e(t)$. The latter is required to avoid an additive error in the approximation of $\hat{v}(t)$ (see Estimator 1) as the tracking process continues.

We now provide the details of our monitoring scheme, explain our design choices and discuss the accuracy guarantees of our technique. In each of these steps, we examine the satisfiability of Requirements 1-3 in conjunction with our algorithmic sketch. For ease of exposition, we start our discussion with Requirement 2. For the same reason, we henceforth omit the temporal reference t as we refer to theoretic details of a single instance of the monitoring and (if necessary) synchronization phases. We also elaborate on how our tunable accuracy guarantees evolve over time separately in Section 3.

Monitored Estimator and Approximation Quality Requirement. Let us assume that each S_i performs a single sampling trial for now (i.e., $M = 1$, practically we show $M = 2$ to 4), first because during a partial synchronization the coordinator asks only for the vectors sampled in the first trial and second because the utility of more trials will be introduced and discussed later on. Consider a multivariate random variable $\Delta' v_i = \frac{\Delta v_i}{g_i}$ with probability g_i, and zero otherwise. Notice that the expected value $E[\Delta' v_i]$ is a $d-$dimensional vector and $E[\Delta' v_i] = [E[\Delta' v_i^1], \cdots, E[\Delta' v_i^d]]$, where $\Delta' v_i^j$ denotes the j-th component (dimension) of the vector $\Delta' v_i$. We demonstrate in Lemma 1 that, based on the drift vectors $e + \frac{\Delta v_i}{g_i}$ of the set K (of one out of M trials), an unbiased estimate $\hat{v}$ of the global average v can be derived at any given time stamp t utilizing a *Horvitz-Thompson Estimator* [23, 24]:

$$\hat{v} = e + \frac{\sum_{i=1}^N \Delta' v_i}{N} = e + \frac{\sum\limits_{S_i \in K} \frac{\Delta v_i}{g_i}}{N} \tag{1}$$

Note that the global average is $v = e + \Delta v$, with $\Delta v = \sum\limits_{i=1}^N \Delta v_i / N$. Hence, Estimator 1 estimates Δv as $\hat{\Delta v} = \sum\limits_{S_i \in K} \frac{\Delta v_i}{g_i} / N$. The estimator weighs each sampled site with $1/g_i$. The reason for this is fairly intuitive: if site S_i, which is sampled with probability g_i, individually appears in the sample, then, on average, we expect to have $1/g_i$ sites with similar probabilities in the full population (since $g_i \cdot 1/g_i = 1$); thus, the single occurrence of S_i in the sample is essentially a "representative" of $1/g_i$ sites in the full population [30, 24].

Lemma 1. *For Estimator 1 the following hold:*
(a) Estimator 1 is an unbiased estimator of v, i.e., $E[\hat{v}] = v$, when sampling $\forall S_i \in \{S_1, \ldots, S_N\}$ with $0 \leq g_i \leq 1$.
(b) $E[\hat{v}] \in Conv(\ e + \Delta v_1, \ldots, e + \Delta v_N)$
(c) $\hat{v} \in Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$

Proof. In Appendix. □

Since Estimator 1 is unbiased, we can utilize tail inequalities [26] to satisfy Requirement 2. Note that we do not assume independence of individual dimensions of either local, or global vectors that we examine. Nonetheless, according to our algorithmic sketch, S_i's independently decide to include themselves in K or not, based on g_i. The Vector Bernstein's Inequality [31] (presented below) will be particularly useful in our subsequent analysis.

Vector Bernstein's Inequality [31]. Let $y_1, \ldots, y_N$ be independent random vectors with $E[y_i] = 0$. Let $B > 0$ denote an upper bound on $\|y_i\|$ (i.e., $\|y_i\| \leq B$), and let $\sigma^2 \geq \sum_{i=1}^{N} E[\|y_i\|^2]$. Then, for all $0 < \delta \leq 1$ and $0 \leq \varepsilon \leq \sigma^2/B$ such that[2] $\varepsilon = (1 + \sqrt{\ell n(1/\delta)}) \cdot \sigma$:

$$P(\|\sum_{i=1}^{N} y_i\| \geq \varepsilon) \leq \delta \tag{2}$$

The inequality states that if we add N random vectors of bounded length whose expectation is the zero vector, their sum will produce a vector placed near (no farther than ε) to zero with probability at least $1 - \delta$. The proximity (ε) of the vector sum to zero depends on an upper bound σ on the overall standard deviation[3] and the chosen probability bound δ. Note that the above bound does not depend on the dimensionality d of the vectors. In our case, each y_i corresponds to $\frac{\Delta' v_i - \Delta v_i}{N}$. Moreover, $B \geq \{\|\frac{\Delta v_i}{N}\|, \|\frac{\Delta v_i}{g_i \cdot N} - \frac{\Delta v_i}{N}\|\} \, \forall S_i \in \{S_1, \ldots, S_N\}$ depending on whether $S_i \in K$, or not. Additionally, simple calculations show that $\sigma^2 \geq \sum_{i=1}^{N} E[\|y_i\|^2]$, as required by the Vector Bernstein's Inequality, yields $\sigma^2 \geq \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2 \cdot g_i} - \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2}$.

Using Inequality 2 we partially satisfy Requirement 2, since we have not yet discussed how B, σ and, thus, ε can be a-priori set. In Section 3 we will choose a sampling function providing an ε that is upper bounded by a constant value known to each S_i before a monitoring phase begins. Based on this, we can fully satisfy Requirement 2.

Monitoring Scheme and Efficiency Requirement. Based on Lemma 1, sampled sites need to monitor $Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$ where (i) the estimation $\hat{v}$ of v lies, as Lemma 1(c) shows, and (ii) where the true global average v is expected to lie since $E[\hat{v}] = v$. In order to track $Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$ according to the existing GM framework, each $S_i \in K$ would need to construct local hyperspheres of the form $B(e + \frac{1}{2}\frac{\Delta v_i}{g_i}, \frac{1}{2}\|\frac{\Delta v_i}{g_i}\|)$, with the union of these local hyperspheres covering the convex hull that encompasses $\hat{v}$. However, these hyperspheres are larger (by a factor of $1/g_i$) than the ones mentioned in our algorithmic sketch. Let us now examine the reason for this important difference.

Compared to the basic GM method (Section 1.1), the above scheme omits hyperspheres of sites that do not get sampled, thus reducing the monitored area. On the other hand, since $g_i \leq 1$, the hyperspheres $B(e + \frac{1}{2}\frac{\Delta v_i}{g_i}, \frac{1}{2}\|\frac{\Delta v_i}{g_i}\|)$ have larger radii than the $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ used in the basic GM algorithm, and the centers of the spheres are also different. As an example, Figure 4(a) depicts the area that needs to be monitored, which corresponds to the balls of sites S_2 and S_3 covering the shaded part of Figure 4(a), according to Lemma 1. Hence, such a scheme may on one hand reduce FP decisions due to the fact that it uses fewer Δv_i vectors in its convex hull, but on the other hand it may also cause more FP synchronizations because it constructs larger spherical constraints centered at different positions as shown in Figure 4(a). In other words, it may perform better than GM, but this is in no way guaranteed. Obviously, this violates Requirement 1. The following lemma demonstrates how and why our sampling-based monitoring abides by Requirement 1 by using a bounded, controllable number M of sampling trials.

In a nutshell, in accordance with our algorithmic sketch, each site $S_i \in \{S_1, \ldots, S_N\}$ performs M, instead of one, independent sampling attempts using its own sampling function g_i. g_i is a fixed number < 1 for S_i during a single execution of the monitoring phase we examine here. Each such attempt produces a sample K_μ, $1 \leq \mu \leq M$ which in turn gives an instance of Estimator 1, i.e., $\hat{v}_\mu \in \{\hat{v} = \hat{v}_1, \ldots, \hat{v}_M\}$. The following lemma progressively shows that, if we choose g_is so that the expected sample size of each trial is properly upper bounded, almost surely (0.99 probability), at least one $\hat{v}_\mu \in \{\hat{v}_1, \ldots, \hat{v}_M\}$ will be included in the balls of the corresponding sample K_μ without needing to scale these balls by $1/g_i$. Please refer to Table 1 for a summary of the utilized notation.

Lemma 2. *(a) For a single sampling trial with sample of cardinality $|K|$:*
$\hat{v} \in Conv(\{e + \frac{|K|}{N \cdot g_i}\Delta v_i\} : \forall S_i \in K) \Rightarrow \hat{v} \in \bigcup_{S_i \in K} B(e + \frac{|K|}{N \cdot g_i}\frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i}\|\frac{\Delta v_i}{2}\|) \Rightarrow \exists S_i \in K : \hat{v} \in B(e + \frac{|K|}{N \cdot g_i}\frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i}\|\frac{\Delta v_i}{2}\|)$

---

[2]The inequality yields slightly higher ε for practical δ values we consider in this work, but we set ε as above for simplicity and ease of exposition.

[3]Note that $\sigma^2 \geq \sum_{i=1}^{N} E[\|y_i\|^2] \geq \sum_{i=1}^{N} E[\|y_i\|^2] - (E[\|y_i\|])^2 = \sum_{i=1}^{N} Var[\|y_i\|]$, therefore σ^2 bounds the sum of individual length variances.

(b) On expectation $\frac{|K|}{N \cdot g_i} \leq 1 + \frac{1}{N} \approx 1$, $\forall S_i \in K$.

(c) Assume each site $S_i \in \{S_1, \ldots, S_N\}$ performs $1 \leq M \leq \left\lceil \frac{log(0.01)}{log(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N})} \right\rceil$ independent (among sites and among trials) sampling trials, using its own sampling function g_i. Further assume that g_is are chosen so that in each trial the expected sample size is bounded by $\ell n(1/\delta) \sqrt{N}$, i.e., $E[|K_\mu|] = \sum_{i=1}^{N} g_i \leq \ell n(1/\delta) \sqrt{N}$, $\forall \mu \in [1, M]$. Then, with 0.99 probability, there will be at least one trial that includes a version $\hat{v}_\mu$ of Estimator 1 in the GM-spheres (i.e, not scaled by $1/g_i$) of S_is $\in K_\mu$:

$$P \left(\nexists \hat{v}_\mu \in \{\hat{v}_1, \ldots, \hat{v}_M\} : \hat{v}_\mu \in \bigcup_{S_i \in K_\mu} B(e + \frac{\Delta v_i}{2}, \|\frac{\Delta v_i}{2}\|) \right) \leq 0.01$$

Proof. In Appendix. □

Lemma 2(a) essentially states that examining $B(e + \frac{1}{2}\frac{\Delta v_i}{g_i}, \frac{1}{2}\|\frac{\Delta v_i}{g_i}\|)$ is redundant since in fact we need to examine balls of $N/|K|$ smaller radius, which is a considerable quantity for $|K| \ll N$. The issue that arises is that since each site samples itself independently, we do not know $|K|$ during the monitoring unless sampled sites communicate with each other via the coordinating source. This is obviously something we must avoid. What we know according to Lemma 2(b) is that we probably do not need the sampled sites to consult local constraints larger than $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ because, in highly distributed settings, on expectation $\frac{|K|}{N \cdot g_i} \leq 1 + \frac{1}{N} \approx 1$, $\forall S_i \in K$ and thus we expect $B(e + \frac{|K|}{N \cdot g_i}\frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i}\|\frac{\Delta v_i}{2}\|) \subseteq B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$, $\forall S_i \in K$.

The next question is what if we are not in the expected case. Note that in principle, we do not require $B(e + \frac{|K|}{N \cdot g_i}\frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i}\|\frac{\Delta v_i}{2}\|) \subseteq B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$, for all the sampled sites to achieve monitoring the position of $\hat{v}$. It suffices for the site (at least one, but there may be more) for which $\hat{v} \in B(e + \frac{|K|}{N \cdot g_i}\frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i}\|\frac{\Delta v_i}{2}\|)$ to actually have a coefficient $\frac{|K|}{N \cdot g_i} \leq 1$.

Let us examine the statement of Lemma 2(c). If each site in the network attempts M, instead of one, sampling trials using its sampling function g_i (whatever it is), then M not necessarily disjoint samples $K_1, \ldots, K_M$ will be drawn globally in the network, each providing (if we choose to probe it at the coordinator) a value of Estimator 1, i.e., $\hat{v}_\mu \in \{\hat{v} = \hat{v}_1, \ldots, \hat{v}_M\}$. If sites utilize a g_i that yields $E[|K_\mu|] \leq \ell n(1/\delta) \sqrt{N}$, $\forall \mu \in [1, M]$ (as the one we introduce in Section 3), we guarantee that the probability of having at least one $\hat{v}_\mu \in B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ of some $S_i \in K_\mu$ is at least 0.99. Equivalently, there is a 0.01 probability of failing to track at least one $\hat{v}_\mu$. The value of M is directly tunable according to the application defined δ.

We stress that these repeated sampling trials cannot increase the size of any of the local constraints inscribed by sites, but may only increase their number from $|K_1|$ (single trial) to $|K_1 \cup \cdots \cup K_M|$. We show below that $M = 2$ to 4 in the highly distributed settings we consider in this work. In other words, instead of asking sites in $K \equiv K_1$ to participate in the monitoring process, we essentially require sites in $K_1 \cup \cdots \cup K_M$ to do so. The above process does not require the sites to communicate as the coordinator can compute M based on δ and broadcast it to the sites at the beginning of the tracking process. Then sites need to independently perform M sampling attempts based on their g_i. During the monitoring phase, Requirement 1 is absolutely satisfied because:

$$\bigcup_{S_i \in K_1 \cup \cdots \cup K_M} B(e + \frac{\Delta v_i}{2}, \|\frac{\Delta v_i}{2}\|) \subseteq \bigcup_{i=1}^{N} B(e + \frac{\Delta v_i}{2}, \|\frac{\Delta v_i}{2}\|)$$

Remarkably, upon a partial synchronization, the coordinator does not need to probe $|K_1 \cup \cdots \cup K_M|$ sites that participated in the union of the M sampling trials. This is because, based on the Inequality 2 and a priori known, fixed error $\varepsilon = \varepsilon(\delta)$, all the produced estimators share the same worst case approximation quality guarantees. It thus suffices to probe the sampled sites of one, say the first, trial, i.e., set $K_1 \equiv K$, compute $\hat{v} = \hat{v}_1$ and check $B(\hat{v}, \varepsilon)$.

Figure 4(b) depicts the improvement in the construction of local balls, especially compared to the local constraints induced by Lemma 1 in Figure 4(a). The upcoming Example 2 shows that this is a tremendous improvement in a
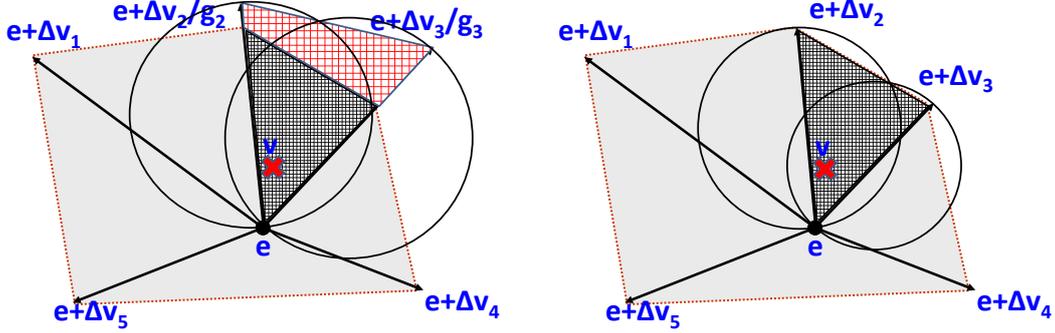
(a) Sampling-Based tracking according to Lemma 1, violating Requirement 1 ($S_2, S_3 \in K$)

(b) Requirement 1 abiding sampling-based scheme in Lemma 2 ($S_2, S_3 \in K$)

Figure 4: Sampling-based monitoring over distributed data streams. Shaded areas belong to the convex hull that needs to be monitored according to Lemma 1 and Lemma 2, respectively. The gray area corresponds to the convex hull that is formed by the entire set of sites in the network (N=5).

practical scenario of our running example. Hence, the new local constraints adhere to Requirement 1 and no additional FPs can be provided by the sampling based scheme.

A final question regards how large practical values of M should be. For practical values of M, given δ and N, interesting observations can be extracted. According to Figure 3 and Table 2, as the degree of distribution N increases a couple of sampling trials would suffice in order to track an Estimator 1 with probability way above 99% (see last column of Table 2) entailed by the lemma and we stress that this is an upper bound on M. Therefore, for the focus of this paper, i.e., highly distributed settings, in practice even a single trial $K \equiv K_1$ can be sufficient. For ease of presentation, we again return to the assumption that $M = 1$, $K \equiv K_1$ and that $\hat{v} = \hat{v}_1$ is surely included in $\bigcup_{S_i \in K} B(e + \frac{\Delta v_i}{2}, \|\frac{\Delta v_i}{2}\|)$. We stress that overall we keep our analysis consistent by (a) accounting for the effect of M in the worst case bound on our sample size and (b) include the 0.01 tracking failure probability in the false negative rates of our scheme.
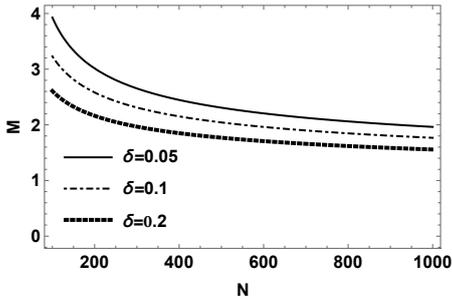
Figure 3: M versus N for various values of δ.

δ	N	$\sim M$	Prob. of failing tracking at least one $\hat{v}$
0.05	100	4	0.008
0.05	500	3	0.002
0.05	1000	2	0.009
0.1	100	4	0.003
0.1	500	2	0.01
0.1	1000	2	0.005
0.2	100	3	0.004
0.2	500	2	0.005
0.2	1000	2	0.003

Table 2: Illustration of practical values of M

Given we track $\hat{v}$, if $\hat{v} = v$ we manage to monitor the exact value of v. Even if $\hat{v} = v$ does not hold, from Inequality 2, we know that with high probability, at least $1 - \delta$, $v \in B(\hat{v}, \varepsilon)$. When $v \in B(\hat{v}, \varepsilon)$, the worst case scenario during the monitoring process occurs when $\hat{v}$ is located on the periphery of $\bigcup_{S_i \in K} B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$. Then, the fact that $\hat{v}$ lies on the boundary of some GM sphere, combined with the fact that $v \in B(\hat{v}, \varepsilon)$, guarantees that the largest distance v may travel outside the union of the spheres is ε. The first option to handle this situation is to expand the radius of the balls inscribed by sites to $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\| + \varepsilon)$ so that with probability $1 - \delta$ they include $v \in B(\hat{v}, \varepsilon)$. However, then, Requirement 1 is no longer satisfied because we are going to have fewer ($|K| \ll N$) balls only for the sampled sites, but with expanded radii. The second option is to allow such an error, which may lead to a FN decision. We opt for the second option and focus on its effect on the FN rate.
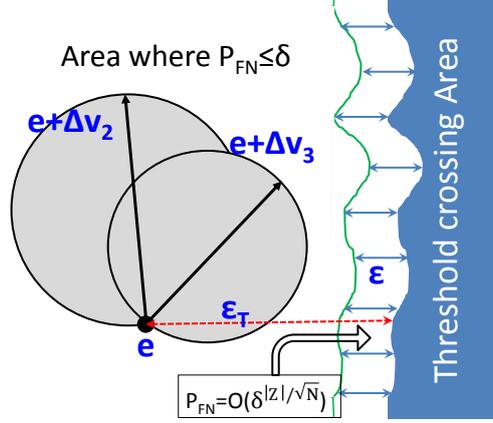
Figure 5: *P_{FN} wrt the distance from the threshold surface. In the white (left) area $P_{FN} \leq \delta$ since no ball approaches the surface more than ε.*

Example 2. *Before proceeding with our running example, we need to point out that, as clarified many times so far, our techniques are designed to operate over highly distributed settings and may not be applicable in very low network scales. For instance, the prerequisites of the Vector Bernstein inequality may not be met, if we choose a setup with few sites and set a very low δ. On the other hand, it is difficult to visualize a running example with a convex hull and hyperspheres stemming from many sites. Even the illustration of Figure 6(a), including only 5 sites, might have been cumbersome if Δv_is or T were chosen differently. Given these, our examples may be loosened representations of the basic principles we discuss since they are primarily focused on easing the illustration. To best keep up with our theoretic analysis and also provide eligible illustrations, in Figure 6(b) and forth we assume $N = 10$ and that the vectors of $S_6, S_7, S_8, S_9, S_{10}$ coincide with those of S_1, S_2, S_3, S_4, S_5, respectively, in Figure 6(a).*

Figure 6(b) depicts the union of balls monitored by our algorithm assuming $|K| = 2$ and that two of the vectors with the highest g_i, namely $\{S_1, S_2\}$, are included in K. The computation of g_i is based on Δv_is of Figure 6(a) and also on details of the sampling function discussed in Section 3. Since we have not yet shown how g_i is tuned, for now, what is important to note is that due to Lemma 2 we are able to perform the monitoring using the balls of Figure 6(b) instead of those entailed by Lemma 1(c). Figure 6(c) shows in practice this important difference comparing in the same figure the local constraints of S_1, S_2 scaled by $1/g_i$ (large balls), as entailed by Lemma 1(c), versus those of Lemma 2.

Satisfying the Tunable Accuracy Requirement. According to our algorithmic sketch and our analysis so far, under the assumption that we track $\hat{v}$, a FN decision may occur in two mutually exclusive cases:

(a) During the distributed monitoring process, upon judging potential threshold crossings of local hyperspheres that were not expanded by an ε factor (as previously described). We consider two sub-cases: In subcase (a1), the local constraint $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ of every site $S_i \in K$ has a minimum distance from the threshold surface larger than ε. Subcase (a2) covers the case when the above condition does not hold for at least one $S_i \in K$.

