The Adversarial Robustness of Sampling

Omri Ben-Eliezer* Eylon Yogev†

Abstract

Random sampling is a fundamental primitive in modern algorithms, statistics, and machine learning, used as a generic method to obtain a small yet "representative" subset of the data. In this work, we investigate the robustness of sampling against adaptive adversarial attacks in a streaming setting: An adversary sends a stream of elements from a universe U to a sampling algorithm (e.g., Bernoulli sampling or reservoir sampling), with the goal of making the sample "very unrepresentative" of the underlying data stream. The adversary is fully adaptive in the sense that it knows the exact content of the sample at any given point along the stream, and can choose which element to send next accordingly, in an online manner.

Well-known results in the static setting indicate that if the full stream is chosen in advance (non-adaptively), then a random sample of size $\Omega(d/\varepsilon^2)$ is an ε-approximation of the full data with good probability, where d is the VC-dimension of the underlying set system $(U, \mathcal{R})$. Does this sample size suffice for robustness against an adaptive adversary? The simplistic answer is *negative*: We demonstrate a set system where a constant sample size (corresponding to a VC-dimension of 1) suffices in the static setting, yet an adaptive adversary can make the sample very unrepresentative, as long as the sample size is (strongly) sublinear in the stream length, using a simple and easy-to-implement attack.

However, this attack is "theoretical only", requiring the set system size to (essentially) be exponential in the stream length. This is not a coincidence: We show that in order to make the sampling algorithm robust against adaptive adversaries, the modification required is solely to replace the VC-dimension term d in the sample size with the cardinality term $\log|\mathcal{R}|$. That is, the Bernoulli and reservoir sampling algorithms with sample size $\Omega(\log|\mathcal{R}|/\varepsilon^2)$ output a representative sample of the stream with good probability, even in the presence of an adaptive adversary. This nearly matches the bound imposed by the attack.

*Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel.

†Department of Computer Science, Technion, Haifa, Israel. Supported by the European Union's Horizon 2020 research and innovation program under grant agreement no. 742754, and by a grant from the Israel Science Foundation (no. 950/16).

1 Introduction

Random sampling is a simple, generic, and universal method to deal with massive amounts of data across all scientific disciplines. It has wide-ranging applications in statistics, databases, networking, data mining, approximation algorithms, randomized algorithms, machine learning, and other fields (see e.g., [CJSS03, JMR05, JPA04, CDK+11, CG05, CMY11] and [Cha01, Chapter 4]). Perhaps the central reason for its wide applicability is the fact that it (provably, and with high probability) suffices to take only a small number of random samples from a large dataset in order to "represent" the dataset truthfully (the precise geometric meaning is explained later). Thus, instead of performing costly and sometimes infeasible computations on the full dataset, one can sample a small yet "representative" subset of a data, perform the required analysis on this small subset, and extrapolate (approximate) conclusions from the small subset to the entire dataset.

The analysis of sampling algorithms has mostly been studied in the *non-adaptive* (or *static*) setting, where the data is fixed in advance, and then the sampling procedure runs on the fixed data. However, it is not always realistic to assume that the data does not change during the sampling procedure, as described in [MNS11, GHR+12, GHS+12, HW13, NY15]. In this work, we study the robustness of sampling in an *adaptive* adversarial environment.

The adversarial environment. In high-level, the model is a two-player game between a randomized streaming algorithm, called Sampler, and an adaptive player, Adversary. In each round,

1. Adversary first submits an element to Sampler. The choice of the element can depend, possibly in a probabilistic manner, on all elements submitted by Adversary up to this point, as well as all information that Adversary observed from Sampler up to this point.

2. Next, Sampler probabilistically updates its internal state, i.e., the sample that it currently maintains. An update step usually involves an insertion of the newly received element to the sample with some probability, and sometimes deletion of old elements from the sample.

3. Finally, Adversary is allowed to observe the current (updated) state of Sampler, before proceeding to the next round.

Adversary's goal is to make the sample as *unrepresentative* as possible, causing Sampler to come with false conclusions about the data stream. The game is formally described in Section 2.