(b) During the synchronization process, where the coordinator probes the sample and uses $B(\hat{v}, \varepsilon)$ to determine if a full synchronization is necessary.

Note however, that these types of FNs cannot occur simultaneously since case (b) can happen only when a threshold crossing is detected during the monitoring process. We set out our discussion from case (a), which is more complicated.

Case (a1). If $v \in B(\hat{v}, \varepsilon)$, whenever every S_i's $\in K$ local constraint $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ has a minimum distance from the threshold surface larger than ε as shown in Figure 5, our choice of not expanding these spheres to $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\| + \varepsilon)$ does not affect the quality of the monitoring process. This is true because even if v lies outside the monitored area, with high probability $1 - \delta$ it has not changed sides with respect to T because $v \in B(\hat{v}, \varepsilon)$. Therefore,

when the minimum distance of the union of balls of sampled sites from the threshold surface is larger than ε (see Fig. 5), we cannot have a FN decision unless $v \notin B(\hat{v}, \varepsilon)$. The latter has a probability of δ and $P_{FN} \leq \delta$.

Case (a2). We now examine how the P_{FN} probability is bounded when there exists at least one site $S_i \in K$ with $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ placed closer to the threshold surface than ε. Looking at Figure 5, this corresponds to the zone around the threshold surface marked with an ε and corresponding double arrows. If there exists $S_i \in K$ with $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ entering the ε-zone in Figure 5, this means that v is likely to have crossed the threshold surface despite $v \in B(\hat{v}, \varepsilon)$. Let ε_T (red, dotted line in Fig. 5) denote the minimum distance of e from the threshold surface, computed once during a full synchronization process and kept until the upcoming one. Simple calculations show that if no site S_i has a $\|\Delta v_i\| > \varepsilon_T$, the global average cannot have switched side with respect to the threshold surface and no FN decision can occur. Hence, for a FN decision to occur we need *at least one* sampled site to enter the ε-zone and there should exist a number (*at least one*) of sites in the network that have drifted more that ε_T distance from e and *are not* included in K. If at least one of the threshold crossing sites is sampled then a local violation will be detected and no FN can occur at this stage. Therefore, assuming that at a given time point $|Z|$ sites cross the threshold, $P_{FN} \leq \prod_{S_i \in Z} (1 - g_i)$, since a FN will occur when none of these $|Z|$ sites is included in the sample. As we are going to show in Section 3, for a properly constructed g_i, even in case that for some sites their drift vectors enter the ε-zone, P_{FN} has an upper bound proportional to $\delta^{\frac{|Z|M}{\sqrt{N}}}$ which decreases exponentially with the number of threshold crossing sites and sampling trials. We further show that this bound on P_{FN} is pessimistic, as it is computed on the pathological case where for all $S_i \in Z$, $\|\Delta v_i\| = \varepsilon_T$. What happens in practice, because v is the average of the drift vectors, is that in order for v to cross the threshold surface, the threshold surface is crossed by either several moderate in length drift vectors (in which case $|Z|$ is large), or by fewer but larger drift vectors. In the latter case, we show in Section 3 that the sampling probability of such sites is larger, making it less likely that they will all be omitted from the sample.

Case (b). In our algorithmic sketch, during a synchronization the coordinator, trying to reduce the cost of a potential FP decision, first attempts to save communication by collecting the Δv_i vectors only of the sampled sites (of the first trial). It then computes $\hat{v}$, checks $B(\hat{v}, \varepsilon)$ and only if the latter ball crosses the threshold surface, a full synchronization takes place. An FN decision may occur only when $v \notin B(\hat{v}, \varepsilon)$ which happens with probability at most δ and thus $P_{FN} \leq \delta$.

Based on Inequality 2, we set $\varepsilon = (1 + \sqrt{\ell n(1/\delta)}) \cdot \sigma$. In the next section we provide a sampling function that upper bounds σ by a constant value and tunes ε according to the application defined δ. Having bound σ, we showed in this section that P_{FN} can be also bounded by δ. Thus, Requirement 3 is satisfied as well. In the next section we further exhibit that based on the constructed g_i, δ also successfully tunes the sample cardinality $|K|$ and, thus the anticipated savings of the sampling-based scheme in terms of FP reduction and bandwidth preservation.

3. Setting the Sampling Function

In our sampling-based scheme, each S_i individually decides whether to include itself in K (more generally in each trial $K_1, \ldots, K_M$, but we still assume $M = 1$ at this point) or not, using a sampling function g_i. Our generic technique can accommodate any g_i that samples multidimensional site vectors. However, not all functions yield the desired properties for our scheme. We next construct, in a step-by-step fashion, a suitable g_i and reason about our choices based on the properties that each element attributes to our scheme. We eventually derive a proper g_i that simultaneously (a) ensures a sample of $O(\ell n(1/\delta) \sqrt{N})$ size, (b) upper bounds σ and, thus, ε by an a priori (before acquiring the sample) constant value controlled by δ. In that, g_i allows the sampling-based scheme to comply with Requirement 2, (c) given the previous upper bound that determines the size of the ε−zone (Fig. 5), g_i tunes the probability of FNs (Requirement 3).

• $\|\Delta v_i\|$ **should be included in the numerator of** g_i**.** According to our algorithmic sketch in Section 2.2, upon a local violation the coordinator probes only sites $S_i \in K$ (of the first trial) and checks $B(\hat{v}, \varepsilon)$ for threshold crossing in order to call for a full synchronization, or not. To inscribe $B(\hat{v}, \varepsilon)$, the radius ε should be (bounded by) a constant value. In order to come up with a constant value for ε, according to Inequality 2, we need to bound σ. In Section 2.2, we showed that $\sigma^2 \geq \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2 \cdot g_i} - \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2}$. Apart from g_i, the only variable term included in the latter inequality

is the size of the drift $\|\Delta v_i\|$. To eliminate this variable term, $\|\Delta v_i\|$ should be included in the numerator of g_i (see Inequality 3).

• $\ell n(1/\delta)$ **needs to be included in the numerator of** g_i**.** As mentioned in Requirement 3, ε should be tunable by (a function of) δ, i.e., $\varepsilon = \varepsilon(\delta)$, so as to allow the size of the $\varepsilon-$ zone to be controlled by the application. Hence, $\ell n(1/\delta)$ needs to be included in the numerator of g_i. Due to the presence of $(1 + \sqrt{\ell n(1/\delta)}) \cdot \sigma \ (= \varepsilon)$ in Inequality 2, we will later show in Equation 4 that placing $\ell n(1/\delta)$ in the numerator of the function allows the application to express the size of the $\varepsilon-$zone as a *fraction* (< 1) of the bound of the approximation error between $\hat{v}$ and v.

• **A term** N^x**,** $x > 0$ **is needed in the denominator of** g_i**.** To ensure communication savings and reduced monitored area, we need $|K| \ll N$. The expected sample cardinality of our scheme is given by $\sum_{i=1}^{N} g_i$ (per trial). Since this sum iterates over all the N terms, to ensure $|K| \ll N$, we need a term N^x in the denominator of g_i in order to obtain an expected communication cost of $O(N^{1-x})$. What is then required is to compute a proper value for $x > 0$.

• **A constant** U **such that** $U > h \cdot \|\Delta v_i\|$**,** $h > 1$ **is necessary in the denominator of** g_i**.** Having required that $\|\Delta v_i\|$ lies in the numerator of g_i, to achieve $O(N^{1-x})$ cardinality, the presence of a constant U such that $U > h \cdot \|\Delta v_i\|$ for some $h > 1$ in the denominator of g_i is necessary as well.

Guidance for setting U: In a setup where sites receive ± 1 updates per dimension [32, 33, 34] over a sliding window of w size, the maximum $\|\Delta v_i\|$ that may occur is proportional to $\sqrt{d} \cdot w$. In case of unbounded inputs, a generalization of the bound used in [33] would suffice. In particular, [33] focuses on linear functions and assumes that an estimation of the global count (in one dimension) is available beforehand. It thus sets U equal to that total absolute count estimation. The equivalent in our multidimensional scenario is to utilize the values in the dimensions of e, i.e., the last known global average estimation, and express U as a function of its L_1 norm. Finally, another plausible option is to set U according to the minimum distance of e from the threshold surface, which may however require to first compute an optimal reference vector (see [21]) e^* instead of e.

All our previous remarks are satisfied upon setting:

$$g_i = \frac{\ell n(1/\delta) \cdot \|\Delta v_i\|}{U \cdot N^x}$$

The expected communication cost is a tunable (using δ) fraction of N, proportional only to $\sum_{i=1}^{N} g_i \leq \ell n(1/\delta) \cdot N^{1-x}$ (per trial). We then seek for a proper value for $x > 0$. Recalling the Vector Bernstein's Inequality (Inequality 2) and using the above g_i, for σ we obtain (detailed computations in [35]):

$$\sum_{i=1}^{N} E[\|y_i\|^2] = \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2 \cdot \frac{\|\Delta v_i\| \cdot \ell n(1/\delta)}{U \cdot N^x}} - \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2} \quad \Leftrightarrow$$

$$\sum_{i=1}^{N} E[\|y_i\|^2] \leq \left(\frac{U \cdot N^x}{2 \cdot \ell n(1/\delta) \sqrt{N}} \right)^2 = \sigma^2 \tag{3}$$

In order to express ε as a fraction of U and as a function of δ, while at the same time avoiding an undesirable dependence on the network scale N, we pick $x = 1/2$. Then, $\varepsilon = (1 + \sqrt{\ell n(1/\delta)}) \cdot \sigma = (\frac{1 + \sqrt{\ell n(1/\delta)}}{2 \cdot \ell n(1/\delta)}) \cdot U$, while B can be set to $B = \frac{\|\Delta v_i\|}{N \cdot g_i} = \frac{U}{\ell n(1/\delta) \cdot \sqrt{N}}$. Notice that the choice of $\ell n(1/\delta)$ in the numerator of g_i is the lowest value that we could use in order to obtain a tunable by δ increase in the expected communication cost, while at the same time being able to express ε as a percentage of U. The latter claim is true due to the fact that $(\frac{1 + \sqrt{\ell n(1/\delta)}}{2 \cdot \ell n(1/\delta)}) < 1$, $\forall \delta < e^{-1}$ (i.e., a range that contains the typical values for δ).

The Sampling Function. Before summarizing the properties of our sampling-based GM scheme according to the proposed g_i, we first need to account for the M factor introduced in Lemma 2, but left out of our subsequent discussion for ease of exposition. As noted in Lemma 2, $1 \leq M \leq \left\lceil \frac{\log(0.01)}{\log(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N})} \right\rceil$. Hence, the overall worst case bound on the expected number of sites participating in the monitoring process is $M \cdot E[|K|] \leq M \ell n(1/\delta) \sqrt{N}$. The M factor is

dominated by $\ell n(1/\delta) \cdot \sqrt{N}$ in highly distributed settings as shown in Figure 3 and Table 2. Furthermore, note that the ε value computed above is not affected by M, but is a constant, for fixed δ, value for any given sample. Hence, even if we draw M samples, each of them will have the same bound for the estimator error ε and the $\varepsilon-$ zone attached to the threshold surface (see Figure 5) will be identical either for 1 or M samples. Therefore, pointing out the characteristics of our sampling-based GM scheme, for $\delta \in (0, e^{-1})$ we have:

$$\begin{cases} g_i = \frac{\|\Delta v_i\| \ell n(1/\delta)}{U \cdot \sqrt{N}} & \text{Sampling Function} \\ \varepsilon = (\frac{1+\sqrt{\ell n(1/\delta)}}{2 \cdot \ell n(1/\delta)}) \cdot U & \hat{v} \text{ Estimation Error} \\ O(\ell n(1/\delta) \sqrt{N}) & \text{Expected Sample Size} \end{cases} \qquad (4)$$

Example 3. *Recalling our running example, the maximum $\|\Delta v_i\|$ an update cycle may yield at S_i is $\sqrt{d} = \sqrt{3} \approx 1.73$. Thus after up to 100 updates globally in the network, i.e., 10 update cycles with $N = 10$, $U = 17.3$ and for $\delta = 0.05$, $\varepsilon = 7.89$. Figure 6(d) illustrates the "buffer" this $\varepsilon-$zone adds on the initial threshold surface of the running example.*

The following table computes, for $N = 100$ and $N = 961$, the values of ε, the range of g_i values in this example, and $\ell n(1/\delta) \sqrt{N}$ (an upper bound on $\sum_{i=1}^{N} g_i$) for δ values of 0.1 and 0.05. We note that the ratio of this upper bound over N becomes smaller as N increases. Moreover, note that smaller δ values result in smaller ε and larger g_i values, as smaller δ values point to a requirement for fewer FNs.

δ	N	$\sqrt{N}$	*Range of g_i*	ε	$\ell n(1/\delta) \sqrt{N}$
0.1	*100*	*10*	*[0,0.23]*	*9.5*	*24*
0.1	*961*	*31*	*[0,0.074]*	*9.5*	*72*
0.05	*100*	*10*	*[0,0.3]*	*7.89*	*30*
0.05	*961*	*31*	*[0,0.097]*	*7.89*	*93*

$\square$

Note that the inequalities $g_i < 1$ and $\varepsilon \leq \sigma^2/B$ hold for sufficiently high N, as required by the Vector Bernstein's Inequality. We further point out that in Equations 4, ε is directly controllable using the parameter δ. Furthermore, observe that when δ decreases, then ε also decreases, while the expected sample size increases logarithmically. This is a trade-off between bandwidth consumption and accuracy that our sampling-based scheme achieves by a single, application defined parameter δ.

Recall that because $\Delta v_i(t) = v_i(t) - v_i(t_s)$, $\|\Delta v_i\|$ expresses the deviation magnitude of S_i's local vector since the last synchronization. Hence, even if a site has no update in the current monitoring phase, its sampling probability can still be high if $\|\Delta v_i\|$ has built up from past, after the last synchronization, updates. The reason is quite intuitive, since a site with a high $\|\Delta v_i\|$ has at any time the potential to severely affect the placement of actual global average (Section 1.1) and its estimation's (also analyzed below Estimator 1).

Notice that the proposed g_i does not explicitly impose a lower bound on sample size. However, even if no site gets sampled (such a case becomes less likely as the number of sites increases), our algorithm will estimate - according to Estimator 1 - that $\hat{v} = e$. In any such case, according to our analysis using the Vector Bernstein Inequality, our estimation is accurate within ε from the true average with (controllably) high probability. On the other hand, our framework needs to ensure that it samples enough sites when the global vector v does cross the threshold surface, to avoid FNs. In Lemma 3 below, we bound P_{FN} based on both the number of threshold crossing sites and the distance of the spheres from the $\varepsilon-$ zone and, thus, the threshold surface.

Completing the puzzle for P_{FN} bounds. The following Lemma 3 elaborates on the P_{FN} bounds yielded by the chosen g_i. We point out that the lemma expresses the overall, final P_{FN} and does not entail the computation of a union bound over all sampled sites. The condition $\forall S_i \in K_\mu, B(e + \frac{\Delta v_i}{2}, \frac{\|\Delta v_i\|}{2}) \bigcap \varepsilon - zone = \emptyset$ simply says that the union of hyperspheres for the sampling trial that tracks $\hat{v}_\mu$ (at least one such trial exists with 99% probability - Lemma 2(c)) constructed by sites does not intersect the $\varepsilon-$zone. Since this holds if and only if $\forall S_i \in K_\mu$ no local ball enters the $\varepsilon-$zone, therefore the condition.

Lemma 3. *The Sampling-Based GM Scheme being set according to Equation 4, also accounting for Lemma 2(c), yields:*

- $P_{FN} \leq \delta + 0.01 \quad$ *if M trials* $\bigwedge \forall S_i \in K_\mu, B(e + \frac{\Delta v_i}{2}, \frac{\|\Delta v_i\|}{2}) \bigcap \varepsilon - zone = \emptyset$
- $P_{FN} = O(\delta^{\frac{|Z|M}{\sqrt{N}}}) \qquad\qquad\qquad\qquad$ *otherwise*

where $\mu \in [1, M]$ *any one of the sampling trials for which* $\hat{v}_\mu \in \bigcup_{S_i \in K_\mu} B(e + \frac{\Delta v_i}{2}, \|\frac{\Delta v_i}{2}\|)$, *Z (more precisely Z(t)) denotes the set of threshold crossing sites. Thus, one can properly tune* δ *to obtain the desired FN probability.*

In Figure 5 and the corresponding analysis of Section 2.2 we reasoned about the first part of Lemma 3 in Case(a1) and Case(b), i.e., when no local constraint enters the ε−zone. To this, we add the 0.01 tracking failure probability entailed by Lemma 2(c). In other words, a FN is produced if either we fail to track a version of Estimator 1 - 0.01 probability - or given we track at least one version of Estimator 1 and being outside the ε−zone, we miss the true position of v, due to $v \notin B(\hat{v}(t), \varepsilon)$, with δ probability.

Regarding the second part of the lemma, in Section 2.2 - Case(a2) and Figure 5, we mentioned that for one sampling trial per site, when some of the sampled sites enter the ε-zone, $P_{FN} \leq \prod_{S_i \in Z}(1 - g_i)$, where Z the set of threshold crossing sites globally in the network. Due to the value of the chosen g_i we obtain $P_{FN} \leq \prod_{S_i \in Z}(1 - \frac{\|\Delta v_i\| \ell n(1/\delta)}{U \cdot \sqrt{N}})$. However, we also noted that for a site $S_i \in Z$, $\|\Delta v_i\| > \varepsilon_T$, since otherwise S_i cannot have crossed the threshold surface. Therefore, since $g_i < 1$, substituting above for a single sampling trial we get:

$$P_{FN} \leq (1 - \frac{\ell n(1/\delta) \cdot \varepsilon_T}{U \cdot \sqrt{N}})^{|Z|} \leq e^{-\frac{|Z| \cdot \ell n(1/\delta) \cdot \varepsilon_T}{U \cdot \sqrt{N}}} = \delta^{\frac{|Z| \cdot \varepsilon_T}{U \cdot \sqrt{N}}}$$

Then, considering the fact that each site performs M sampling trials provides the outcome of Lemma 3 when some of the sampled sites enter the ε−zone.

Note that according to [21] a common reference point (instead of e) can be chosen so that ε_T is maximized. All these optimizations are orthogonal to our techniques and also enable us to treat ε_T as a sufficiently large, a priori known constant in our analysis. As mentioned in Section 2.2, this bound on P_{FN} is a worst-case bound that is derived from a pathological case, in which for all $S_i \in Z$, $\|\Delta v_i\| = \varepsilon_T$. However, what happens in practice, because v is the average of the drift vectors, is that in order for v to cross the threshold surface, the threshold surface is crossed (i) by either several moderate in length drift vectors, in which case $|Z|$ is large and P_{FN} is small, since it decreases exponentially with the number of threshold crossing sites, or (ii) by fewer but larger drift vectors that, thus, have larger sampling probabilities. In the latter case, it is less likely that they are all omitted from the sample.

Thus, apart from ensuring $P_{FN} \lesssim \delta$ when no local constraint enters the ε-zone as discussed in Section 2.2, we also bounded the complementary case, and note that P_{FN} may become even lower than δ when $|Z|$ is sufficiently larger than $\sqrt{N}$. We emphasize that the minimum distance of e from the threshold surface, i.e., ε_T (see Figure 5), is computed during a synchronization and is, thus, a known parameter until the next central data collection. In any case, the size of the ε−zone can be tuned to the desirable extent using δ as discussed above.

Time Evolving P_{FN} bounds. According to Lemma 3, at time t, a FN occurs with probability at most P_{FN}. Let t_{now} denote the current execution of the monitoring phase and t_{next} the upcoming one. Upon a FN occurrence at t_{now} and until t_{next} the following mutually exclusive events may occur: (E1) the global vector $v(t_{now})$ switches side with respect to the threshold and $v(t_{next})$ remains at the same side, in which case the lifetime of the threshold crossing spans two monitoring phases and (E2) $v(t_{now})$ crosses the threshold, while $v(t_{next})$ jumps back. In (E2) the threshold crossing was lost with the prescribed P_{FN} probability according to our previous analysis. However, in (E1) the probability of missing the event of the threshold crossing in consecutive monitoring phases is decreased.

To understand why, observe that if we have two as in (E2) or more general n monitoring phases in which the threshold crossing persists, the probability of failing to pinpoint **at least once** is bounded by **at most** $(P_{FN})^n$. Notice that S_is sample themselves independently of whether they were included in the previous sample. According to our algorithmic sketch in Section 2.2, the sample (of all trials) is emptied in every execution of the monitoring and (if necessary) the synchronization phase and based on Equation 4 sites utilize a $g_i(t)$ that is also oblivious to previous samples.

(a)

$N = 5, T = \ell og(5) + 0.01, e = [8,8,8],$
$e + \Delta v_1 = [8,9,14], e + \Delta v_2 = [8,13,8], e + \Delta v_3 = [8,10,10], e + \Delta v_4 = [8,6,5], e + \Delta v_5 = [5,8,5]$

(b)

$N = 10, |K| = 2, T = \ell og(10) + 0.01, g_1 = g_6 \approx 0.17,$
$g_2 = g_7 \approx 0.14, g3 = g_8 \approx 0.08, g_4 = g_9 \approx 0.1,$
$g_5 = g_{10} \approx 0.12.$ In all g_is (Sec. 3), $\delta = 0.05, U \approx 12.$

(c)

Effect of scaling the sampled balls by g_i as entailed by Lemma 1(c) (large) vs Lemma 2 (smaller balls). Functions g_1, g_2 of K are those used in Figure 6(b).