Adversarial scenarios are common and arise in different settings. An adversary uses *adversarial examples* to fool a trained machine learning model [SZS+14, MHS19]; In the field of online learning [Haz16], adversaries are typically adaptive [SS17, LMPL18]. An online store suggests recommended items based on a sample of previous purchases, which in turn influences future sales [Sha12, GHR+12]. A network device routes traffic according to statistics pulled from a sampled substream of packets [DLT05], and an adversary that observes the network's traffic learns the device's routing choices might cause a denial-of-service attack by generating a small amount of adversarial traffic [NY15]. A high-frequency stock trading algorithm monitors a stream of stock orders places buy/sell requires based on statistics drawn from samples; A competitor might fool the sampling algorithm by observing its requests and modifying future stock orders accordingly. An autonomous vehicle receives physical signals from its immediate environment (which might be adversarial [SBM+18]) and has to decide on a suitable course of action.

Even when there is no apparent adversary, the adaptive perspective is sometimes natural and required. For instance, adaptive data analysis [DFH+15, WFRS18] aims to understand the challenges arising when data arrives online, such as data reuse, the implicit bias "collected" over time in scientific discovery, and the evolution of statistical hypotheses over time. In graph algorithms, [CGP+18] observed that an adversarial analysis of dynamic spanners would yield a simpler (and quantitively better) alternative to their work.

In view of the importance of robustness against adaptive adversaries, and the fact that random sampling is very widely used in practice (including in streaming settings), we ask the following.

Are sampling algorithms robust against adaptive adversaries?

Bernoulli and reservoir sampling. We mainly focus on two of the most basic and well-known sampling algorithms: Bernoulli sampling and reservoir sampling. The Bernoulli sampling algorithm with parameter $p \in [0,1]$ runs as follows: whenever it receives a stream element x_i, the algorithm stores the element with probability p. For a stream of length n the sample size is expected to be np; and furthermore, it is well-concentrated around this value. We denote this algorithm by BernoulliSample.

The classical reservoir sampling algorithm [Vit85] (see also [Knu97, Section 3.4.2] and a formal description in Section 2) with parameter $k \in [n]$ maintains a uniform sample of fixed size k, acting as follows. The first k elements it receives, $x_1, \ldots, x_k$, are simply added to the memory with probability one. When the algorithm receives its i^{th} element x_i, where $i > k$, it stores it with probability k/i, by overriding a uniformly random element from the memory (so the memory size is kept fixed to k). We henceforth denote this algorithm by ReservoirSample.

Attacking sampling algorithms. To answer the question above of whether sampling algorithms are robust against adversarially chosen streams, we must first define a notion of a representative sample, as several notions might be appropriate. However, we begin the discussion with an example showing how to attack the Bernoulli (and reservoir) sampling algorithm with respect to merely any definition of "representative".

Consider a setting where the stream consists of n points $x_1, \ldots, x_n$ in the one-dimensional range of real numbers $[0,1]$. BernoulliSample receives these points and samples each one independently with probability $p < 1$. One can observe that, in the static setting and for sufficiently large p, the sampled set will be a good representation of the entire n points for various definitions of the term "representation". For example, the median of the stream will be ε-close[1] to the median of the sampled elements with high probability, as long as $p = \frac{c}{\varepsilon^2 n}$ for some constant $c > 0$ (this also holds for any other quantile).

Consider the following *adaptive* adversary which will demonstrate the difference of the adaptive setting. Adversary keeps a "working range" at any point during the game, starting with the full range $[0,1]$. In the first round, Adversary chooses the number $x_1 = 0.5$ as the first element in the stream. If x_1 is sampled, then Adversary moves to the range $[0.5,1]$, and otherwise, to the range $[0,0.5]$. Next, Adversary submits x_2 as the middle of the current range. This continues for n steps; Formally, Adversary's strategy is as follows. Set $a_1 = 0$ and $b_1 = 1$. In round i, where i runs

[1]The term "close" here means that the median of the sampled set will be an element whose order among the elements of the full stream, when the elements are sorted by value from smallest to largest, is within the range $(1 \pm \epsilon)n/2$, with high probability where the parameter ϵ depends on the probability p.

from 1 to n, Adversary submits $x_i = \frac{a_i+b_i}{2}$ to BernoulliSample; If x_i is sampled then Adversary sets $a_{i+1} = x_i, b_{i+1} = b_i$, and otherwise, it sets $a_{i+1} = a_i, b_{i+1} = x_i$. The final stream is $x_1, \ldots, x_n$.