(d)

ε -zone after 10 update cycles (up to 100 updates in the network), $\delta = 0.05$ $U = \sqrt{3} \cdot 10 \approx 17.3,$
$\varepsilon \approx 7.89, T = \ell og(10) + 0.01.$

(e)

GM vs CV from Fig 6(a). CV directly tracks the convex hull because C (not shown) is assumed convex. GM tracks larger area using balls.

(f)

Safe Zone C defined by the (infinite) plane through $[0,0,14], [15,11,0], [0,7,7]$. Areas of negative positive & zero signed distance are marked.

(g)

Another safe zone defined by a ball. This safe zone, almost touches the threshold surface, but is also bounded away from it.

(h)

ε_C -zone after 10 update cycles (up to 100 updates in the network), $\delta = 0.05, U = \sqrt{3} \cdot 10 \approx 17.3,$
$\varepsilon_C \approx 7, T = \ell og(10) + 0.01.$

(i)

Comparing sampling-based monitoring schemes.
$N = 10, \delta = 0.05, U = 17.3, e + \Delta v_4 = [8,3,3],$
$d_C(e + \Delta v_4) \approx 4.5, g_4 = 0.39 > g_4^C = 0.25.$

Figure 6: Illustrations of our running example

Therefore, Lemma 3 provides a worst case P_{FN} bound not only for a single instance of the monitoring or synchronization phases, but also for threshold crossings that span multiple such phases. In Section 6.4 we experimentally validate the fact that even if FNs do occur, they are immediately pinpointed in the near future.
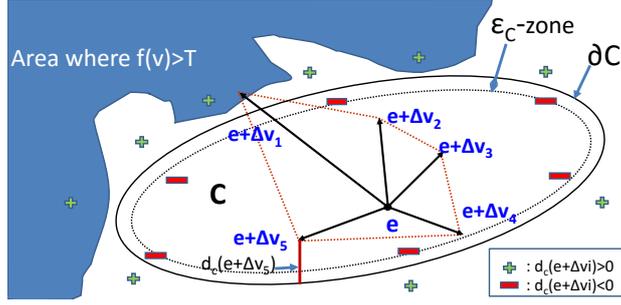
Figure 7: Illustration of the CV concept. Had all $e + \Delta v_i$ vectors been inside C, no synchronization would be required. Since $e + \Delta v_1$ exits C, S_1 informs the coordinator. The basic elements for signed distance calculation are also included in the figure, while respective Euclidean distance signs are marked in proper areas of the input domain.

4. Extending and Revisiting Sampling-Based Monitoring in the Convex Safe Zone Context

The GM method on which we built our sampling-based techniques so far "is generic, i.e., it can be applied to any function defined over the average of the local vectors at the sites" [14]. Nonetheless, the latest advancements regarding geometry-based monitoring [14, 27] show that for certain function classes, such as L_p norm related queries and join aggregates, the communication performance of the tracking process can be drastically improved by utilizing a clever convexization idea. The techniques we discuss here revisit and optimize our generic sampling-based scheme (Section 2.2 and Section 3) to exploit the characteristics of the latter, novel setting without compromising its general applicability and functionality.

Convex Safe Zone (CV) Approach. The basic principle behind the convex safe zone (CV) approach [14, 27] is to compute in hand a sufficiently large convex subset of the input domain, say C, where the global average can move without crossing the threshold surface. In other words, C is an, ideally maximal, convex subset of the input domain that does not intersect the threshold surface. The scheme can avoid tracking a superset, that is the union of local balls, but instead distributively monitor the exact convex hull. To achieve that, sites need to simply check if $e + \Delta v_i \in C$ and call for a synchronization otherwise. This is because, by convexity of C, the convex hull cannot exit C and invade the threshold surface if all of its vertices are included in it. In the latter case, the global average cannot have changed sides with respect to the threshold surface. Overall, the CV concept can simplify local tests and it can reduce false positives by monitoring the exact convex hull. Figure 7 schematically illustrates the CV concepts, further analyzed in the upcoming example.

Example 4. *To better conceive the advantage of the CV approach consider our running example and in particular Figure 6(e). Assume that a convex safe zone C, approaching T sufficiently well, exists. C is not shown in Figure 6(e) but possible choices of C for our running example are discussed in Example 5 (Figure 6(f) and Figure 6(g)). For now, what is important to notice, is that due to the existence of C, each site simply checks whether its local vector $e + \Delta v_i$ lies in C. Globally this is interpreted to monitoring the exact convex hull, represented by the black area in Figure 6(e). On the contrary, the generic GM approach monitors the gray area representing the union of spheres in Figure 6(e). As shown in this figure, provided a good C has been computed in hand, using CV results in tracking a drastically reduced part of the input domain.*

Unfortunately, even when a large C can be computed for the monitored function, the safe zone idea still faces scalability issues since, as we discussed in Section 1.2, in highly distributed settings the monitored convex hull by itself is large, apart from the fact that the monitored spheres cover a superset of it. This fact is also confirmed in our experimental evaluation (Section 6.6) upon applying the CV approach on high network scales.

To solve these scalability issues the techniques we presented so far and our proposed sampling function are still applicable. The only difference is that now, the local constraint checked by each sampled site is whether $e + \Delta v_i \in C$ or not, instead of assessing intersection of local balls with the threshold surface. However, in the current work we advance far beyond simply applying our generic sampling-based scheme on top of the CV method.

Extending and Revising the Generic Scheme. More precisely, we first (Section 4.1) introduce a novel mapping of the CV monitoring task, from a multidimensional input domain of arbitrary dimension to an equivalent task on

a unidimensional space. Our mapping is exact, i.e., lossless regarding true threshold crossings and it does not introduce additional false positive centralization decisions compared to CV. Beyond its exciting theoretic interest, our mapping can practically reduce the negative communication impact of a FP synchronization proportionally to the dimensionality d of the monitored space by having each site transmit a single value instead of a vector. Unfortunately, this mapping still lacks desired scalable characteristics. This is because it does not prevent FP centralizations during which the required number of communicated messages is still $O(N)$, but can reduce the amount of communicated data of such a FP decision. What is more important is that our mapping opens new opportunities for our sampling-based GM concept. As a matter of fact, we revisit our sampling-based GM scheme together with the robust sampling function we propose, rendering them applicable to the unidimensional setting. The revised $(1-d)$ version of our sampling-based scheme possesses additional advantages as it reduces the value of the approximation error of the monitored $1-d$ quantity. Based on this reduced error argument, our monitoring scheme, revised for the CV concept, both in expectation and also in practice (Section 6.6) manages to reduce the number of false decisions throughout the tracking process. In addition, because it takes advantage of the introduced unidimensional mapping, it can also reduce the amount of communicated data during each FP decision, proportionally to the dimensionality d.

4.1. A Novel Exact Mapping from Arbitrary Dimensionality to Unidimensional Tracking

Here, we temporarily leave aside any sampling effort so as to purely study our mapping and its monitoring properties.

The mapping. Let $C \subset \mathbb{R}^d$ be a convex subset of the admissible region, i.e., where v can move without causing the monitored function to cross the threshold surface. The signed distance of $e + \Delta v_i \in \mathbb{R}^d$ from C is defined as:

$$d_C(e + \Delta v_i) = \begin{cases} -dist(e + \Delta v_i, \partial C), & e + \Delta v_i \in C \\ 0, & e + \Delta v_i \in \partial C \\ dist(e + \Delta v_i, \partial C), & e + \Delta v_i \in \overline{C} \end{cases}$$

where ∂C denotes the boundary of C, $\overline{C} \equiv R^d \backslash C$ and $dist(e + \Delta v_i, \partial C) = \inf_{c_i \in \partial C} \|e + \Delta v_i - c_i\|$.

Figure 7 provides an example of the above definition for given C and $e + \Delta v_i$ vectors where the sign of the distance is marked in respective areas of the input domain. The following novel lemma essentially acts as the tool for mapping the $d-$ dimensional monitoring task of the CV concept to a unidimensional tracking setup. To formulate and prove the lemma we temporarily simplify the notation for ease of exposition; p_i below corresponds to $e + \Delta v_i$, while l_i equals $d_C(e + \Delta v_i)$.

Lemma 4. *Let C be a closed convex set in a normed linear space, and assume that (subscript order is used for simplicity) $p_1, \ldots, p_k$ are inside C, and $p_{k+1}, \ldots, p_N$ are outside C. Denote the signed distance of p_i from the boundary of C by l_i (negative if $p_i \in C$, positive otherwise). Assume that $\sum_{i=1}^{N} l_i \leq 0$. Then the average of $p_1, \ldots, p_N$ is inside C.*

Proof. Denote by c_i the point on the boundary of C which is closest (possibly not unique if $p_i \in C$) to p_i. Denote $p_i = c_i + u_i$, so $|l_i| = \|u_i\|$.

Since a convex set is closed under averaging, it suffices to show that $p_1 + \ldots + p_N$ is equal to the sum of N points all of which are in C. To achieve this, note first that

$$p_1 + \ldots + p_N = p_1 + \ldots + p_k + (c_{k+1} + u_{k+1}) + \ldots (c_N + u_N) =$$
$$p_1 + \ldots + p_k + c_{k+1} + \ldots + c_N + (u_{k+1} + \ldots u_N)$$

Now, denote $u = u_{k+1} + \ldots + u_N$. From the triangle inequality, $\|u\| \leq \|u_{k+1}\| + \ldots + \|u_N\| = l_{k+1} + \ldots + l_N \leq |l_1| + \ldots + |l_k|$, because $\sum_{i=1}^{N} l_i \leq 0$. So we have

$$p_1 + \ldots + p_N = \left(p_1 + \frac{|l_1| \cdot u}{|l_1| + \ldots + |l_k|} \right) + \ldots + \left(p_k + \frac{|l_k| \cdot u}{|l_1| + \ldots + |l_k|} \right) + c_{k+1} + \ldots + c_N$$

19

To complete the proof, note that for $1 \le i \le k$, it holds that $p_i + \frac{|l_i| \cdot u}{|l_1| + \ldots + |l_k|} \in C$. This is because

$$\|\frac{|l_i| \cdot u}{|l_1| + \ldots + |l_k|}\| = \frac{|l_i|}{|l_1| + \ldots + |l_k|} \|u\| \le \frac{|l_i|}{|l_1| + \ldots + |l_k|} (|l_1| + \ldots + |l_k|) = |l_i|$$

but since for $i = 1 \ldots k$ the distance of p_i from the boundary is $|l_i|$, it is clear that $p_i + \frac{|l_i| \cdot u}{|l_1| + \ldots + |l_k|} \in C$. $\qquad\square$

Plainly put, the lemma states that if the sum of distances from the boundary of the points $e + \Delta v_i \in C$ is larger than the sum of distances from the boundary of the points outside C, in which case the overall sum of signed distances will be negative, then the average of all points is inside C. The following corollary, which we henceforth use in our analysis, is a straightforward adaptation of Lemma 4 for the case of the average of signed distances.

Corollary 1. *Given $C \subset \mathbb{R}^d$ defined as above, if the average of signed distances $d_C(e + \Delta v_i)$ is negative, $\forall S_i \in \{S_1, \ldots, S_N\}$, then the global average v is inside C i.e.,*

$$D_C = \frac{\sum_{i=1}^{N} d_C(e + \Delta v_i)}{N} < 0 \Rightarrow v \in C$$

Monitoring Properties. Obviously, the condition $D_C < 0$ of Corollary 1 surely holds when $d_C(e + \Delta v_i) < 0$, $\forall S_i \in \{S_1, \ldots, S_N\}$. This already forms the condition for executing the distributed monitoring process in the unidimensional setup induced by Lemma 4. In other words, every site in the network checks whether its signed distance from C is negative and calls for a synchronization otherwise. This is lossless with respect to possible threshold crossings due to Corollary 1. Unequivocally, a positive signed distance $d_C(e + \Delta v_i)$ at a site S_i simultaneously means that $e + \Delta v_i \notin C$ in which case the CV approach [14, 27] also contacts the coordinating source. Therefore, our mapping does not introduce additional false positives. The latter two properties combined imply that our mapping is exact i.e., no false negatives can be caused and no additional false positives are introduced. On the other hand, note that $D_C > 0$ can hold without the global average necessarily exiting C. Thus, there is still a possibility that a FP is assessed by the coordinator after having $D_C > 0$ and centralizing the local vectors of the sites in order to compute a new e, as required by the geometric scheme.

Overall, using our novel unidimensional mapping technique, we can limit the effect of a FP by centralizing a single value from each site instead of $d-$dimensional vectors. This can reduce the amount of communicated data by a factor of d in the case of a detected FP, but in terms of communicated messages the $O(N)$ cost of the initial CV approach remains, while the number of FPs is identical as well. Below, we first discuss a couple of options for setting C and the corresponding local tests that should be performed using our running example. We next discuss how we can revise our sampling-based geometric monitoring scheme (Section 2.2) for the unidimensional setting so as to reduce (apart from the size of the communicated data) the number of false decisions throughout the tracking process without increasing the worst case bound on the expected sample size.

Example 5. *By the design of the CV approach, the safe zone C is to be given as input before the monitoring task begins (but can be adjusted from one synchronization to another). There are more than one options for setting C depending on the monitored function. Optimizing this choice for certain functions has been studied in related work [16, 27, 14] and is out of our scope. We here show exemplary safe zones that are applicable in our running example, discussing the effect of the CV approach and of our new Lemma 4.*

In Figure 6(g) we depict one such possible choice of C defined by a (infinite) plane (see caption for details). The area above the drawn plane corresponds to 3-dimensional vectors possessing $d_C(e + \Delta v_i) < 0$, while the area below the plane (including the threshold surface) corresponds to $d_C(e + \Delta v_i) > 0$ as marked on the figure. Thus, any S_i of our example needs to simply check if $d_C(e + \Delta v_i) > 0$, i.e., its $e + \Delta v_i$ vector penetrates the plane shown in the figure towards the true threshold surface.

In Figure 6(g) we illustrate a different choice for C which corresponds to a hyperball (chosen so that it fits in the figure). In this case $d_C(e + \Delta v_i) < 0$ for all vectors that are inside this ball, while $d_C(e + \Delta v_i) > 0$ outside. That is, if a $e + \Delta v_i$ vector of some site is included in the ball describing C, then $d_C(e + \Delta v_i) < 0$ and $d_C(e + \Delta v_i) > 0$ otherwise. As shown in Figure 6(g) this spherical safe zone better approaches the threshold surface compared to the one in Figure 6(g), but is upper bounded as well. Safe zones in the form of hyperballs are more easy to compute and adjust between synchronizations (see Section 5 in [21]). Therefore, we also employ such safe zones in Section 6.6. $\qquad\square$

4.2. Sampling-based Monitoring Revisited

Algorithmic Sketch. As we did in our generic sampling-based scheme in Section 2.2, in the revised version we first provide an algorithmic sketch of the monitoring process and then proceed with the theoretic background justifying our choices. Throughout our description in this algorithmic sketch we restore the temporal reference t for completeness and to keep the connection with the corresponding sketch of Section 2.2.

- *Initialization Phase:* Compared to the corresponding phase in Section 2.2, the difference here is that a sufficiently large subset $C \subset R^d$ of the admissible region (the area of the input domain where the monitored function does not cross the threshold) has been computed offline and broadcasted to all sites. Still at the end of this phase, $e(t)$ constitutes the last known average data vector and $t_s, v_i(t_s)$ have been appropriately set across the network.

- *Monitoring Phase:* During the monitoring phase, each S_i receives updates of its local vector $v_i(t)$ and computes $\Delta v_i(t) = v_i(t) - v_i(t_s)$. At the beginning of each monitoring phase the sample of sites participating in the tracking process is empty, i.e., $K(t) = \emptyset$ (for all M trials mentioned below). Again, to determine if $S_i \in K(t)$, each site independently flips a biased coin, but this time with adjusted (discussed shortly) success probability of $g_i^C(t)$. $g_i^C(t)$ is a revised sampling function tailored for our new scheme, still independently computed by each site. Given C, $e(t)$ and $\Delta v_i(t)$, each $S_i \in K(t)$ checks the local condition which this time is $d_C(e(t) + \Delta v_i(t)) < 0$. If $d_C(e(t) + \Delta v_i(t)) < 0, \forall S_i \in K(t)$ the monitoring process continues. In case at least one $S_i \in K(t)$ detects $d_C(e(t) + \Delta v_i(t)) \geq 0$, it calls for a synchronization. This process may be repeated $1 \leq M \leq \left\lceil \frac{\log(0.01)}{\log\left(e^{-0.042\sqrt{\ln(1/\delta)N}}\right)} \right\rceil$ times (independent trials), independently at each site $S_i \in \{S_1, \ldots, S_N\}$.

- *Synchronization (Partial or Full) Phase:*

    - *Partial* Synchronization: During the initial attempt for a partial synchronization, the coordinator requires only the sites sampled in one, say the first, trial to contribute their $d_C(e(t) + \Delta v_i(t))$ which are now single values instead of vectors. Using these signed distance values, it derives an unbiased estimate $\hat{D_C}(t)$ (using Estimator 5, introduced shortly) of $D_C(t)$ and checks whether $\hat{D_C}(t) + \varepsilon_C > 0$. If this does not hold, then the coordinator deduces that this was a FP alarm with high probability $1 - \delta$ and the tracking continues unaffected. Otherwise, a full synchronization takes place.

    - *Full* Synchronization: This time our scheme does not directly probe also the whole data vectors of sites that did not participate in the first trial, but instead performs an additional preliminary check. Initially, the rest of the sites report their $d_C(e(t) + \Delta v_i(t))$ so that the exact $D_C(t)$ is computed. If $D_C(t) < 0$ the coordinator assesses this was **surely** (Corollary 1) a FP decision using a sole signed distance value per site. Thus, it still avoids having sites communicating their d-dimensional vectors and the tracking process continuous unaffected. If the FP test conducted by the coordinating source fails again, a full synchronization takes place, as all indicators are that a true threshold crossing took place. The coordinator thus computes a new $e(t) = e(t_s)$ vector which is broadcasted to the underlying sites. Upon the receipt of the new $e(t)$, each site updates $t_s, v_i(t_s)$.

Monitored Estimator and Sampling Function. Let us assume that each S_i performs a single sampling trial, first because the coordinator asks only for the vectors sampled in the first trial during a partial synchronization and second because the utility of more trials will be introduced upon commenting on the tunable guarantees of this new sampling scheme.

Due to Lemma 4 and Corollary 1, this time our focus is not on ensuring that we monitor an accurate estimation of the global average. Instead, we are interested in obtaining and monitoring an accurate estimation $\hat{D_C}$ of D_C. In that, we must make sure that with high (tunable) probability $1 - \delta$ (details on that follow) the coordinator is going to be informed when D_C is positive and, thus, the global average may not be in C anymore.

Let a random variable $d'_C(e + \Delta v_i) = \frac{d_C(e + \Delta v_i)}{g_i^C}$ if $S_i \in K$ with probability g_i^C, and zero otherwise. Our (Horvitz Thompson again) signed distance estimator takes the following form:

$$\hat{D_C} = \sum_{i=1}^{N} \frac{d'_C(e + \Delta v_i)}{N} = \sum_{S_i \in K} \frac{d_C(e + \Delta v_i)}{N \cdot g_i^C} \tag{5}$$

21

Corollary 2. *Estimator 5 is an unbiased estimator of D_C, i.e., $E[\hat{D}_C] = D_C$.*

Proof. Special $1-d$ case of Lemma 1(a) by replacing $\Delta' v_i$ with $d'_C(e + \Delta v_i)$. $\qquad\square$

Additionally, to see the correspondence with Lemma 2, observe that for any given sample K it holds that:

$$\hat{D}_C = \sum_{S_i \in K} \frac{d_C(e + \Delta v_i)}{N \cdot g_i^C} = \sum_{S_i \in K} \frac{\frac{|K|}{N \cdot g_i^C} d_C(e + \Delta v_i)}{|K|} \Rightarrow \exists S_i \in K : \begin{cases} \hat{D}_C \leq \frac{|K|}{N \cdot g_i^C} d_C(e + \Delta v_i) & , \hat{D}_C \geq 0 \\ \hat{D}_C \geq \frac{|K|}{N \cdot g_i^C} d_C(e + \Delta v_i) & , \hat{D}_C < 0 \end{cases} \Rightarrow$$

$$\exists S_i \in K : \begin{cases} \hat{D}_C - \frac{|K|}{N \cdot g_i^C} \frac{d_C(e + \Delta v_i)}{2} \leq \frac{|K|}{N \cdot g_i^C} \frac{d_C(e + \Delta v_i)}{2} & , \hat{D}_C \geq 0 \\ \hat{D}_C - \frac{|K|}{N \cdot g_i^C} \frac{d_C(e + \Delta v_i)}{2} \geq \frac{|K|}{N \cdot g_i^C} \frac{d_C(e + \Delta v_i)}{2} & , \hat{D}_C < 0 \end{cases} \Rightarrow \exists S_i \in K : |\hat{D}_C - \frac{|K|}{N \cdot g_i^C} \frac{d_C(e + \Delta v_i)}{2}| \leq \frac{|K|}{N \cdot g_i^C} \frac{|d_C(e + \Delta v_i)|}{2}$$

The latter result could describe a ball, centered at $\frac{|K|}{N \cdot g_i^C} \frac{d_C(e+\Delta v_i)}{2}$ with $\frac{|K|}{N \cdot g_i^C} \frac{|d_C(e+\Delta v_i)|}{2}$ radius, which includes $\hat{D}_C$. Thus, it is equivalent to Lemma 2(a), but due to the unidimensional nature of our current setup the ball reduces to an interval as shown above. Because we are only interested in knowing whether $\hat{D}_C < 0$ or not, this time local constraints at sites neglect the term $\frac{|K|}{N \cdot g_i^C}$ (and thus Lemma 2(b) and (c)) and focus only on $d_C(e + \Delta v_i) \lessgtr 0$.

Given the above, we must choose a proper sampling function g_i^C which keeps the desired properties of g_i used in our generic sampling-based scheme. Beforehand, recall (Requirement 3) that at the application level our monitoring scheme requires the identification of the δ parameter to tune its accuracy together with the monitoring task $f(v(t)) \gtrless T$ itself. This parameter is set in a way that is oblivious to the details of the input domain and the tracking scheme. Therefore, the δ that is taken into consideration in our revised version is equivalent to the one in Sections 2.2 and 3.

For ease of presentation, and without any loss of generality due to isometry, let us assume that e is a zero vector. Recall from Section 3 that $\|\Delta v_i\| \leq U$. If $\Delta v_i \in C$, then $d_C(e + \Delta v_i) = -dist(\Delta v_i, \partial C) < 0$ and we simply assume that $\|\Delta v_i\|$ cannot receive its maximum value (U), simultaneously pointing towards T, without having Δv_i out of C. Therefore, $|d_C(e + \Delta v_i)| \leq U$ in this case. If $\Delta v_i \notin C$, then there must be a point $c' \in \partial C$ where the vector Δv_i exits C and obviously c' and Δv_i are collinear. Then, by also considering that $dist(\Delta v_i, \partial C)$ by definition involves the smallest possible distance of Δv_i from the boundary of C, we get:

$$|d_C(e + \Delta v_i)| = dist(\Delta v_i, \partial C) \leq \|\Delta v_i - c'\| \stackrel{colinearity}{\underset{0 \leq \alpha \leq 1}{=}}$$

$$\|\Delta v_i - \alpha \cdot \Delta v_i\| \leq \|\Delta v_i\| \leq U \Leftrightarrow |d_C(e + \Delta v_i)| \leq U \qquad (6)$$

Our revised sampling function takes the form of:

$$g_i^C = \frac{|d_C(e + \Delta v_i)| \ell n(1/\delta)}{U \cdot \sqrt{N}}$$

This sampling function, g_i^C, gives us the opportunity to perform the tracking process with reduced approximation error $\varepsilon_C \leq \varepsilon$ without increasing the expected sample size $|K|$ yielded by g_i. Details on this follow.

Approximation Quality. Recall that approximation quality refers to the ability of the estimation $\hat{D}_C$ to be within predefined distance, denoted by ε_C, from D_C with high probability. In the current unidimensional setup we use McDiarmid's Inequality [36] as the probabilistic tool replacing the functionality of Vector Bernstein's Inequality [31] of Section 2.2.

McDiarmid's Inequality [36]. Suppose $y_1, y_2, \ldots, y_N$ are independent variables and assume that a function θ satisfies $|\theta(y_1, y_2, \ldots, y_N) - \theta(y_1, y_2, \ldots, y_{i-1}, \hat{y}_i, y_{i+1}, \ldots, y_N)| \leq \beta_i$ for $1 \leq i \leq N$, where $\hat{y}_i$ replaces y_i with its estimation. Then, for any $\varepsilon_C > 0$:

$$Pr[E[\theta(y_1, y_2, \ldots, y_N)] - \theta(y_1, y_2, \ldots, y_N) \geq \varepsilon_C] \leq e^{\left(-\frac{2\varepsilon_C^2}{\Sigma_{i=1}^{N} \beta_i^2}\right)} \qquad \square \qquad (7)$$

The case of $\theta(y_1, y_2, \ldots, y_N) - E[\theta(y_1, y_2, \ldots, y_N)] \geq \varepsilon_C$ is symmetric. In our setting, y_is correspond to $d_C(e + \Delta v_i)$ and, since according to Corollary 2 our estimator is unbiased, we can employ McDiarmid's Inequality using D_C as

22

$\theta(.)$. Before doing so, we concentrate on bounding the β_i values. If we replace the $j-$th additive term in D_C with its estimation, $\frac{d'_C(e+\Delta v_j)}{N}$, the difference in the function value is:

$$|\sum_{i=1}^{N}\frac{d_C(e+\Delta v_i)}{N} - \sum_{[i\in\{1..N\}\setminus j]}\frac{d_C(e+\Delta v_i)}{N} - \frac{d'_C(e+\Delta v_j)}{N}| \leq$$

$$|\frac{d'_C(e+\Delta v_j)}{N}| \leq |\frac{d_C(e+\Delta v_j)}{N \cdot g_j^C}| \leq \frac{U}{\ell n(1/\delta)\sqrt{N}} = \beta_j = \beta$$

Now, applying Inequality 7 for the above (common) β we receive:

$$\Pr[\sum_{i=1}^{N}\frac{d_C(e+\Delta v_i)}{N} - \sum_{i=1}^{N}\frac{d'_C(e+\Delta v_i)}{N} \geq \varepsilon_C] \leq e^{\left(-\frac{2\cdot\ell n^2(1/\delta)\cdot\varepsilon_C^2}{U^2}\right)}$$

The above inequality satisfies our approximation quality requirements because it shows that the error in the monitored estimator will not exceed ε_C with known probability $1 - e^{\left(-\frac{2\cdot\ell n^2(1/\delta)\cdot\varepsilon_C^2}{U^2}\right)}$. Furthermore, note that this probability would be analogous should we have used Hoeffding's [26] instead of McDiarmid's Inequality as the former is a special case of the latter upon the function of interest involves mere averaging or summation. We just preferred to use the generalization provided by McDiarmid for ease of presentation.

Setting the right side of previous inequality equal to δ and solving for $\varepsilon_C = \varepsilon_C(\delta) = \frac{1}{\sqrt{2\cdot\ell n(1/\delta)}}U$ we get:

$$\Pr[D_C - \hat{D}_C \geq \varepsilon_C] = \Pr[D_C \geq \hat{D}_C + \varepsilon_C] \leq \delta \qquad (8)$$

It is important to note that $\varepsilon_C \leq \varepsilon$ and thus the monitored quantity in the unidimensional setting is more accurately monitored compared to our generic, multidimensional, sampling-based scheme (Section 2.2). Figure 6(h) shows the ε_C-zone for our running example using the safe zone C of Example 5 (details are included in the corresponding caption). Furthermore, we will shortly discuss Figure 9 which plots the ratio between the approximation error yielded by the vector Bernstein inequality (Inequality 2) and McDiarmids Inequality used in our revised scheme.

Tunable Accuracy. Let us now focus on the overall P_{FN} probability and the way it can be determined by δ. Lemma 5 formalizes our tunable accuracy guarantees. The proof is included in the Appendix, but we here briefly discuss an alternative way to go and reason about why we do not use it in Lemma 5.

For a false negative synchronization decision to be possible to occur we need $\hat{D}_C < 0$ and $D_C \geq 0$. Based on the latter observation and the approximation quality guarantee in Inequality 8, we are FN safe when $\hat{D}_C < -\varepsilon_C$ because then, with probability $1 - \delta$, $D_C < 0$ as well. $\hat{D}_C < -\varepsilon_C$ surely holds when $\frac{|K|\cdot d_C(e+\Delta v_i)}{N} < -\varepsilon_C, \forall S_i \in K$, because then summing for every sampled site we get:

$$\frac{|K|\cdot d_C(e+\Delta v_i)}{N} < -\varepsilon_C, \forall S_i \in K \Rightarrow \frac{|K|\cdot d_C(e+\Delta v_i)}{N \cdot g_i^C} < -\varepsilon_C \Rightarrow \hat{D}_C = \sum_{S_i \in K}\frac{d_C(e+\Delta v_i)}{N \cdot g_i^C} < -\varepsilon_C$$

For instance, given the fact that our expected sample size (i.e., the expected $|K|$) remains proportional to $\sqrt{N}$ (a property also summarized below), $\frac{|K|\cdot d_C(e+\Delta v_i)}{N} < -\varepsilon_C, \forall S_i \in K$ essentially translates to $-\varepsilon_C$ being proportionally above $\frac{d_C(e+\Delta v_i)}{\sqrt{N}}, \forall S_i \in K$. So, when $\frac{|K|\cdot d_C(e+\Delta v_i)}{N} < -\varepsilon_C, \forall S_i \in K$ we just showed that $P_{FN} \leq \delta$. But such an approach introduces a dependence on the network size which does not favor our approaches, since we expect $|K| \ll N$. Therefore, instead of the above restrictive requirement, in the first case of Lemma 5 we include a more sensible criterion. In particular, we require that all site vectors do not get too close to the boundary of C and that sites perform a number of sampling trials in each execution of the monitoring phase during an update cycle. Our proof reasons about this choice. Note that this requirement is restricted to the monitoring phase. During a partial synchronization, the coordinator simply probes vectors of sites of one, say the first, trial to decide the necessity of a full synchronization. Then, based on Inequality 8 we are always FN safe with probability δ during that phase.

Lemma 5. *The revised Sampling-Based GM Scheme in the convex safe zone (CV) context under the mapping of Lemma 4 yields:*

- $P_{FN} \leq \delta + 0.01 \quad$ *if M trials* $\bigwedge \forall S_i \in \{S_1, \ldots, S_N\}, \frac{|d_C(e + \Delta v_i)|}{\sqrt{2}} > \varepsilon_C$
- $P_{FN} = O(\delta^{\frac{|Z|M}{\sqrt{N}}}) \qquad\qquad$ *otherwise*

where $1 \leq M \leq \left\lceil \frac{log(0.01)}{log\left(e^{-0.042\sqrt{\ell n(1/\delta)N}}\right)} \right\rceil$ *and Z (more precisely Z(t)) denotes the set of threshold crossing sites. Thus, one can properly tune δ to obtain the desired FN probability.*

Proof. In Appendix. □
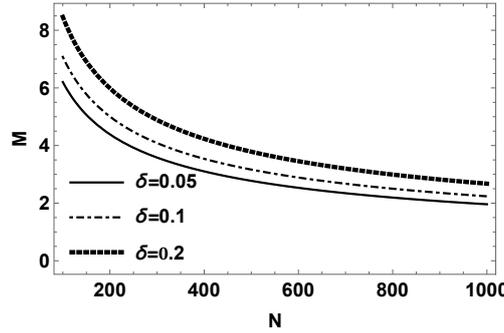
Figure 8: *M versus N for various values of δ in the CV context.*

Figure 8 shows practical values for M varying the value of N for different δ choices. It can easily be observed that, for highly distributed settings, 2-4 sampling trials can be sufficient for the first case of Lemma 5 to hold. In the proof of Lemma 5 we show that for $\hat{D}_C < -\varepsilon_C$ to hold, we need to ensure enough sample size $|K|$. Because of this, notice that in Figure 8 as δ decreases, so does M, because $|K|$ expectedly increases. This is in contrast to Figure 3, because in Lemma 2 we are interested in $\frac{|K|}{N \cdot g_i}$ and when δ decreases, $|K|$ expectedly increases while $\frac{1}{N \cdot g_i}$ decreases (decreasing δ increases g_i, g_i^C).

In Figure 8 observe that for lower N this revised scheme may require a few more sampling trials compared to Figure 3. However, this is not supposed to worsen the FP rate of the current scheme compared to the scheme of Section 2.2. The reason, as we discuss in Example 6, is that if $\partial C \cong \partial T$, a threshold crossing site producing an FP will have $g_i \geq g_i^C$. Obviously, if C is large $g_i \gg g_i^C$.

Regarding FNs, the bound of the first case of Lemma 5 is equivalent to that of Lemma 3, but under different conditions. The difference is that in order to achieve this bound we require all the sites to avoid the ε_C-zone, while the first case of Lemma 3 poses this requirement (for the $\varepsilon-$zone) only for the sampled sites of a single trial. Assume that ε was equal to ε_C for comparison purposes. Now, notice that the requirement of the first case of Lemma 3, for $\partial C \cong \partial T$, is interpreted to the center of the ball being away from C by $> \varepsilon + \|\frac{\Delta v_i}{2}\|$: $\|e + \frac{\Delta v_i}{2} - \partial C\| > \varepsilon + \|\frac{\Delta v_i}{2}\|$. This is already more restrictive than simply requiring $\|e + \Delta v_i - \partial C\| > \varepsilon$, because $\|e + \frac{\Delta v_i}{2} - \partial C\| > \varepsilon + \|\frac{\Delta v_i}{2}\| \Rightarrow \|e + \Delta v_i - \partial C\| > \varepsilon$, but not vice versa. So if we interpret the latter, looser condition in our current scheme, we are going to have $|d_C(e + \Delta v_i)| > \varepsilon_C$. For reasons explained in our proof we require the slightly different $\frac{|d_C(e + \Delta v_i)|}{\sqrt{2}} > \varepsilon_C$ condition to hold. This condition should hold for every site and not only the sampled ones, but this is due to the fact that $|d_C(e + \Delta v_i)|$ approaches zero near ∂C contrary to $\|\Delta v_i\|$.

The second case of the lemma is derived similarly to that of Lemma 3. Our analysis in Section 3 regarding the evolution of P_{FN} probability with time, also holds here for analogous reasons.

Example 6. *Getting back to our running example, we use Figure 6(i) to illustrate the basic differences between the sampling-based monitoring protocols of Section 2.2 and the current one, for the simple case of a single sampling trial. In Figure 6(i) we include the true threshold surface and the safe zone of Figure 6(f) (the discussion would be similar for*
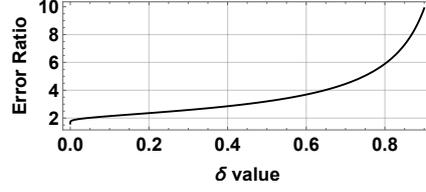
Figure 9: Error Ratio: Vector Bernstein (without simplification, Section 2.2) over the McDiarmid's Inequality used in our revised scheme.

the safe zone of Figure 6(g)). We focus on site S_4 (see Figure 6(a)) which now possesses a different $e + \Delta v_4 = [8, 3, 3]$. Because $e + \Delta v_4$ invades the threshold surface and the safe zone, $B(e + \frac{\Delta v_4}{2}, \frac{\|\Delta v_4\|}{2})$ crosses the threshold surface and simultaneously $d_C(e + \Delta v_4) > 0$ as Figure 6(i) shows. We again assume $N = 10$ but, in Figure 6(i) we only show $e + \Delta v_4$ to keep the illustration focused.

Consider no actual threshold crossing exists, i.e., both v is in the admissible region and $D_C < 0$ and in the rest of the sites no local constraint is violated. Hence, a FP synchronization will be caused in each of the sampling-based monitoring schemes depending on whether S_4 is sampled with g_i, in the framework of Section 2.2, or g_i^C in the current framework. Let us now examine which of the two monitoring protocols is more likely to produce a FP.

Based on $e + \Delta v_4$ mentioned above $\|\Delta v_i\| \approx 7$, while the signed distance is $d_C(e + \Delta v_4) \approx 4.5$. Thus, for $\delta = 0.05$ and $U = 17.3$, $g_4 = 0.39$ and $g_4^C = 0.25$. $g_4 > g_4^C$ and therefore if S_4 produces a single random number ρ (i.e., simultaneously executing the two protocols) and check $g_4 > g_4^C \geq \rho_i$, $g_4^C \geq \rho$ means that $g_4 \geq \rho$, but not vice versa. Thus, using the protocol of this section and the new g_i^C may avoid such a FP. g_4^C can produce a FP, while g_i does not, only if $e + \Delta v_4$ is positioned in the area between C and T due to the small distance between the two surfaces.

Therefore, the generic observation is that if $\partial C \cong \partial T$ as in Figure 6(g), the sampling-based monitoring scheme of the current section will produce less FPs compared to that of Section 2.2. This is because for sites crossing C, $g_i > g_i^C$ always holds. In practice, because C is not attached to each and every position of T and $\varepsilon_C \leq \varepsilon$, the scheme introduced in the current section also reduces FNs (see Section 6.6). □

Discussion and Summary. Summarizing the revised sampling-based GM scheme tailored for our novel unidimensional mapping, for $\delta \in (0, e^{-1})$ we derive:

$$\begin{cases} g_i^C = \frac{|d_C(e + \Delta v_i)| \ell n(1/\delta)}{U \cdot \sqrt{N}} & \text{Sampling Function} \\ \varepsilon_C = \frac{1}{\sqrt{2}\sqrt{\ell n(1/\delta)}} \cdot U & \hat{D}_C \text{ Estimation Error} \\ O(\ell n(1/\delta)\sqrt{N}) & \text{Expected Sample Size} \end{cases} \quad (9)$$

Hence for identical δ and U (due to Inequality 6), our revised sampling-based scheme provides equivalent (worst case) expected sample size, but reduces the approximation error since $\varepsilon_C \leq \varepsilon$. The amount by which the approximation error is reduced depends on the chosen δ. Figure 9 plots the error ratio yielded by the Vector Bernstein Inequality [31] over McDiarmid's Inequality [36]. It can easily be observed that for the δ values we consider in our sampling schemes the error is reduced by roughly a factor of 2 or more. This means that the monitored quantity is tracked more accurately and therefore, from a monitoring perspective, this fact should have the impact of reducing the amount of false decisions throughout the tracking process. Our experimental evaluation (Section 6.6) shows that this is indeed the case. For instance, the techniques devised in this section can optimize our generic sampling-based GM scheme of Section 2.2 further reducing the amount of FNs up to a factor of 6 or (also due to our proposed unidimensional mapping) providing additional decrement to the size of the communicated data by up to a factor of 5.

5. Related Work

Abundant works focuses on efficiently performing monitoring tasks over distributed data streams. Some of them were already mentioned in Section 1, while [1] presents a recent survey on related techniques. Here, we concentrate on studies closely related to the GM framework and site sampling techniques.

Monitoring General Threshold Functions. The basic operation of the GM framework was introduced in [5]. Our work, after pointing out the shortcomings that arise in highly distributed data streams, proposes techniques to effectively confront existing scalability issues of the GM framework. [5] also proposes a balancing optimization to the basic scheme to further reduce the communication cost of their approach. However, this balancing technique is merely a heuristic for which we experimentally (Section 6) show that is hardly adequate in highly distributed settings.

The GM framework has been enhanced in [21], where ellipsoidal instead of spherical local constraints are considered. These methods are orthogonal to the algorithms that we develop. However, [21] assumes that data follows a multivariate normal distribution; furthermore, [21] also suffers from scalability issues, since using ellipsoids instead of hyperspheres neither alters the fact that the higher N is, the larger the area being monitored nor reduces the cost of a false positive central data collection.

The recently introduced prediction-based GM [18, 19] constitutes another technique that is orthogonal to the methods that we present in our work. However, [18, 19] heavily depends on accurate predictions of the local vectors maintained at each site. Nonetheless, accurately predicting several vector components over many sites becomes increasingly harder with the increase of the network scale. This is also demonstrated in our experimental evaluation. The latest advancements regarding GM-based monitoring [14, 27] show that for certain function classes the performance of the tracking process can be improved by utilizing a convex safe zone idea. The basic principle behind the safe zone approach is to compute in hand a sufficiently large convex subset of the input domain that does not intersect the threshold surface. Using the convexity property of this subset, an exact convex hull (i.e., without considering covering spheres) can be checked for threshold crossings in a distributed manner. As we claim in Section 4 and experimentally validate in Section 6.6, the techniques we develop in this work are directly applicable in the setup of [14, 27] to amend scalability issues. Beyond that, based on the ideas of [14, 27], in the current work we present a novel mapping of the monitoring task from a multidimensional input domain of arbitrary dimension to an equivalent task on a unidimensional space and accordingly propose revised sampling-based tracking mechanisms tailored for the new setting.

A number of works design techniques that are geometric in nature but, contrary to our approach, focus only on specific types of functions. [37] considers functions with bounded deviation and introduces a tentative bound algorithm to monitor threshold queries in distributed databases (rather than distributed streams), while [38] focuses on vectorial top-k aggregation queries over distributed databases. The work in [12] couples sketch summaries with the GM framework focusing on join aggregates, special cases of L_2-norms and range aggregates (e.g., quantiles, wavelets, and heavy-hitters over the streams). The work in [13] utilizes GM for outlier detection in sensor networks, reducing the problem to multiple monitoring tasks, with each task involving only the pair of nodes whose similarity is to be monitored. [39] proposes an approach, for monitoring *heterogeneous* streams by defining constraints tailored to fit the specific data distributions of sites.

Site Sampling Techniques. The sampling component we develop is a part of our tracking schemes that jointly allow the monitoring of any generic function $f : \mathbb{R}^d \to \mathbb{R}$. This is the main breakthrough that distinguishes our overall contributions compared to individual site sampling techniques that can only handle linear functions such as counts and frequencies [32, 33, 34] or second frequency moments [40]. Besides this crucial distinction, our sampling component alone possesses more generic characteristics compared to existing site sampling approaches [32, 33, 34, 40]. These characteristics can be summarized as follows: (a) our analysis, from approximation quality issues to extracted estimators and expected communication savings is *multidimensional* in nature. This also holds for the extensions discussed in Section 4 which may take advantage of the unidimensional mapping we propose, but the sampling function and extracted estimator are derived based on the (topological) relation among site local vectors and convex bodies (safe zones) of arbitrary dimensionality. On the contrary, [32, 33, 34, 40] define sampling schemes operating on a single dimension, (b) our techniques are tailored to support *monitoring* procedures while [32, 33] focus on one-shot queries, (c) Our algorithms do not incorporate any assumption about local input monotonicity or boundedness and are capable of handling *unbounded, non-monotonic local inputs* (updates). The techniques in [32, 33, 34, 40] assume bounded updates, while [32, 33, 34] are restricted to positive inputs only. (d) [32, 33, 34, 40] are focused on ensuring *accuracy relative* to the current global frequency or count which can be known only after acquiring the sample. Our algorithms abide by predetermined accuracy constraints, which is a necessary feature in our setting. Due to the above limitations, [32, 33, 34, 40] are not applicable in our setup.