Note that at any point throughout the process, Adversary always submits an element that is larger than all elements in the current sampled set, and also smaller than all the non-sampled elements of the stream. Therefore, the end result is that after this process is over, with probability 1, the k sampled elements are precisely the *smallest* k elements in the stream. Of course, the median of the sampled set is far from the median of the stream as such a subset is very *unrepresentative* of the data. Actually, one might consider it as the "most unrepresentative" subset of the data.

The exact same attack on BernoulliSample works almost as effectively against ReservoirSample. In this case, the attack will cause all of the k sampled elements at the end of the process to lie among the first $O(k \ln n)$ elements with high probability. For more details, see Section 5.

The good news. This attack joins a line of attacks in the adversarial model. Lipton and Naughton [LN93] showed that an adversary that can measure the time of operations in a dictionary can use this information to increase the probability of a collision and as a result, significantly decrease the performance of the hashtable. Hardt and Woodruff [HW13] showed that linear sketches are inherently non-robust and cannot be used to compute the Euclidean norm of its input (where in the static setting they are used mainly for this reason). Naor and Yogev [NY15] showed that Bloom filters are susceptible to attacks by an adaptive stream of queries if the adversary is computationally unbounded and they also constructed a robust Bloom filter against computationally bounded adversaries.

In our case, we note that the given attack might categorize it as "theoretical" only. In practice, it is unrealistic to assume that the universe from which Adversary can pick elements is an infinite set; how would the attack look, then, if the universe is the discrete set $[N] = \{1, \ldots, N\}$? Adversary splits the range $[0, 1]$ to half for n times, meaning that the precision of the elements required is exponential; The analogous attack in the discrete setting requires N to be exponentially large with respect to the stream size n. Such a universe size is large and "unrealistic": for Sampler to memorize even a single element requires memory size that is linear in n, whilst sampling and streaming algorithms usually aim to use an amount sublinear in n of memory.

Thus, the question remains whether there exist attacks that can be performed on elements using substantially less precision, that is, on a significantly smaller size of discrete universe. In this work, we bring good news to both the Bernoulli and reservoir sampling algorithms by answering this question *negatively*. We show that both sampling algorithms, with the right parameters, will output a representative sample with good probability regardless of Adversary's strategy, thus exhibiting robustness for these algorithms in adversarial settings.

We note that any *deterministic* algorithm that works in the static setting is inherently robust in the adversarial adaptive setting as well. However, in many cases, deterministic algorithms with small memory simply do not exist, or they are complicated and tailored for a specific task. Here, we enjoy the simplicity of a generic randomized sampling algorithm combined with the robust guarantees of our framework.

What is a representative sample? Perhaps the most standard and well-known notion of being representative is that of an ε-approximation, first suggested by Vapnik and Chervonenkis [VC71] (see also [MV17]), which originated as a natural notion of discrepancy [Cha01] in the geometric literature. It is closely related to the celebrated notion of VC-dimension [VC71, Sau72, She72], and

captures many quantitative properties that are desired in a random subset. Let $X = (x_1, \ldots, x_n)$ be a sequence of elements from the universe U (repetitions are allowed) and let $R \subseteq U$. The *density* of R in X is the fraction of elements in X that are also in R (i.e., $d_R(X) = \Pr_{i \in [n]}[x_i \in R]$).

A *set system* is simply a pair $(U, \mathcal{R})$ where $\mathcal{R} \subseteq 2^U$ is a collection of subsets. A non-empty subsequence S of X is an *ε-approximation* of X with respect to the set system $(U, \mathcal{R})$ if it preserves densities (up to an ε factor) for all subsets $R \in \mathcal{R}$.

Definition 1.1 (ε-approximation)**.** *We say that a (non-empty) sample S is an* ε-approximation *of X with respect to $\mathcal{R}$ if for any subset $R \in \mathcal{R}$ it holds that $|d_R(X) - d_R(S)| \leq \varepsilon$.*

If the universe U is well-ordered, it is natural to take $\mathcal{R}$ as the collection of all consecutive intervals in U, that is, $\mathcal{R} = \{[a, b] : a \leq b \in U\}$ (including all singletons $[a, a]$). With this set system in hand, ε-approximation is a natural form of "good representation" in the streaming setting, pointed out by its deep connection to multiple classical problems in the streaming literature, like approximate median, and more generally, quantile estimation [MRL99, GK01, WLYC13, GK16, KLL16] and range searching [BCEG07]. In particular, if S is an ε-approximation of X w.r.t. $(U, \mathcal{R})$, then any q-quantile of S is ε-close to the q-quantile of X; this holds simultaneously for all quantiles (see Section 1.2).