In [35], we presented our generic sampling-based geometric monitoring scheme for scalable, distributed monitoring of non-linear functions. In the current work, we build on the work of [35] coming up with a revised Lemma 2.

In Section 4 we extend and optimize our sampling-based framework upon operating in the context of the convex safe zone idea [14, 27]. The latter concept describes the most recent advances in GM based functional monitoring. We accordingly enhance our experimental evaluation to practically test our aforementioned extensions in Section 6.6. Eventually, Section 7 also adds up to the work of [35] by providing a comprehensive study on sum- (instead of average-) parameterized function monitoring that has not been analyzed in any of the GM related works mentioned in our above discussion.

6. Experiments

We developed a simulation environment in Java in order to evaluate the performance of our techniques. For our sampling-based monitoring techniques we use **SGM** to refer to the generic sampling-based monitoring scheme presented in Section 2.2, materialized by the sampling function of Section 3 upon using a single sampling trial, i.e., $M = 1$. This forms a worst case for the FN rate of the technique. We use **M-SGM** to denote the same sampling-based monitoring scheme upon employing a number of trials M as prescribed in Lemma 2(c). Since we show that the two alterations exhibit equivalent communication performance (and thus FP rates), in the text we refer to SGM for either of the two, but use the worst case alternative (SGM) upon commenting on *FN* rates. We compare the communication cost (number of messages) as well as the number of FP and FN synchronization decisions of SGM, M-SGM against the other GM related techniques proposed in the literature. More precisely:

- The Geometric Monitoring framework of Section 1.1 introduced in [5], termed **GM**.

- The GM framework enhanced with the balancing optimization presented in [5], termed **BGM**. In BGM, the coordinator tries to probe a subset of Δv_is that may partially cancel out the crossing ones due to their different direction. If such a subset exists, it knows that $v(t)$ has not crossed the threshold without requiring a full synchronization. Please refer to [5] for more details.

- The Prediction-Based Geometric Monitoring Framework, and in particular the CAA technique proposed in [18, 19], henceforth referred to as **PGM**. We adopt a Velocity-Acceleration predictor and present the best performance PGM shows upon varying the window according to which predictions are formed from 3 to 10 measurements (roughly the amount of data updates received hourly).

We emphasize that these competitor approaches are also orthogonal to our sampling-based SGM framework. Despite this fact, to better perceive the benefits of our novel SGM approach and expose its features, in our experimentation we form a worst case scenario for SGM by not applying any orthogonal approach on top of it. Moreover, note that BGM and PGM are not orthogonal to each other, since the CAA approach [18, 19] switches among monitoring models instead of balancing their drift vectors.

GM, BGM, PGM and SGM are generic and can serve the monitoring purposes of any given function employing balls in their local threshold crossing checks. Moreover, we experiment with the convex safe zone idea of [14, 27] discussed in the introductory part of Section 4, denoted by **CVGM**, and the revised version of our sampling-based scheme, denoted by **CVSGM**, presented in Section 4.2. This is because CVSGM essentially optimizes SGM in the context of the convex safe zone idea. For CVSGM we do not include a corresponding, separate line for M-CVSGM, first because we have already shown for SGM that the effect of a roughly equivalent M on FPs is marginal, second because we exhibit that we do not need more than one trials for the scheme of Section 4 to provide improvements in terms of faulty decisions compared to SGM, M-SGM and third in order to keep the respective plot readable. We study CVGM, CVSGM separately in Section 6.6, however, we stress that our comparative analysis remains holistic combining the performance results of all the competitors. Additionally, since our CVSGM approach takes advantage of our novel unidimensional mapping (Section 4.1), we also examine its performance in terms of transmitted bytes along with the number of transmitted messages.

Data Sets. We utilize two real world datasets. The first dataset is the Reuters Corpus (RCV1-v2) [41] data, termed *Reuters*, also used in related work [5, 21, 18]. It is composed of 804414 records of news stories which have been categorized and have been tagged with a list of terms (features). As in previous works [5, 21, 18], we focus on tracking the terms *Febru, Ipo, Bosnia* and their co-occurrences with the Corporate/Industrial category. We monitor
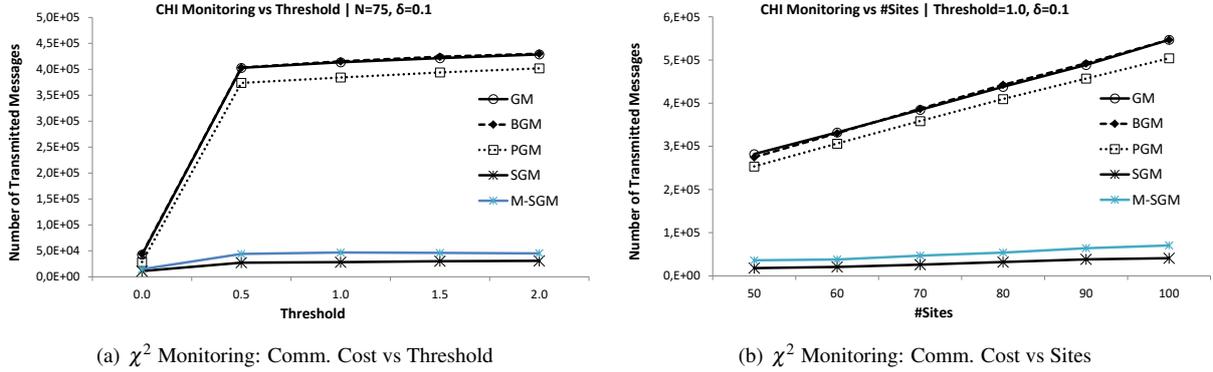
(a) χ^2 Monitoring: Comm. Cost vs Threshold

(b) χ^2 Monitoring: Comm. Cost vs Sites
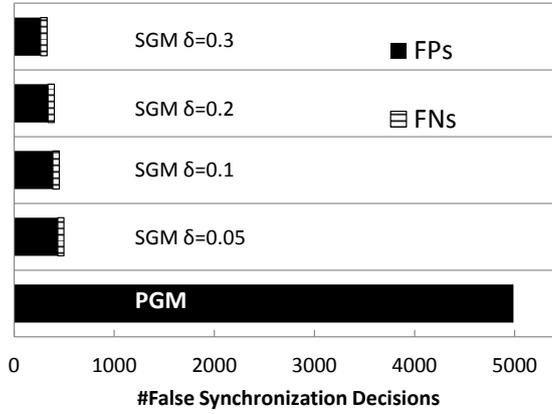
(c) χ^2 Monitoring: Sensitivity to δ

Figure 10: Reuters Data Set: Performance of our Techniques on χ^2 Monitoring

the relevance between a (term, category) pair using the χ^2 function which had been used in previous GM-related work (please refer to [18, 19] for more details). We use a sliding window of 200 documents for this dataset which is roughly the amount of news stories received daily [18, 19]. Since the number of records in the data is limited, we use the Reuters data to evaluate our algorithms in medium sized distributed settings of size $N = 50$ to $N = 100$ and provide some initial comparisons with related works [5, 18, 19] which also use up to 100 sites.

The second dataset, termed *Jester* [42], contains 4.1 million ratings between -10 and 10 on 100 jokes from 73421 users. We use this dataset to approximately monitor the sum in buckets of equi-width histograms of the above rating range, based on L_∞ distance as well as the Jeffrey Divergence (JD) [43]. More precisely, we use these functions to measure the distance (cost) of encoding the current global histogram at each time instance, to the one communicated during the last central data collection. In addition to L_∞ and JD, the third function we experiment on, is the tracking of the Self-Join (SJ) size [19, 12, 6] (essentially the L_2 Norm) of the vector hosting the expected counts in the aforementioned histogram buckets. Since users provide ratings for 100 distinct cases, we utilize a sliding window of 100 observations for this dataset. Regarding the degree of distribution, we vary N between 100 and 1000.

Metrics and Parameter Settings. Throughout our study, for each (dataset, function) pair we initially measure the number of communicated messages while varying the value of the threshold T, keeping the number of sites to a fixed value that equals the average ($N = 75$ for Reuters and $N = 500$ for Jester) of the aforementioned distribution ranges.

Then, for the average threshold, we investigate the communication cost for increasing network scales. In the Jester dataset where larger network sizes can be tested, we also investigate the cost of messages per site (instead of just the total number of messages), as this gives an indication on how the cost of each site scales when the number of sites

increases. For large numbers of sites and for constrained applications such as sensor networks, where an increased number of transmitted messages results in reduced network lifetime, this per site cost should ideally remain steady (or slowly increase) as we scale to larger networks.

For SGM, we also vary the tolerance constraint i.e., the application defined probability δ, between 0.05 and 0.3 with a default value of 0.1, and perform a sensitivity analysis on the number of false (FP, FN) synchronization decisions in comparison with the FPs produced by the best (at each case) of the competitors. In addition, for CVSGM which incorporates our unidimensional mapping and can resolve FPs by centralizing a single value per site instead of a vector, we also measure the overall number of transmitted bytes compared to SGM.

Main Findings. The experimental analysis demonstrates that our sampling-based SGM method significantly reduces the number of transmitted messages and the number of false positives, with the benefits becoming more profound when the number of sites increases. Please note that, since in GM each false positive results in communication from all sites and the coordinator, significantly reducing the number of false positives translates into a corresponding bandwidth reduction (or energy consumption reduction, in case of applications with sensor sites) on *each* site. This is validated by the scalability experiment performed on the (larger) Jester dataset, where the corresponding benefits *per site*, compared to GM, increase in larger network setups. Moreover, the false negatives of SGM (and thus M-SGM) are in all cases lower than the specified tolerance parameter δ while M-SGM, SGM have marginal differences in their FP rates in all cases. Finally, our results demonstrate that the CVSGM approach that revises SGM in the convex safe zone context can indeed optimize SGM by further reducing the number of false (FP,FN) decisions and the amount of communicated bytes due to the unidimensional mapping it employs.

6.1. Reuters Dataset Monitoring

We first focus on the Reuters dataset. Figure 10(a) and Figure 10(b) present the communication cost of GM, BGM, PGM and SGM while varying the threshold and the scale of the distributed network for the default value of $\delta = 0.1$ when monitoring the χ^2 function. We present results for the "Febru" term; the trend is similar for "Bosnia" and "Ipo".

In both figures, the plots corresponding to GM, BGM almost coincide, showing that in this data set the balancing optimization does not reduce communication cost. The reason is that, when many sites cross the threshold surface moving towards similar directions, there exists an additive effect on their Δv_i's. Therefore, the coordinator probes almost all of the non-crossing sites so as to balance the added drift. PGM performs only slightly (about 20 percent) better than BGM and GM.

SGM, however, reduces bandwidth consumption from 3.8 times for $T = 0$, to more than an order of magnitude compared to the other candidates for the rest of threshold values (Fig. 10(a)). In addition, SGM requires less than an order of magnitude (between 13 and 16 times) messages than its competitors across different network scales (Fig. 10(b)). The benefits of SGM increase with the number of sites as, not only does it reduce the number of FPs (also depicted in Figure 10(c)), but also in most potential FPs, it only requires transmission from $O(\sqrt{N})$ sites, in comparison to $O(N)$ sites for the other techniques.

Figure 10(c) presents a sensitivity analysis of the effect of δ on the number of FP, FN decisions. Recall that FPs are responsible for the unnecessary portion of communicated messages. As a result, our sensitivity analysis also exposes the trade-off among bandwidth consumption caused by FPs and accuracy in terms of FNs for SGM. The horizontal bars depict the number of FP decisions of PGM (which, as we just showed, performs better than GM and BGM) compared to the FP and FN decisions of SGM, under different δ values ranging between 0.05 and 0.3. FPs and FNs for each given δ are drawn in stacked bars as explained by the corresponding legends, while the overall length of the bars represents the total number of false decisions.

SGM yields more than an order of magnitude reduction in the amount of false decisions (represented by the total length of the stacked bar of FP, FN counts) compared to the second best alternative, PGM. Figure 10(c) demonstrates that increasing δ causes FP decisions to be reduced by more than 15% in each bar of the histogram. FNs slightly increase with increasing δ values, but are very rare in all cases. The reduction in FP with increased δ values is easily understood, since the expected sample size is proportional to $\ell n(1/\delta)\sqrt{N}$. Thus, increasing δ decreases the sample size and, thus, the monitored convex hull responsible for FPs. Regarding FNs, they are rare in all cases. For instance, for SGM and $\delta = 0.05$, ~8000 updates arrive per site (see the size of the Reuters data set and $N \leq 100$) and the number of FNs is just 61, which corresponds to a ratio lower than 0.01. SGM typically results in fewer FNs (in this experiment, always by at least a factor of 5) than the δ according to which the tolerance to FNs is tuned.

(a) L_∞ Monitoring: Comm. Cost vs Threshold

(b) L_∞ Monitoring: Comm. Cost vs Sites

(c) L_∞ Monitoring: Sensitivity to δ
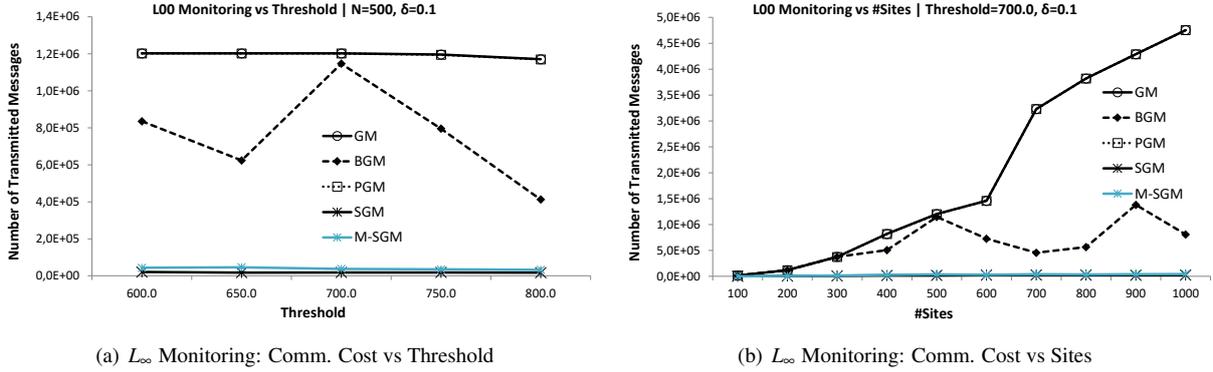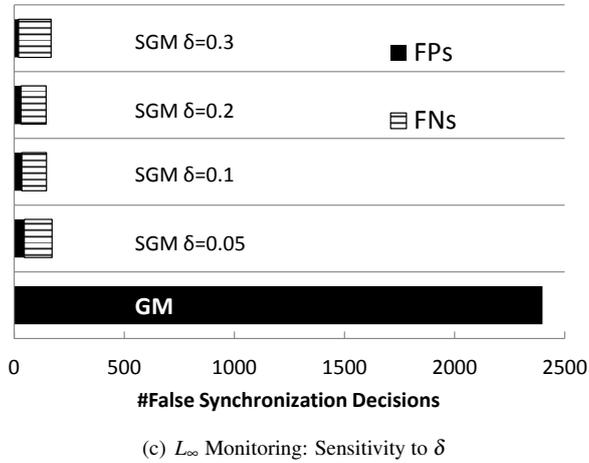
Figure 11: Jester Data Set: Performance of our Techniques on L_∞ Monitoring

6.2. Jester Dataset Monitoring

We now focus on the Jester dataset which, due to its larger size, allows us to also perform tests in larger topologies. Please note that in all the plots of this section, the lines corresponding to SGM, M-SGM almost coincide and thus, only the line of M-SGM is visible. The default number of sites used in this dataset is 500. Two general observations follow from Figure 11, Figure 12:

• In this larger scale dataset, the performance of the PGM approach is equivalent to the baseline GM. This validates our claim in Section 5, where it was noted that PGM may perform well in small to medium sized network distributions, but increasing the network scale renders the existence of inaccurate predictors in some sites more probable and, thus, PGM becomes more prone to FPs.

• Figure 11 shows that balancing may be more of help in reducing the bandwidth consumption in that particular (function, dataset) pair. Nonetheless, in Figure 12 where only the function and the threshold value (surface) is altered, BGM provides no improvements. This, together with its poor performance on the Reuters data experiments, comes as no surprise, as BGM adopts a simple heuristic, hoping to probe sites with drift vectors of opposite direction compared to the threshold crossing ones. Hence, contrary to SGM, BGM does not guarantee communication reduction. Furthermore, contrary to our proposed SGM approach, none of the BGM or PGM mechanisms provide a way to tune the expected bandwidth consumption according to posed accuracy standards.

Focusing on specific figures, in Figure 11(a) we point out that the bandwidth consumption achieved by our SGM (star-marked line approaching the x axis in the figure) approach is from 25 to 64 times lower than the best alternative (BGM). Moreover, upon varying the network scale between 100 and 1000 sites in Figure 11(b), the communication
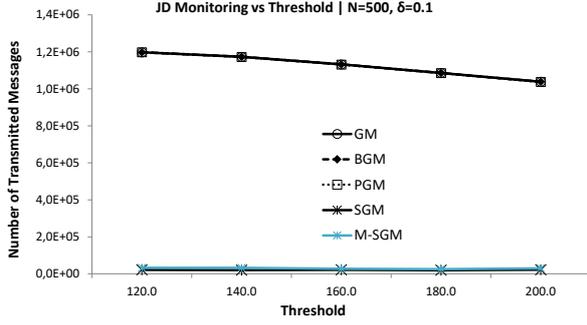
(a) JD Monitoring: Comm. Cost vs Threshold
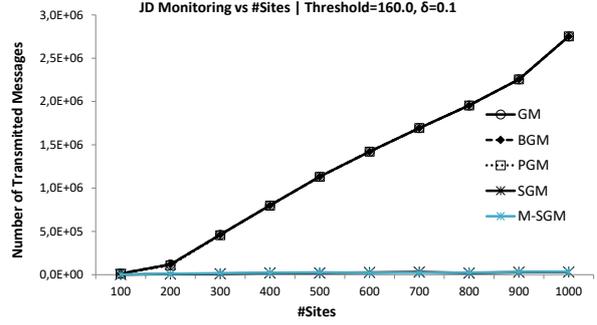
(b) JD Monitoring: Comm. Cost vs Sites
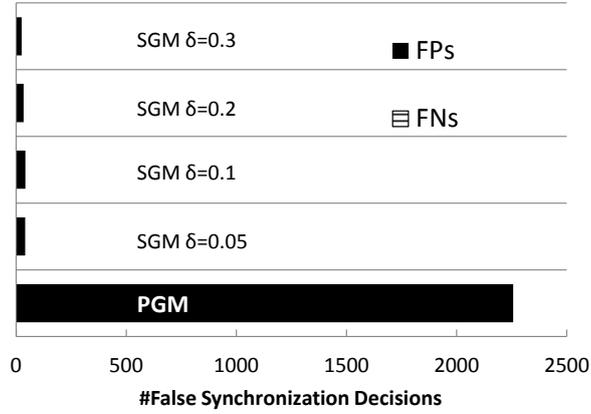
(c) JD Monitoring: Sensitivity to δ

Figure 12: Jester Data Set: Performance of our Techniques on Jeffrey Divergence Monitoring

cost reduction by SGM reaches a factor of 64 for $N = 900$, while constantly being at least 20 times lower compared to BGM, for different degrees of network distribution. Comparing the cost of SGM against PGM or GM, for instance when $N = 1000$, SGM results in 206 times fewer messages, with at least 20 times fewer messages for any other network scale.

Concluding our discussion on L_∞, the sensitivity analysis in Figure 11(c) shows the trade-off among unnecessary bandwidth consumption due to FPs and accuracy in terms of FN decisions for different δ's. SGM is compared to the FPs in GM, as BGM causes full synchronization only progressively, thus FPs cannot be counted in a distinct manner. As this figure shows, the number of FP decisions tends to be reduced upon increasing δ, while the number of FNs tends to increase, both of which are the expected behavior. Given that about 4850 updates arrive in every site of the network for the Jester dataset, the 148 FNs for $\delta = 0.3$ correspond to a ratio of just 3% false negatives, while the corresponding ratio for the $\sim$110-120 FNs and for the rest of the examined δ values never exceeds 2.3%. Hence, once again the amount of FNs is less than δ.

In Figure 12 we focus on monitoring the Jeffrey Divergence. In Figure 12(a) and Figure 12(b), all three competitive techniques (GM, BGM and PGM) exhibit comparable performances. Our SGM framework reduces the consumed bandwidth by up to a factor of 56 across different thresholds for $N = 500$ and the communication gains progressively approach two orders of magnitude when increasing the network scale to $N = 1000$ sites (Fig. 12(b)). Regarding the number of false synchronization decisions and the sensitivity on the chosen δ, Figure 12(c) shows the absence of FNs and the reduction of FPs by about 20% as we increase δ above 0.1.
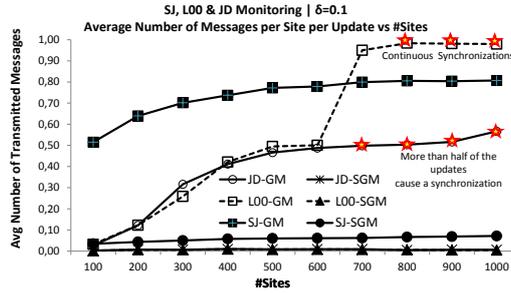
Figure 13: Average Number of Messages Transmitted by *each Site* per Data Update

Additional Results: The total communication cost (across network scales) of Self-Join (SJ) size monitoring is presented in Section 6.6 - Figure 16 (a) together with the CVGM and CVSGM performance. There, we include a single line for SGM, since as all the above illustrations show M-SGM has equivalent performance. For SJ, SGM yields more than an order of magnitude reduction in transmission compared to GM, PGM. BGM can fall short compared to SGM by up to a factor of 8, but typically yields between 2-3 times worse communication cost in terms of the number of transmitted messages.

6.3. Messages Per Site in Large Networks

In order to validate our claim regarding scalability to larger network distributions and to resource constrained environments, apart from measuring the total communication cost (number of messages) transmitted in the network, we also study the average number of messages transmitted by *each site*. That is, we measure the average number of messages a site transmitted per each update of its data. An average value close to 1 indicates that each site in the network transmitted a message after each data update, which is equivalent to a synchronization process.

Figure 13 presents the average number of transmissions per site and data update for the GM and SGM schemes in L_∞, Jeffrey Divergence and in Self-Join size monitoring when varying the network scale. Figure 13 shows that increasing the scale in GM (and the other alternatives that have similar performance to GM in Fig. 11(b), Fig. 12(b) and Fig. 10(b)) results in a continuous increase in the number of transmitted messages per site. This is more evident in L_∞ monitoring where, starting at 800 sites, GM behaves as the naive choice of continuous central data collection, since at least one site exhibits a local violation, which results in communication by all other sites as well. For Self-Join size, it is not difficult to see that even for mediocre network scales of 100 sites, more than half of the updates caused a synchronization in GM, while this percentage exceeds 80% for topologies with 800 or more sites. In Jeffrey Divergence monitoring this effect is less pronounced until $N = 500$, but still each site transmits a message in over half of its data updates for larger network sizes. On the contrary, the SGM approach is very slightly affected by the increase in network distribution, since the number of sampled sites increases with the square root of the network size. Thus, the benefits of SGM not only increase with larger network topologies, but it is also more appropriate for resource constrained environments, such as battery-powered sensor networks, where it is desirable to reduce the amount of communication per site in order to prolong the network lifetime.

6.4. Duration of FNs

A discriminating factor between FP and FN synchronization decisions is due to how they affect the monitoring process. FP decisions have an instant effect, as the coordinator becomes aware of a FP, and the bandwidth overhead it caused, by the end of a synchronization. Contrary to FPs, a FN decision has both the instant effect of saving bandwidth (while it should actually not!), as well as a persistent effect. In particular, upon a FN occurrence and as long as the threshold crossing lasts, the application continues to assume that the monitored function lies on one side of T, while $f(v(t))$ had actually switched sides. This misconception is maintained by the coordinator until either a synchronization (FP or not) takes place, or $v(t)$ again switches to its initial side with respect to T. In Section 3 we argued about the fact that the upper bound on the probability of FNs further decreases when threshold crossings span multiple monitoring phases. Consequently, we enhance our study by concentrating on the anticipated duration of a

Table 3: FN Duration - χ^2 Monitoring						

	Threshold					
	0,5		1		1,5	
#Sites	Mode	Mdn	Mode	Mdn	Mode	Mdn
60	1	3	1	3	1	2
70	1	4	1	3	1	2
80	1	3	1	3	1	3
90	2	3	1	4	1	2
100	2	3	1	2	1	1

Table 4: FN Duration - *SJ* Monitoring						

	Threshold					
	190		200		210	
#Sites	Mode	Mdn	Mode	Mdn	Mode	Mdn
600	2	2	1	1	1	1
700	1	1	1	1	1	1
800	1	1	1	1	2	1
900	1	1	1	1	1	1
1000	1	1	1	3	1	1

FN decision, indicatively providing results for χ^2 and *SJ* monitoring. These results correspond to SGM as it stands for the worst case FN rate. We report holistic aggregates and in particular the Mode and Median (denoted by Mdn) statistics for FN duration .

As both Table 3 and Table 4 demonstrate, the most frequent situation is the one where our proposed SGM approach compensates the coordinator for a FN decision immediately after its occurrence, i.e., the corresponding duration is 1 time unit. This is expressed by the "Mode=1" value in the vast majority of the cases (listed in the corresponding tables). On the other hand, interpreting the cited median values, we can observe that most of the times SGM needs no more than 3 time units to compensate for a FN for χ^2 (Table 3), while requiring 1 time unit (i.e., Mode=Mdn) for SJ (Table 4).

Overall, we can conclude that even when SGM does produce FN decisions (recall that JD is practically FN free) it possesses the ability to immediately compensate the tracking process for them. This is due to the fact that for low δ values, the probability of missing the event of a threshold crossing in consecutive time units decreases with the number of time units.

6.5. Comparison with a Bernoulli Sampling Variant

A question that naturally arises is what if we choose a simpler g_i, instead of the one proposed in Section 3, which uses the SGM framework but naively samples sites with equal probability, i.e., performs Bernoulli sampling. For a fair comparison with our techniques, in case of this Bernoulli sampling variant each site's g_i is set to $\ell n(1/\delta)/\sqrt{N}$ yielding analogous expected sample size ($O(\ell n(1/\delta)\sqrt{N})$) as the function that we proposed in Section 3 and M=1 is used. Please note that the Bernoulli sampling variant still utilizes optimizations that we proposed in this paper, such as the observation that sampled sites do not need to scale their Δv_i vectors by $1/g_i$ (Lemma 2).

We compare SGM incorporating the g_i of Section 3 (as in all previous evaluations), with the Bernoulli sampling variant in terms of the number of transmitted messages for different network scales. Figure 14 presents the respective comparison pairs for each monitored function (L_∞, *JD*, *SJ*) in the Jester dataset. Pairwise comparisons shown in the figure include SGM's performance marked with the respective function abbreviation (e.g. L_∞-SGM), against the respective variant (e.g., L_∞-Bernoulli).

According to Figure 14 we observe the following: (a) in SJ monitoring, SJ-Bernoulli performs 2-3 times worse than our proposed SJ-SGM across the examined network scales, (b) in Jeffrey Divergence monitoring, JD-Bernoulli
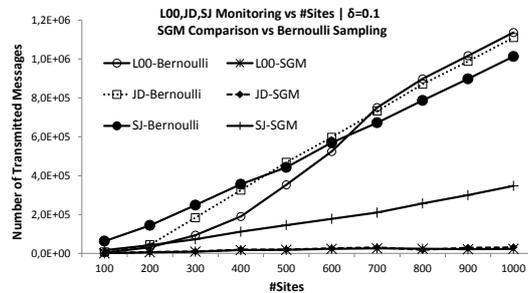
Figure 14: Comparison of SGM vs Bernoulli Sampling Variant

(a) χ^2 Monitoring: Transmitted Messages vs Sites

(b) χ^2 Monitoring: FPs vs δ

(c) χ^2 Monitoring: Transmitted Bytes vs δ
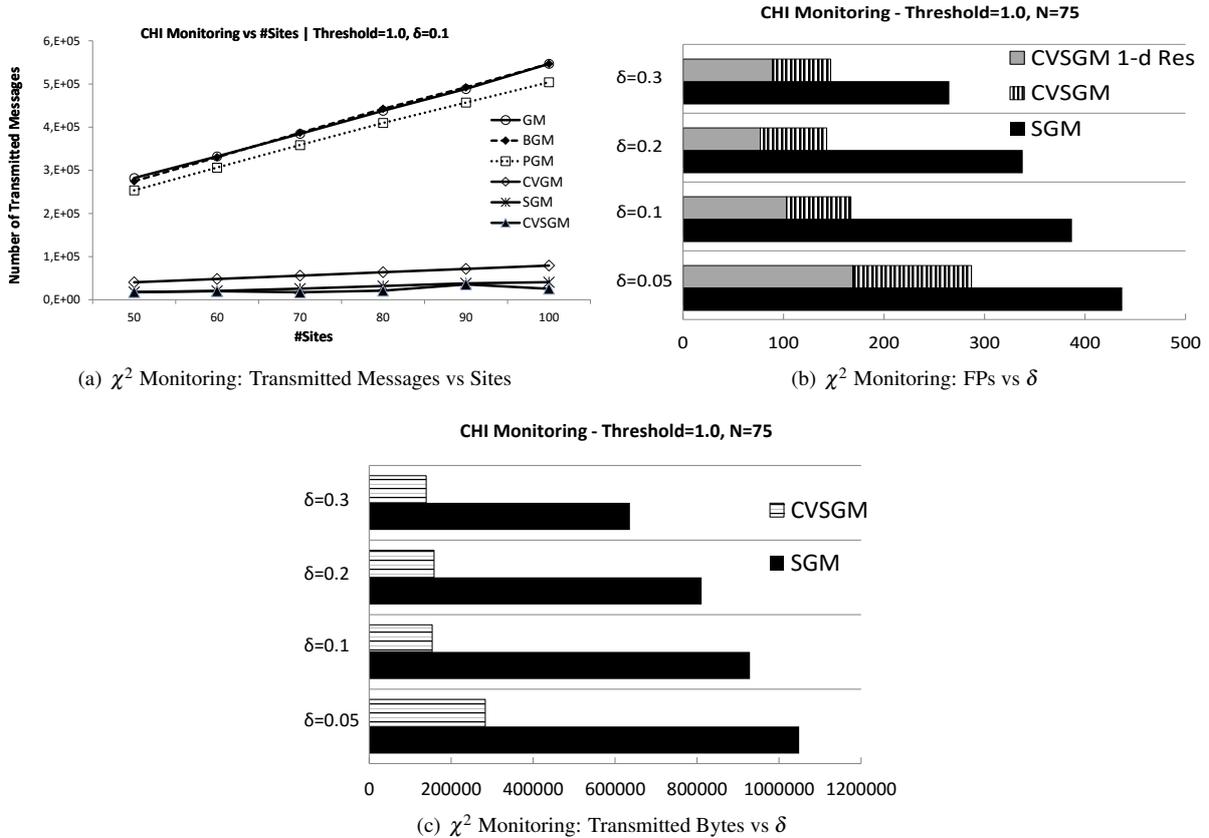
Figure 15: Reuters Data Set: Impact of Revised Sampling on χ^2 Monitoring

falls short from 6 to 36 times compared to our JD-SGM and (c) $L_\infty-$Bernoulli provides from 5 to 50 times more transmitted messages than our $L_\infty-$SGM proposal. These ratios exhibit the ability of our proposed g_i to decrease communication burden compared to other, straightforward, sampling function choices. The differences in the performance with the Bernoulli sampling variant are mainly attributed to the fact that, contrary to the g_i we proposed in this work (Section 3), Bernoulli sampling does not take into consideration the size of the local deviation vector $\|\Delta v_i\|$. Thus, sites with small deviations that less affect the global average but lie near the threshold surface, are equally probable to be included in the sample as peers with large $\|\Delta v_i\|$ that push the global average away from it. A plausible characteristic is that such a behavior is not allowed by our proposed sampling function which incorporates $\|\Delta v_i\|$ in its calculation formula.

6.6. CV Related Experiments

In this section we concentrate on studying the impact of the ideas introduced in Section 4 throughout the monitoring process. First we present the communication cost (number of transmitted messages) when applying the convex safe zone concept [14, 27] on our monitored functions across various network scales without applying any of our sampling techniques. This approach is denoted as CVGM in our graphs, where we also include the other approaches to provide a holistic picture on their respective performances.

Next, we focus on our sampling-based schemes and apply our revised tracking method of Section 4.2 that takes advantage of the novel unidimensional mapping introduced in Section 4.1, which we term as CVSGM. Recall that, based on our discussion in Section 4.1, our proposed mapping can reduce the amount of communicated data proportionally to the data dimensionality, while Section 4.2 shows that our revised sampling-scheme reduces the approximation error roughly by a factor of 2 (see Figure 9) for the practical δ values we examine in our evaluation. Therefore for CVSGM,

(a) *SJ* Monitoring: Transmitted Messages vs Sites

(b) *SJ* Monitoring: FPs vs δ

(c) *SJ* Monitoring: Transmitted Bytes vs δ
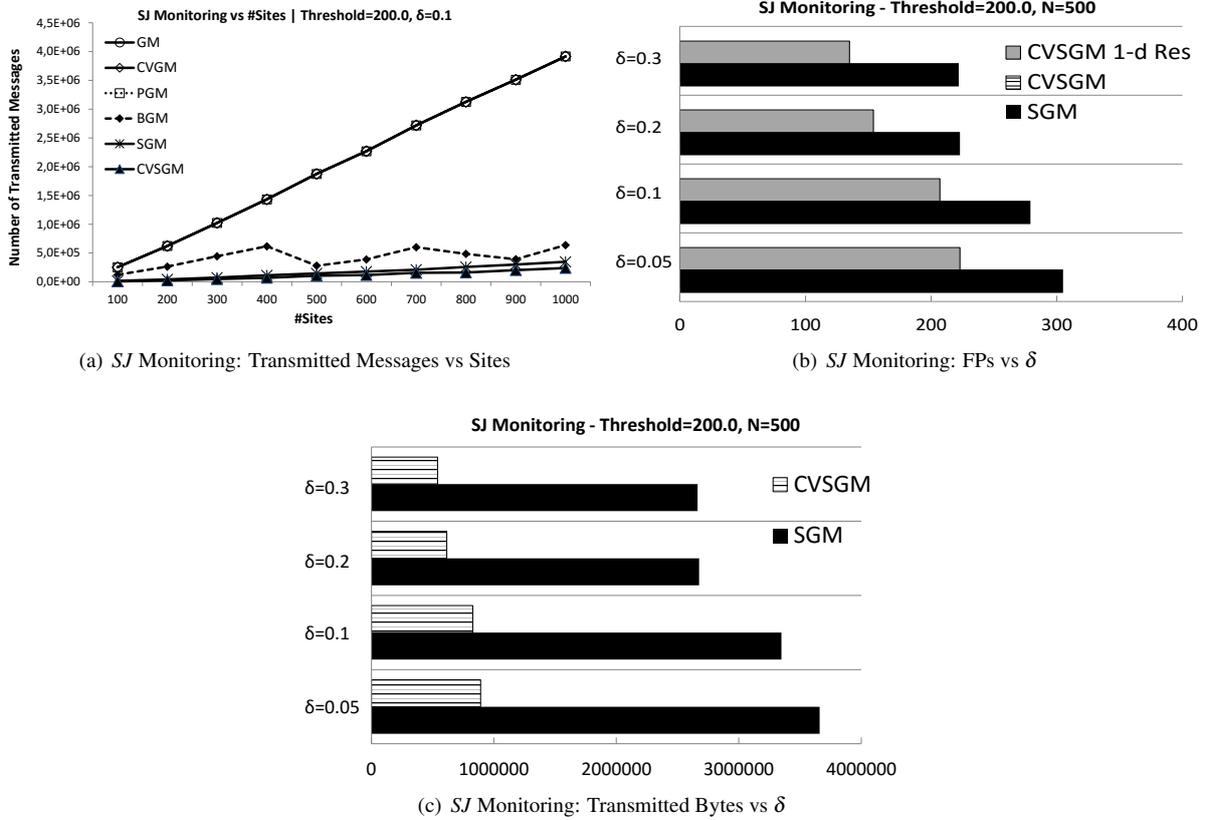
Figure 16: Jester Data Set: Impact of Revised Sampling on *SJ* Monitoring

apart from measuring the total number of transmitted messages versus network scale, we analyze both the number of FPs and FNs they yield as well as the number of communicated bytes. The first is to account for the effect of the reduced approximation error on the number of false decisions, while the second illustrates the cumulative effect of our unidimensional mapping. Since we found only slight (positive) effect of CVSGM on Jeffrey Divergence monitoring, we next focus on χ^2, Self-Join size and L_∞ functions. In the current section, for each of the tested cases, we computed the convex safe zone as the subset of the admissible region (i.e., where v can move without causing the monitored function to cross the threshold surface) that corresponds to the maximal non-intersecting hypersphere [27].

Starting with the CVGM approach, as Figure 15(a) shows, it provides reduced number of messages compared to GM, BGM, PGM that is only 2-3 times more than the respective number of SGM for network scales up to 100 sites. However, Figures 16(a) and 17(a) expose the scalability issues that CVGM encounters for larger network scales (up to 1000 sites) where its performance coincides with the worst competitors in the graphs. This validates the claim we made at the introductory part of Section 4.

Turning to our sampling-based techniques, based on Figure 10(c), we see that the vast majority of false synchronization decisions come from FPs and we found that CVSGM provides an identical number of FNs for the χ^2 and SJ functions. Hence, for these functions we focus on CVSGM's impact on FP decisions, while for L_∞, based on Figure 11(c), we are primarily concerned with FNs.

Figure 15(a) and Figure 16(a) depict, apart from the performance of non sampling-based tracking schemes, the improvement in terms of the number of transmitted messages provided by CVSGM compared to SGM in χ^2 and SJ monitoring (respective lines approach the horizontal axis). In these two graphs CVSGM can reduce the number of transmitted messages by 25% to 60% under different network scales while keeping the δ parameter in its default 0.1 value. Let us now examine the reason for this improvement. Figure 15(b) and Figure 16(b) depict the number of FP

35

(a) L_∞ Monitoring: Transmitted Messages vs Sites

(b) L_∞ Monitoring: FNs vs δ
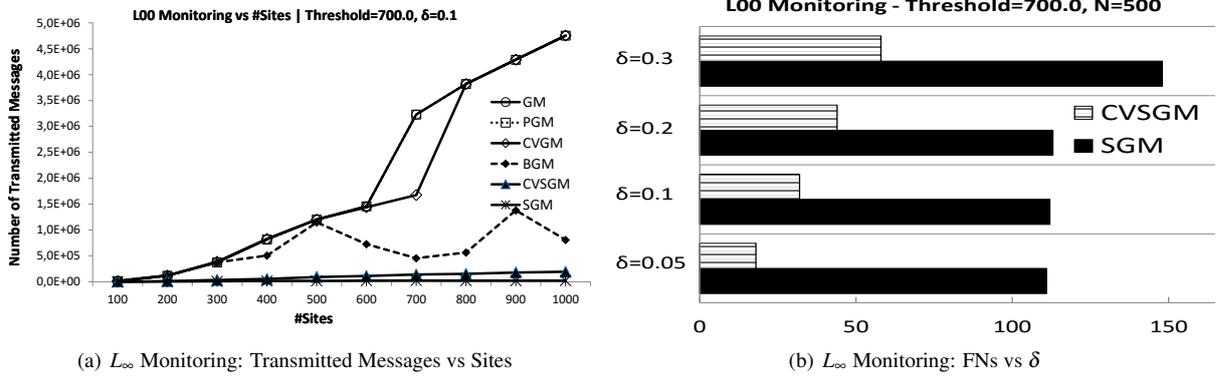
Figure 17: Jester Data Set: Impact of Revised Sampling on L_∞ Monitoring
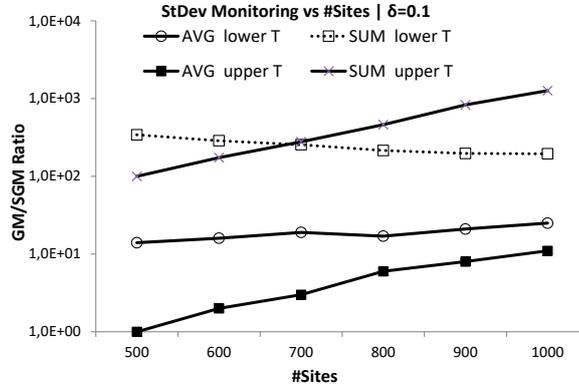
Figure 18: GM/SGM Ratio: Sum- vs Average-parameterized *stdev*.

synchronization decisions for different δ values for χ^2 and SJ respectively. The stacked bar including CVSGM and "CVSGM 1-d Res" shows the overall number of FPs for CVSGM and we will return to analyze its components in the next paragraph. It can be observed that CVSGM can on average halve the amount of FPs for χ^2 monitoring, while the respective improvement for SJ monitoring is on the average about $1/3$. These results are attributed to the ability of our proposed CVSGM approach to reduce the amount of false decisions (also providing equivalent FNs) compared to SGM due to the reduced approximation error (Figure 9) which, expectedly and as we just showed practically, makes the tracking process more accurate.

Proceeding one step further, of particular interest in Figures 15(b) and 16(b) is the stacked bar including CVSGM and "CVSGM 1-d Res". The latter counts the portion of FP decisions of CVSGM which were resolved by having sites transmit a single value for their signed distance (Section 4.1) instead of their $d-$dimensional vectors, in which case the coordinator simply checks if the average signed distance is negative (Lemma 4). *Contrary to the width of SGM and CVSGM, for CVSGM 1-d Res, the closer the bar to CVSGM, the more we gain because of our proposed unidimensional mapping.* As CVSGM 1-d Res in Figure 15(b) demonstrates, 40% to 50% of the FPs caused in CVSGM can be resolved by transmitting only the $1-d$ signed distance value instead of sites local vectors, while for SJ monitoring in Figure 16(b) almost each and every FP decision is efficiently resolved using our unidimensional mapping across the given δ values (hence the corresponding bar is nearly empty). We again stress that CVSGM 1-d Res does not represent additional FP decisions regarding CVSGM, but the portion of the total FPs (accumulated in the respective stacked bar) that were resolved by transmitting a single value per site.

Let us now study the impact of both our proposed unidimensional mapping and the reduced approximation error in terms of the bytes communicated throughout the tracking process. For χ^2 monitoring, Figure 15(c) illustrates

that for CVSGM, our proposed mapping mechanism can boost the almost 2 times average savings due to fewer FPs (Figure 15(b)) to up to a factor of 4.3 (for $\delta = 0.1$) in terms of communicated bytes compared to SGM. More interestingly, in the case of SJ monitoring where, according to Figure 16(b), CVSGM reduces FPs by the more moderate 1/3 ratio, savings in terms of communicated bytes reach a factor of 5 due to the fact that almost all of the FPs can be resolved using a single signed distance value per site. It can easily be deduced that for tracking tasks of higher dimensionality the savings provided by our novel mapping scheme of Section 4.1 would be even larger. We leave experimentation with monitoring tasks of higher dimensionality to future work.

With respect to L_∞ monitoring, for which we concentrate on FNs where our unidimensional mapping alone does not have an impact on the transmitted bytes, Figure 17(a) shows that CVSGM exhibits an increased number of transmitted messages compared to SGM. However, as Figure 17(b) demonstrates, CVSGM considerably decreases the amount of FN synchronization decisions for all the tested δ values. For instance, for $N = 500$ and $\delta = 0.1$ FNs are reduced from 112 in SGM to 32 in CVSGM, i.e., by a factor of 3.5, which is equivalent to >40K additional messages devoted to meaningful centralization decisions. Therefore, increasing the number of communicated messages is not only acceptable but also desired in this case due to true threshold crossings. Moreover, in Figure 17(b) our CVSGM approach can reduce the amount of FNs up to a factor of 6.2 which is an improvement solely attributed to the ability of CVSGM to reduce the approximation error of the monitored (unidimensional) estimator.

7. Analysis of Sum-parameterized Functions

At the very beginning of the discussion in Section 1.2, we argued that the scalability issues that arise when monitoring sum-parameterized functions are much more pronounced compared to the average case. There, we stressed that this is because, apart from having more sites contributing vectors to form the convex hull (as happens with average input), sum-parameterization requires all site drift vectors to be scaled proportionally to N. Hence, the size of the convex hull and of the covering spheres increases with the network scale, making it much more prone to false positives.

In this section, we provide the details for monitoring sum-parameterized functions, and study the impact of sum-parameterization from different perspectives as well as in conjunction with our sampling-based approach. First, our study focuses on the effect of sum-parameterization on the tracked input domain and then on the function range, for different function categories. Finally, we prove that monitoring sum-parameterized functions can be transformed to an equivalent problem of monitoring average-parameterized functions, which avoids scaling the drift vectors by N. We then provide a theoretical analysis of the frequency of FP alerts.

7.1. Impact on Monitored Input Domain - The Adapted Drift Vectors Approach

Consider the case where the function that needs to be monitored is parameterized with the sum rather than the global average, i.e., $f(v_{sum}(t)) \lessgtr T$ with $v_{sum}(t) = N \cdot v(t) = \sum_{i=1}^{N} v_i(t)$. Such functions, for instance, include L_p norms during approximate function monitoring queries [12, 19].

The proposed approach, henceforth termed the Adapted Vectors approach, focuses on adapting the drift vectors $(e(t) + \Delta v_i(t))$ used in the initial framework of Section 1.1 so that they can bound a volume of the input domain where $v_{sum}(t)$ should lie. Towards this end, notice that $v_{sum}(t)$ can be expressed as ($e_{sum}(t) = v_{sum}(t_s)$ stands for the global sum at the synchronization time t_s):

$$v_{sum}(t) = \sum_{i=1}^{N} v_i(t) = \sum_{i=1}^{N} \frac{N \cdot v_i(t)}{N} = e_{sum}(t) + \sum_{i=1}^{N} \frac{N \cdot \Delta v_i(t)}{N}$$

The above equation shows that $v_{sum}(t)$ can be expressed as a convex combination of the $(e(t) + \Delta v_i(t))$ vectors scaled by N since $N \cdot (e(t) + \Delta v_i(t)) = e_{sum}(t) + N \cdot \Delta v_i(t)$. As a consequence $v_{sum}(t) \in Conv(e_{sum}(t) + N \cdot \Delta v_1(t), \ldots, e_{sum}(t) + N \cdot \Delta v_N(t))$. In order to monitor the newly constructed convex hull in a distributed manner we also need to scale the sites' local constraints to ensure they cover it:

$$Conv(e_{sum}(t) + N \cdot \Delta v_1(t), \ldots, e_{sum}(t) + N \cdot \Delta v_N(t)) \subset \bigcup_{i=1}^{N} B(e_{sum}(t) + \frac{N}{2} \Delta v_i(t), \frac{N}{2} \|\Delta v_i(t)\|)$$

Hence, the local constraint in each site S_i is $B(e_{sum}(t) + \frac{N}{2}\Delta v_i(t), \frac{N}{2}\|\Delta v_i(t)\|)$, which is checked to judge whether it invades the threshold surface. Obviously, the size of the monitored portion of the input domain and thus the possibility of FP synchronizations increases with N.

7.2. Impact on Function Range

Section 7.1 concentrated on studying the effect of sum-parameterization on the (size of the) monitored input domain. Here, we focus on the range of the tracked function. Note that since $v_{sum}(t) = \sum_{i=1}^{N} v_i(t) = N \cdot v(t)$, we actually have $f(v(t))$ and $f(v_{sum}(t)) = f(N \cdot v(t))$ versions of $f(\cdot)$, which share the same multivariate (vector) input $v(t)$ at a different scale. We now wish to study the effect of the above linear scaling of the input on the function's range. Our aim is to obtain useful conclusions with respect to how the two versions of $f(\cdot)$ change and thus approach or draw away from a given threshold T. For instance, if the two versions of the function are asymptotically equal, then their values will be almost identical for large variable values, and they will simultaneously surpass any given T. On the other hand, if $f(N \cdot v(t))$ asymptotically dominates $f(v(t))$, $f(N \cdot v(t))$ will most likely cross a given threshold before $f(v(t))$ does.

To extract general conclusions with respect to the above questions, extended asymptotic analysis [44] to multivariate scalar functions can be used. We study the Relative Rate of Growth, $RRG = \lim\limits_{\|v(t)\|_\infty \to \infty} \left| \frac{f(N \cdot v(t))}{f(v(t))} \right|$, with $\|v(t)\|_\infty$ denoting the L_∞ norm. We analyze (not necessarily disjoint) function categories that include the majority of functions used in GM related work, and show that most of the times we can derive a tight bound. In the rest of this subsection we omit the temporal reference t.

Homogeneous Functions of constant degree α have the property: $f(v_{sum}) = N^\alpha \cdot f(v)$. Hence, $RRG = N^\alpha$. When $\alpha = 0$, the two versions of the function are actually (not only asymptotically) equal. This is indeed the case for many functions that have been used in GM-related work, including, χ^2 score [21, 18], cosine similarity [13] or correlation coefficient [21]. Additionally, general L_p norms [19, 12] are homogeneous of degree 1. Linear growth also holds for divergence measures such as Kullback-Leibler and Jeffrey Divergence.

Polynomial Functions are very popular, for example $v_{sum} = [x, y]$, $f(v_{sum}) = 2 \cdot x^2 + 4 \cdot x \cdot y + y^2 - 7$. In this case, $f(v_{sum}) = f_1(v) \cdot N^\alpha$, with α denoting the degree of the polynomial. The function $f_1(\cdot)$ is derived by dividing $f(v_{sum})$ by N^α, and if the polynomial is homogeneous it equals $f(v)$. In our initial example where the polynomial is quadratic [39], i.e., $\alpha = 2$, $f_1(v) = 2 \cdot \frac{x^2}{N^2} + 4 \cdot \frac{x \cdot y}{N^2} + \frac{y^2}{N^2} - \frac{7}{N^2}$. In any case, $f(\cdot), f_1(\cdot)$ will be of the same degree and $RRG = N^\alpha$.

Rational Functions constitute quotients of polynomial functions. Therefore, they behave similarly to polynomials with α being determined by subtracting the degrees of the numerator and denominator polynomial.

Composite Functions combining the above categories with:

Logarithmic Functions of the form, $f(v_{sum}) = log_\beta(g(v_{sum}))$. A logarithmic function $f(\cdot)$ of base β with a rational, polynomial or general homogeneous input $g(\cdot)$ yields $f(v_{sum}) = f_1(v) - \alpha \cdot log_\beta(N)$, where α is defined as described in the respective function category. Mutual Information used in our running example is an instance of a logarithmic function with a rational parameter, where $\alpha = 1$. Hence, the factor N in the input has an additive effect on the function value. However, the two versions of the function are again asymptotically equal since, $RRG = 1$.

Exponential Functions of the form $f(v_{sum}) = \beta^{g(v_{sum})}$, $\beta \neq 0$, often exhibit the rate at which a tracked quantity decays or grows [6]. The value of RRG depends on $g(v_{sum})$. If it is a polynomial, then $RRG = \infty$ which is equivalent to $f(v)$ being dominated by $f(v_{sum})$. For a rational $g(v_{sum})$, $\alpha > 0$ yields $RRG = \infty$ (dominance) and 1 (asymptotic equality) otherwise.

7.3. A Function Transformation Approach

We now generalize our discussion of Section 7.2 to show how the monitoring task of a sum-parameterized function, i.e. $f(v_{sum}(t)) = f(v(t) \cdot N) \lessgtr T$, can be transformed to an equivalent average-parameterized problem. The latter avoids scaling the drift vectors of GM by N, in principle offering an alternative solution (instead of the Adapted Vectors approach of Section 7.1). We term the alternative approach proposed here as the Function Transformation approach, and the question is then whether it encounters the additional scalability issues of sum-parameterization, even after the scaling by a factor of N is no longer performed. We answer this question in the negative, by formally proving

that *irrespectively of the monitored function, threshold and transformation operators*, the GM-based monitoring task of the Adapted Vectors and the Function Transformation approaches yield equivalent tracking schemes.

The Function Transformation Approach is based on the simple observation we make in Section 7.2 that $v(t) = \frac{v_{sum}(t)}{N}$. Hence, the monitoring task involving the sum-parameterized function is equivalent to $f(v(t) \cdot N) \lessgtr T$. In Section 7.2 we showed that for a wide range of function categories, $f(v(t) \cdot N)$ can be decomposed to a function of the global average $f_1(v(t))$ and a function of N. For instance, for homogeneous, polynomial or rational functions it holds that $f(v_{sum}(t)) = f_1(v(t)) \cdot N^{\alpha}$. Generalizing the previous discussion, $f(v(t) \cdot N)$ can be decomposed into two distinct functions $f(v(t) \cdot N) = f_1(v(t)) \circ f_2(N)$, where $\circ$ denotes an operation between functions from the set $\{+, -, \cdot, \div\}$. Let $\bullet$ denote the inverse operation of $\circ$ (i.e. from $+$ to $-$, $\cdot$ to $\div$ or vice versa); then the tracked function can be transformed to one that is parameterized by the average vector $v(t)$ (T, N are assumed a priori known):

$$f(v_{sum}(t)) \lessgtr T \Leftrightarrow f_1(v(t)) \lessgtr T \bullet f_2(N) \tag{10}$$

and the original geometric monitoring framework can be utilized to accomplish the task as described in Section 1.1. Interestingly, the transformed monitoring task does not require scaling the drift vectors of the sites by N.

The following lemma compares the threshold surfaces of the two methods:

Lemma 6. *Let $C_{v_{sum}} = \{r_{v_{sum}} \in R^d : f(r_{v_{sum}}) = T\}$ denote the threshold surface of the monitoring task $f(v_{sum}(t)) \lessgtr T$ and, for $v(t) = \frac{v_{sum}(t)}{N}$, $C_v = \{r_v \in R^d : f_1(r_v) = T \bullet f_2(N)\}$ the threshold surface defined by the function transformation according to Equivalence 10. Then, irrespectively of $f(\cdot)$, $f_1(\cdot)$, $f_2(\cdot)$:*
a) There is a bijection that maps every $r_{v_{sum}} \in C_{v_{sum}}$ to exactly one $r'_{v_{sum}} \in C_v$.
b) Given (a), every possible distance between $v(t)$ and C_v in the transformed setting is N times shorter than the respective $v_{sum}(t)$ and $C_{v_{sum}}$ distance in the original, sum-parameterized task.

Proof. a) Notice that since $f(r_{v_{sum}}) = T$, then, according to Equivalence 10, $f_1(\frac{r_{v_{sum}}}{N}) = T \bullet f_2(N)$ and thus $\forall r_{v_{sum}} \in C_{v_{sum}}$, $r'_{v_{sum}} = \frac{r_{v_{sum}}}{N} \in C_v$. Similarly, $\forall r_v \in C_v$, $N \cdot r_v \in C_{v_{sum}}$. Hence, there is a bijection that pairs every vector in $C_{v_{sum}}$ with exactly one vector in C_v and vice versa.
b) Let us now examine the ratio of the corresponding distances. Simple calculations show that $\forall r_{v_{sum}} \in C_{v_{sum}}$, $\frac{\|v_{sum}(t) - r_{v_{sum}}\|}{\|v(t) - \frac{r_{v_{sum}}}{N}\|} = \frac{\|v_{sum}(t) - r_{v_{sum}}\|}{\|\frac{v_{sum}(t)}{N} - \frac{r_{v_{sum}}}{N}\|} = N$ and $r'_{v_{sum}} = \frac{r_{v_{sum}}}{N} \in C_v$, as shown in (a). This completes the proof. $\square$

This result holds independently of the distributed protocol (GM or any other) used to execute the monitoring process. The next lemma exploits Lemma 6 and shows that applying GM using the Function Transformation does not amend the scalability issues of the Adapted Vectors approach. More precisely, it proves that *irrespectively* of the category of the monitored function (Section 7.2), the intersection of the union of balls with the threshold surface in the Function Transformation scheme is isomorphic (denoted by $\cong$) to (i.e. there is an invertible linear transformation among the vectors of) the respective intersection in the Adapted Vectors scheme. Further, this linear transformation is an isometry (scales every vector by a factor of N).

Lemma 7. *The Function Transformation and the Adapted Vectors approach result in equivalent GM monitoring schemes, i.e.:* $\bigcup_{i=1}^{N} B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|) \cap C_v \cong \bigcup_{i=1}^{N} B(e_{sum}(t) + \frac{N}{2}\Delta v_i(t), \frac{N}{2}\|\Delta v_i(t)\|) \cap C_{v_{sum}}$.

Proof. Sketch. In the proof of Lemma 6 we showed that there is an invertible linear transformation among the vectors belonging to each threshold surface, i.e., $\forall r_{v_{sum}} \in C_{v_{sum}}$, $\frac{r_{v_{sum}}}{N} \in C_v$ and $\forall r_v \in C_v$, $N \cdot r_v \in C_{v_{sum}}$. To conclude the proof, just consider the subset of these vectors that also belong to the intersection with the respective unions of balls (if non-empty). $\square$

Extending the above to the convex safe zone in Section 4 requires noting that (a) the CV scheme would simply check $e(t) + \Delta v_i(t)$ and $e_{sum}(t) + N \cdot \Delta v_i(t)$ correspondingly, among which there is trivially an invertible linear transformation, and (b) the threshold surface in CV is a special case of $C_{v_{sum}}, C_v$ being convex.

7.4. Effect of Sum-parameterization and Sampling-based Monitoring Performance in Practice

Since we just showed that the scalability issues are much more pronounced for sum-parameterized functions and non-amendable even by function transformation, the usefulness (in terms of communication saving) of our sampling-based approaches is expected to increase in this case. To further elucidate and practically study the effect of sum-parameterization, we perform the following test: we track the same function - standard deviation (*stdev* - square root of variance) - once parameterized by the sum ($stdev(v_{sum}(t))$) and once by the average of sites' local vectors ($stdev(v(t))$), using the Jester data. Note that in this test $stdev(v(t))$ is different than the Function Transformation approach, since to study the pure effect of sum versus average input we do not apply $T \bullet f_2(N)$. Function Transformation has been proven equivalent to the Adapted Vectors approach which is represented by $stdev(v_{sum}(t))$ performance.

With respect to the function value (Section 7.2), $stdev(v_{sum}(t))$ scales linearly with $N \cdot v$, but the results are analogous for the variance that scales quadratically. Thus, our study accounts for linear to super-linear growth with sum input. Having parameterized the function by the sum and average input respectively, we choose two different thresholds, both of which are never actually crossed by any version of the tracked function, so the monitoring in both versions of *stdev* may only produce false positives. This choice is made in order to isolate the effect of sum-parameterization which exacerbates false positives compared to the average case. Since parameterizing *stdev* by the average entails a factor $1/N$ in the calculation of *stdev*, it receives lower (closer to zero) values for the average parameterization case, while it is proportionately higher upon parameterizing the function by the sum of local vectors. Therefore, we choose one threshold close to zero (0.1), and thus to the actual *stdev* upon parameterization by the average, termed "lower T", and one threshold close to 100, termed "upper T" that was close to the *stdev* value upon parameterizing by the sum of local vectors.

The graph in Figure 18 measures the GM/SGM gain ratio over different network scales (N). We make the following two observations:

• In the case of "lower T": comparing "AVG lower T" against "SUM lower T" shows that the ratio GM/SGM of the latter is an order of magnitude (or more) higher than the respective ratio of the former. This holds despite the fact that the "lower T" threshold is far away from the actual values *stdev* receives upon parameterization by the sum of local vectors. The reason for the "SUM lower T" behavior is twofold: a) GM causes many more FPs compared to "AVG lower T" since the drift vectors are scaled by N, b) these FPs are better amended by using our sampling-based approach, as shown by the gains (ratio GM/SGM) which are proportionately higher in "SUM lower T" compared to the "AVG lower T" case. The "SUM lower T" remains stable across network scales (the small decline is due to the fact that as sites are added they contribute their local values to the global sum which changes its position as well) for this case where the threshold is far from the actual values the function receives.

• In case of "upper T": studying "AVG upper T" demonstrates that if we place the threshold extremely far from the actual (close to zero) value of the average-parameterized version of *stdev*, the respective GM/SGM receives its lowest values. This is in contrast to the "SUM lower T" behavior and is attributed to the fact that "AVG upper T" does not require scaling local vectors by N. On the other hand, despite the fact that the sum-parameterized version of *stdev* never truly crosses the threshold, because "upper T" is near the values it receives throughout the tracking process, the GM/SGM ratio for "SUM upper T" increases with the number of sites. In fact, doubling the network scale from 500 to 1000 sites causes over an order of magnitude increase in the GM/SGM ratio, which further demonstrates the advantage of the proposed sampling-based approach.

8. Conclusions

In this work we rendered the GM framework, introduced in [5], capable of operating in highly distributed settings. We initially studied the culprits that cause the GM approach and its variants to become impractical due to severe scalability issues. To counter these issues, we introduced novel sampling-based GM techniques capable of performing the tracking process utilizing only a sample of the available sites. The sample size entailed by our methods is proportional to $\sqrt{N}$ and also depends on the application's accuracy requirements. Our experimental evaluation using a wide variety of functions, network scales, thresholds and related work comparisons shows that our sampling-based techniques can significantly reduce the communication cost throughout the monitoring process while abiding by controllable accuracy guarantees, outperforming the other competitors proposed in the literature. In that, we managed to apply the geometric monitoring concepts on much higher network scales, far beyond what previous related techniques had achieved.

Acknowledgment

References

[1] G. Cormode, The continuous distributed monitoring model, SIGMOD Rec. 42 (1) (2013) 5–14.

[2] G. Cormode, M. Garofalakis, Streaming in a connected world: querying and tracking distributed data streams, in: Proc. of SIGMOD Conference, 2007, pp. 1178–1181.

[3] S. R. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong, Tinydb: an acquisitional query processing system for sensor networks, ACM Trans. Database Syst. 30 (2005) 122–173.

[4] G. Cormode, M. Garofalakis, S. Muthukrishnan, R. Rastogi, Holistic aggregates in a networked world: distributed tracking of approximate quantiles, in: SIGMOD, 2005, pp. 25–36.

[5] I. Sharfman, A. Schuster, D. Keren, A geometric approach to monitoring threshold functions over distributed data streams, in: SIGMOD, 2006, pp. 301–312.

[6] G. Cormode, M. Garofalakis, Approximate continuous querying over distributed streams, ACM Transactions on Database Systems 33 (2) (2008) 9:1–9:39.

[7] G. Cormode, S. Muthukrishnan, K. Yi, Algorithms for distributed functional monitoring, ACM Trans. Algorithms 7 (2) (2011) 21:1–21:20.

[8] M. Tang, F. Li, Y. Tao, Distributed online tracking, in: Proc. of SIGMOD Conference, New York, NY, USA, 2015, pp. 2047–2061.

[9] M. Dilman, D. Raz, Efficient reactive monitoring, in: INFOCOM, Vol. 2, 2001, pp. 1012–1019.

[10] R. Keralapura, G. Cormode, J. Ramamirtham, Communication-efficient distributed monitoring of thresholded counts, in: SIGMOD, 2006, pp. 289–300.

[11] A. Manjhi, V. Shkapenyuk, K. Dhamdhere, C. Olston, Finding (recently) frequent items in distributed data streams, in: ICDE, 2005, pp. 767–778.

[12] M. Garofalakis, D. Keren, V. Samoladas, Sketch-based geometric monitoring of distributed stream queries, In VLDB 6 (10) (2013) 937–948.

[13] S. Burdakis, A. Deligiannakis, Detecting outliers in sensor networks using the geometric approach, in: Proc. of ICDE Conference, 2012, pp. 1108–1119.

[14] A. Lazerson, I. Sharfman, D. Keren, A. Schuster, M. Garofalakis, V. Samoladas, Monitoring distributed streams using convex decompositions, Proc. VLDB Endow. 8 (5) (2015) 545–556.

[15] O. Papapetrou, M. N. Garofalakis, Continuous fragmented skylines over distributed streams, in: IEEE ICDE, 2014, pp. 124–135.

[16] M. Gabel, A. Schuster, D. Keren, Communication-efficient distributed variance monitoring and outlier detection for multivariate time series, in: IEEE IPDPS, 2014, pp. 37–47.

[17] M. Kamp, M. Boley, D. Keren, A. Schuster, I. Sharfman, Communicationefficient distributed online prediction by dynamic model synchronization, in: ECML PKDD, 2014, pp. 623–639.

[18] N. Giatrakos, A. Deligiannakis, M. Garofalakis, I. Sharfman, A. Schuster, Prediction-based geometric monitoring over distributed data streams, in: SIGMOD, 2012, pp. 265–276.

[19] N. Giatrakos, A. Deligiannakis, M. Garofalakis, I. Sharfman, A. Schuster, Distributed geometric query monitoring using prediction models, ACM Trans. Database Syst. 39 (2) (2014) 16:1–16:42.

[20] G. Cormode, M. Garofalakis, Sketching streams through the net: Distributed approximate query tracking, in: Proc. of VLDB Conference, 2005, pp. 312–323.

[21] I. Sharfman, A. Schuster, D. Keren, Shape sensitive geometric monitoring, in: PODS, 2008, pp. 301–310.

[22] Y. Yang, J. O. Pedersen, A comparative study on feature selection in text categorization, in: ICML, 1997, pp. 412–420.

[23] G. Cormode, N. Duffield, Sampling for big data: A tutorial, in: Proc. of ACM SIGKDD Conference, 2014, pp. 1975–1975.

[24] C.-E. Särndal, B. Swensson, J. Wretman, "Model Assisted Survey Sampling", Springer-Verlag New York, Inc. (Springer Series in Statistics), 1992.

[25] P. J. H. Graham Cormode, Minos Garofalakis, C. Jermaine, Synopses for massive data: Samples, histograms, wavelets, sketches, Foundations and Trends in Databases 4 (13) (2011) 1–294.

[26] M. Garofalakis, J. Gehrke, R. Rastogi, Querying and mining data streams: You only get one look a tutorial, in: Proc. of SIGMOD Conference, 2002, pp. 635–635.

[27] D. Keren, I. Sharfman, A. Schuster, A. Livne, Shape sensitive geometric monitoring, IEEE Trans. on Knowl. and Data Eng. 24 (8) (2012) 1520–1535.

[28] A. Arasu, B. Babcock, S. Babu, M. Cieslewicz, Johnand Datar, K. Ito, R. Motwani, U. Srivastava, J. Widom, Stream: The stanford data stream management system, in: M. Garofalakis, J. Gehrke, R. Rastogi (Eds.), Data Stream Management: Processing High-Speed Data Streams, Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 317–336.

[29] D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, S. Zdonik, Monitoring streams: a new class of data management applications, in: VLDB, 2002, pp. 215–226.

[30] S. R. Jeffery, M. Garofalakis, M. J. Franklin, Adaptive cleaning for rfid data streams, in: Proceedings of the 32Nd International Conference on Very Large Data Bases, VLDB '06, VLDB Endowment, 2006, pp. 163–174.

[31] E. J. Candès, Y. Plan, A probabilistic and ripless theory of compressed sensing, CoRR abs/1011.3854.
URL http://arxiv.org/abs/1011.3854

[32] Q. G. Zhao, M. Ogihara, H. Wang, J. J. Xu, Finding global icebergs over distributed data sets, in: PODS, 2006, pp. 298–307.

[33] Z. Huang, K. Yi, Y. Liu, G. Chen, Optimal sampling algorithms for frequency estimation in distributed data, in: INFOCOM, 2011, pp. 1997–2005.

[34] Z. Huang, K. Yi, Q. Zhang, Randomized algorithms for tracking distributed count, frequencies, and ranks, in: Proc. of PODS, 2012, pp. 295–306.

[35] N. Giatrakos, A. Deligiannakis, M. Garofalakis, Scalable approximate query tracking over highly distributed data streams, in: Proceedings of the 2016 International Conference on Management of Data, SIGMOD '16, ACM, New York, NY, USA, 2016, pp. 1497–1512.

[36] C. McDiarmid, On the method of bounded differences, Cambridge University Press, Cambridge, 1989.

[37] G. Sagy, D. Keren, I. Sharfman, A. Schuster, Distributed threshold querying of general functions by a difference of monotonic representation, In VLDB 4 (2010) 46–57.

[38] G. Sagy, I. Sharfman, D. Keren, A. Schuster, Top-k vectorial aggregation queries in a distributed environment, J. Parallel Distrib. Comput. 71 (2) (2011) 302–315.

[39] D. Keren, G. Sagy, A. Abboud, D. Ben-David, A. Schuster, I. Sharfman, A. Deligiannakis, Geometric monitoring of heterogeneous streams, Knowledge and Data Engineering, IEEE Transactions on 26 (8) (2014) 1890–1903.

[40] Z. Liu, B. Radunović, M. Vojnović, Continuous distributed counting for non-monotonic streams, in: Proc. of PODS, 2012, pp. 307–318.

[41] D. D. Lewis, Y. Yang, T. G. Rose, F. Li, Rcv1: A new benchmark collection for text categorization research, Journal of Machine Learning Research 5 (Apr) (2004) 361–397.

[42] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: A constant time collaborative filtering algorithm, Inf. Retr. 4 (2) (2001) 133–151.

[43] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vision 40 (2) (2000) 99–121.

[44] R. R. Howell, On asymptotic notation with multiple variables, Dept. of Computing and Information Sciences, Kansas State University, Tech. Rep.

Appendix A. Proofs of Lemma 1, Lemma 2 and Lemma 5

Lemma 1. *For Estimator 1 the following hold:*
(a) Estimator 1 is an unbiased estimator of v when sampling $\forall S_i \in \{S_1, \ldots, S_N\}$ with $0 \le g_i \le 1$.
(b) $E[\hat{v}] \in Conv(\, e + \Delta v_1, \ldots, e + \Delta v_N)$
(c) $\hat{v} \in Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$

Proof.

(a) To prove that the estimator is unbiased we need to show that $E[\hat{v}] = v$. Recall from Equation 1 that $\sum_{S_i \in K} \frac{\Delta v_i}{g_i} = \sum_{i=1}^{N} \Delta' v_i$ and that $E[\Delta' v_i] = g_i \cdot \frac{\Delta v_i}{g_i} + (1 - g_i) \cdot 0 = \Delta v_i$. By applying the properties of the expected value we get:

$E[\hat{v}] = E[e + \frac{\sum_{i=1}^{N} \Delta' v_i}{N}] = e + \frac{\sum_{i=1}^{N} E[\Delta' v_i]}{N} = e + \frac{\sum_{i=1}^{N} \Delta v_i}{N} = v$

(b) Obvious, since $E[\hat{v}] = v$ and $v \in Conv(\, e + \Delta v_1, \ldots, e + \Delta v_N)$.

(c) $\hat{v}$ is a convex combination of the $e + \Delta' v_i$ vectors and, therefore, lies in their convex hull. Since $\Delta' v_i = 0$ for all sites not included in the sample, the convex hulls $Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$ and $Conv(\{e + \Delta' v_i\} : \forall i \in [1, N])$ coincide. $\square$

Lemma 2. *(a) For a single sampling trial with sample of cardinality $|K|$:*
$\hat{v} \in Conv(\{e + \frac{|K|}{N \cdot g_i} \Delta v_i\} : \forall S_i \in K) \Rightarrow \hat{v} \in \bigcup_{S_i \in K} B(e + \frac{|K|}{N \cdot g_i} \frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i} \| \frac{\Delta v_i}{2} \|) \Rightarrow \exists S_i \in K : \hat{v} \in B(e + \frac{|K|}{N \cdot g_i} \frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i} \| \frac{\Delta v_i}{2} \|)$

(b) On expectation $\frac{|K|}{N \cdot g_i} \le 1 + \frac{1}{N} \approx 1$, $\forall S_i \in K$.

(c) Assume each site $S_i \in \{S_1, \ldots, S_N\}$ performs $1 \le M \le \left\lceil \frac{log(0.01)}{log(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N})} \right\rceil$ independent (among sites and among tri-
als) sampling trials, using its own sampling function g_i. Further assume that g_is are chosen so that in each trial the
expected sample size is bounded by $\ell n(1/\delta) \sqrt{N}$, i.e., $E[|K_\mu|] = \sum_{i=1}^{N} g_i \le \ell n(1/\delta) \sqrt{N}$, $\forall \mu \in [1, M]$. Then, with 0.99
probability, there will be at least one trial that includes a version $\hat{v}_\mu$ of Estimator 1 in the GM-spheres (i.e, not scaled
by $1/g_i$) of S_is $\in K_\mu$:

$$P\left(\nexists \hat{v}_\mu \in \{\hat{v}_1, \ldots, \hat{v}_M\} : \hat{v}_\mu \in \bigcup_{S_i \in K_\mu} B(e + \frac{\Delta v_i}{2}, \| \frac{\Delta v_i}{2} \|) \right) \le 0.01$$

Proof. (a) For ease of exposition, and without loss of generality assume that $e = 0$. For any given sample K, from Estimator 1 we have:

$$\hat{v} = \sum_{S_i \in K} \frac{\Delta v_i}{N \cdot g_i} = \frac{\sum_{S_i \in K} \frac{|K|}{N \cdot g_i} \Delta v_i}{|K|} \Rightarrow \hat{v} \in Conv(\{\frac{|K|}{N \cdot g_i} \Delta v_i\}, \forall S_i \in K)$$

This shows that $\hat{v}$ can be expressed as a convex combination (average) of the K sampled sites and, having obtained the sample, it will be included in $Conv(\{\frac{|K|}{N \cdot g_i} \Delta v_i\}, \forall S_i \in K)$. Then:

$$Conv(\{\frac{|K|}{N \cdot g_i} \Delta v_i\}, \forall S_i \in K) \subseteq \bigcup_{S_i \in K} B(\frac{|K|}{N \cdot g_i} \frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i} \| \frac{\Delta v_i}{2} \|) \Rightarrow \exists S_i \in K : \hat{v} \in B(\frac{|K|}{N \cdot g_i} \frac{\Delta v_i}{2}, \frac{|K|}{N \cdot g_i} \| \frac{\Delta v_i}{2} \|)$$

(b) The above coefficients $\frac{|K|}{N \cdot g_i}$ are produced based on a sampling process where each site independently decides to include itself in K or not, based on its own sampling function g_i. Therefore, each such coefficient is produced as a result of a pair of dependent random variables: the (global) sample size which incorporates the decisions of all sites and the individual decision of a site with respect to $S_i \in K$.

Let x_i denote a random variable that receives the value of 1 if $S_i \in K$ with probability g_i and 0 otherwise. Let X denote the sum of x_is, i.e., $X = \sum_{i=1}^{N} x_i$. Obviously, since $x_i = 0$ for $S_i \notin K$, $X = \sum_{i=1}^{N} x_i = \sum_{S_i \in K} 1 = |K|$. Furthermore, y_i

43

denotes a random variable that receives the value of $\frac{1}{N \cdot g_i}$ if $S_i \in K$ with probability g_i and 0 otherwise. Then $E[X \cdot y_i]$ expresses the expectation of $\frac{|K|}{N \cdot g_i}$ and since the two variables are correlated, the expectation of their product engages their covariance: $E[X \cdot y_i] = E[X] \cdot E[y_i] + Cov(X, y_i)$. Regarding $Cov(X, y_i)$ we have:

$$Cov(X, y_i) = E[(X - E[X])(y_i - E[y_i])] = (X - E[X])(\frac{1}{N \cdot g_i} - \frac{1}{N})g_i + (X - 1 - E[X])(0 - \frac{1}{N})(1 - g_i) =$$

$$(X - E[X])(\frac{1}{N} - \frac{g_i}{N}) + (X - 1 - E[X])(\frac{g_i}{N} - \frac{1}{N}) = (\frac{1}{N} - \frac{g_i}{N})(X - E[X] - X + 1 + E[X]) = (\frac{1}{N} - \frac{g_i}{N}) \overset{g_i \leq 1}{\leq} \frac{1}{N}$$

Because $|K| \leq N$ and for high N the term $\frac{1}{N}$ is negligible:

$$E[X \cdot y_i] = E[X] \cdot E[y_i] + Cov(X, y_i) \leq E[|K|] \cdot \frac{1}{N \cdot g_i} \cdot g_i + \frac{1}{N} = \frac{E[|K|]}{N} + \frac{1}{N} \leq 1 + \frac{1}{N} \approx 1$$

(c) Due to Lemma 2(a) and again assuming $e = 0$ for simplicity, we know that if each $S_i \in \{S_1, \ldots, S_N\}$ performs one sampling trial based on g_i, there will be at least one site S_λ for which $\hat{v} \in B(\frac{|K|}{N \cdot g_\lambda} \frac{\Delta v_\lambda}{2}, \frac{|K|}{N \cdot g_\lambda} \| \frac{\Delta v_\lambda}{2} \|)$. The question is what is the probability that the $\frac{|K|}{N \cdot g_\lambda}$ coefficient utilized in the ball of S_λ, which on expectation is at most $1 + \frac{1}{N}$ (Lemma 2(b)), happens to be > 1 and thus simply inscribing $B(\frac{\Delta v_\lambda}{2}, \| \frac{\Delta v_\lambda}{2} \|)$ in site S_λ may (if S_λ is the only site whose $B(\frac{|K|}{N \cdot g_\lambda} \frac{\Delta v_\lambda}{2}, \frac{|K|}{N \cdot g_\lambda} \| \frac{\Delta v_\lambda}{2} \|)$ includes $\hat{v}$) leave $\hat{v}$ uncovered by the area of the input domain tracked by the sample. Using the same notation as in Lemma 2(b), we essentially seek to quantify the probability of $X \cdot y_\lambda > 1$. Since we only assume primitive knowledge of an upper bound on $E[|K|] \leq \ell n(1/\delta) \sqrt{N}$, fostering Markov's inequality:

$$P(X \cdot y_\lambda > 1) \leq P(X \cdot y_\lambda \geq 1) \leq E[X \cdot y_\lambda] \leq \frac{E[|K|]}{N} + \frac{1}{N} \leq \frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N}$$

The above bounds the probability of failing to monitor $\hat{v}$ in case sites perform a single sampling trial. If we repeat this process in M sampling trials, then M (not necessarily disjoint) samples will be formed, each monitoring a version of Estimator 1, i.e., $\hat{v}_\mu \in \{\hat{v}_1, \ldots, \hat{v}_M\}$ and in each trial there will be at least one site (not necessarily the same S_λ) for which $\hat{v}_\mu \in B(\frac{|K|}{N \cdot g_\lambda} \frac{\Delta v_\lambda}{2}, \frac{|K|}{N \cdot g_\lambda} \| \frac{\Delta v_\lambda}{2} \|)$ and $P(X \cdot y_\lambda > 1) \leq \frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N}$. Therefore, the probability of failing to monitor at least one $\hat{v}_\mu$ is at most $\left(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N} \right)^M$. Should we bound this probability by 0.01 and solving for M we finally get:

$$M \geq \frac{log(0.01)}{log\left(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N} \right)}$$

so it suffices to pick $M = \left\lceil \frac{log(0.01)}{log\left(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N} \right)} \right\rceil$, but since $\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N}$ is an upper bound on the tracking failure probability of a single trial (i.e., the actual failure probability is smaller or equal), therefore the statement of the lemma, $M \leq \left\lceil \frac{log(0.01)}{log\left(\frac{\ell n(1/\delta)}{\sqrt{N}} + \frac{1}{N} \right)} \right\rceil$. $\qquad\square$

Lemma 5. *The revised Sampling-Based GM Scheme in the convex safe zone (CV) context under the mapping of Lemma 4 yields:*

- $P_{FN} \leq \delta + 0.01$ *if M trials $\bigwedge \forall S_i \in \{S_1, \ldots, S_N\}, \frac{|d_C(e + \Delta v_i)|}{\sqrt{2}} > \varepsilon_C$*
- $P_{FN} = O(\delta^{\frac{|Z|M}{\sqrt{N}}})$ *otherwise*

where $1 \leq M \leq \left\lceil \frac{log(0.01)}{log\left(e^{-0.042 \sqrt{\ell n(1/\delta)N}} \right)} \right\rceil$ and Z (more precisely $Z(t)$) denotes the set of threshold crossing sites. Thus, one can properly tune δ to obtain the desired FN probability.

Proof. We first concentrate on the first case of the lemma. Initially notice that if there exists a sampled site with $d_C(e + \Delta v_i) > 0$, that site will inform the coordinator and a synchronization phase will begin. During this phase, the scheme is always FN safe with probability $1 - \delta$ due to Inequality 8.

We then examine the case when all sampled sites have a negative signed distance, i.e., $d_C(e + \Delta v_i) < 0, \forall S_i \in K$ and thus $\hat{D}_C < 0$. According to Inequality 8, in order to be FN safe with probability $1 - \delta$ in this case, $\hat{D}_C < -\varepsilon_C$ should hold. Based on Estimator 5, g_i^C and $\varepsilon_C = \frac{U}{\sqrt{2\ell n(1/\delta)}}$:

$$\hat{D}_C = \sum_{S_i \in K} \frac{d_C(e + \Delta v_i)}{N \cdot g_i^C} < -\varepsilon_C \Leftrightarrow \sum_{S_i \in K} \frac{d_C(e + \Delta v_i)}{N \cdot \frac{|d_C(e + \Delta v_i)|\ell n(1/\delta)}{U \cdot \sqrt{N}}} < -\frac{U}{\sqrt{2\ell n(1/\delta)}} \overset{d_C(e + \Delta v_i) < 0}{\underset{\forall S_i \in K}{\Leftrightarrow}} |K| > \sqrt{\frac{\ell n(1/\delta)N}{2}}$$

This shows that $\hat{D}_C < -\varepsilon_C \Leftrightarrow |K| > \sqrt{\frac{\ell n(1/\delta)N}{2}}$. In other words, if $d_C(e + \Delta v_i) < 0, \forall S_i \in K$, $\hat{D}_C < -\varepsilon_C$ strictly depends on whether the scheme achieved sufficiently high sample size.

When $\forall S_i \in \{S_1, \ldots, S_N\}, \frac{|d_C(e + \Delta v_i)|}{\sqrt{2}} > \varepsilon_C$ as required by the lemma, $g_i^C > \sqrt{\frac{\ell n(1/\delta)}{N}}$. Moreover, by the definition of g_i^C, $|d_C(e + \Delta v_i)| < U$ and thus $g_i^C \leq \frac{\ell n(1/\delta)}{\sqrt{N}}$. For $\delta \in (0, e^{-1})$ as in Equation 4, $\ell n(1/\delta) > \sqrt{\ell n(1/\delta)}$ and overall:

$$\sqrt{\frac{\ell n(1/\delta)}{N}} < g_i^C \leq \frac{\ell n(1/\delta)}{\sqrt{N}} \Leftrightarrow \sqrt{\ell n(1/\delta)N} < E[|K|] = \sum_{i=1}^N g_i^C \leq \ell n(1/\delta)\sqrt{N}$$

We now have a lower, apart from an upper, bound on the expected sample size. Therefore, we can use the lower tail of Chernoff inequality [26] to bound the probability of not getting enough sample size. Formally, for $\alpha = 1 - \frac{1}{\sqrt{2}}$:

$$P\left(|K| \leq \sqrt{\frac{N\ell n(1/\delta)}{2}}\right) = P\left(|K| \leq (1 - (1 - \frac{1}{\sqrt{2}}))\sqrt{\ell n(1/\delta)N}\right) \leq P\left(|K| \leq (1 - (1 - \frac{1}{\sqrt{2}}))E[|K|]\right) =$$

$$P(|K| \leq (1 - \alpha)E[|K|]) \leq e^{-\frac{\alpha^2 E[|K|]}{2}} = e^{-\frac{\left(1 - \frac{1}{\sqrt{2}}\right)^2 E[|K|]}{2}} \leq e^{-0.042 E[|K|]} \leq e^{-0.042\sqrt{\ell n(1/\delta)N}}$$

This is a bound on the probability of failing to reach a sample of at least $\sqrt{\frac{N\ell n(1/\delta)}{2}}$ size after one sampling trial (per site and in the network). Should we instruct sites attempt M sampling trials, then either (a) at some trial a site with $d_C(e + \Delta v_i) > 0$ will get sampled thus causing a (initially partial) synchronization, in which case the scheme is always safe with probability $1 - \delta$ or (b) none of the sites that get sampled throughout the M trials will possess a $d_C(e + \Delta v_i) > 0$, in which case we are safe with probability $1 - \delta$ if in one trial $\hat{D}_C < -\varepsilon_C \Leftrightarrow |K| > \sqrt{\frac{\ell n(1/\delta)N}{2}}$. The probability of having failed to gather enough sample size in at least one out of M trials is given by $\left(e^{-0.042\sqrt{\ell n(1/\delta)N}}\right)^M$. Should we bound this quantity by 0.01 and solving for M we get:

$$M \geq \frac{log(0.01)}{log\left(e^{-0.042\sqrt{\ell n(1/\delta)N}}\right)}$$

Thus, it suffices to pick $M = \left\lceil \frac{log(0.01)}{log\left(e^{-0.042\sqrt{\ell n(1/\delta)N}}\right)} \right\rceil$. Since $e^{-0.042\sqrt{\ell n(1/\delta)N}}$ is an upper bound on the failure probability of a single trial, actually $M \leq \left\lceil \frac{log(0.01)}{log\left(e^{-0.042\sqrt{\ell n(1/\delta)N}}\right)} \right\rceil$ trials are needed. This concludes our proof for the first part of the lemma.

The second case of the lemma is introduced to cover situations where $\exists S_i \in \{S_1, \ldots, S_N\}, \frac{|d_C(e + \Delta v_i)|}{\sqrt{2}} \leq \varepsilon_C$. Then, P_{FN} depends on the number of threshold crossing sites. The analysis is similar to that of Sections 2.2 and 3 resulting in Lemma 3, only this time ε_T stands for the minimum distance of ∂C from the true threshold surface (a priori known and tunable based on the choice of C) and $\|\Delta v_i\|$ is replaced by $|d_C(e + \Delta v_i)|$.

$\square$