1.1 Our Results

Fix a set system $(U, \mathcal{R})$ over the universe U. A sampling algorithm is called *(ε, δ)-robust* if for any (even computationally unbounded) strategy of Adversary, the output sample S is an ε-approximation of the whole stream X with respect to $(U, \mathcal{R})$, with probability at least $1 - \delta$.

Our main result is an upper bound ("good news") on the (ε, δ)-robustness of Bernoulli and reservoir sampling, later to be complemented them with near-matching lower bounds.

Theorem 1.2. *For any $0 < \varepsilon, \delta < 1$, set system $(U, \mathcal{R})$, and stream length n, the following holds.*

- BernoulliSample *with parameter $p \geq 10 \cdot \frac{\ln|\mathcal{R}| + \ln(4/\delta)}{\varepsilon^2 n}$ is (ε, δ)-robust.*
- ReservoirSample *with parameter $k \geq 2 \cdot \frac{\ln|\mathcal{R}| + \ln(2/\delta)}{\varepsilon^2}$ is (ε, δ)-robust.*

The proof appears in Section 4. As the total number of elements sampled by BernoulliSample is well-concentrated around np, the above theorem implies that a sample of total size (at least) $\Theta(\frac{\ln|\mathcal{R}| + \ln\frac{1}{\delta}}{\varepsilon^2})$, obtained by any of the algorithms, BernoulliSample or ReservoirSample, is an ε-approximation with probability $1 - \delta$.

This should be compared with the static setting, where the same result is known as long as $p \geq c \cdot \frac{d + \ln\frac{1}{\delta}}{\varepsilon^2 n}$ for BernoulliSample, and $k \geq c \cdot \frac{d + \ln\frac{1}{\delta}}{\varepsilon^2}$ for ReservoirSample, where d is the VC-dimension of $(U, \mathcal{R})$ and $c > 0$ is a constant [VC71, Tal94, LLS01] (see also [MV17]).

As you can see, to make the static sampling algorithm robust in the adaptive setting one solely needs to modify the sample size by replacing the VC-dimension term d with the cardinality dimension $\ln|\mathcal{R}|$ (and update the multiplicative constant). Below, in our lower bounds, we show that this increase in the sample size is inherent, and not a byproduct of our analysis.

Lower Bounds. We next show that being adaptively robust comes at a price. That is, the dependence on the cardinality dimension, as opposed to the VC dimension, is necessary. By an improved version of the attack described in the introduction, we show the following:

Theorem 1.3. *There exists a constant $c > 0$ and a set system $(U, \mathcal{R})$ with VC-dimension 1, where such that for any $0 < \varepsilon, \delta < 1/2$:*

1. *The* BernoulliSample *algorithm with parameter $p < c \cdot \frac{\ln |\mathcal{R}|}{n \ln n}$ is **not** (ε, δ)-robust.*
2. *The* ReservoirSample *algorithm with parameter $k < c \cdot \frac{\ln |\mathcal{R}|}{\ln n}$ is **not** (ε, δ)-robust.*

Moreover, for any $n^{6 \ln n} \leq N \leq 2^{n/2}$, there exists $(U, \mathcal{R})$ as above where $|\mathcal{R}| = |U| = N$.

The proof can be found in Section 5.

Continuous robustness. The condition of (ε, δ)-robustness requires that the sample will be ε-representative of the stream *in the end of the process*. What if we wish the sample to be representative of the stream *at any point* throughout the stream? Formally, we say that a sampling algorithm is (ε, δ)-*continuously robust* if, with probability at least $1 - \delta$, at *any* point $i \in [n]$ the sampled set S_i is an ε-approximation of the first i elements of the stream, i.e., of $X_i = (x_1, \ldots, x_i)$. The next theorem shows that continuous robustness of ReservoirSample can be obtained with just a small overhead compared to "standard" robustness. (For BernoulliSample one cannot hope for such a result to be true, at least for the above definition of continuous robustness.)

Theorem 1.4. *There exists $c > 0$, such that for any $0 < \varepsilon, \delta < 1/2$, set system $(U, \mathcal{R})$, and stream length n,* ReservoirSample *with parameter $k \geq c \cdot \frac{\ln |\mathcal{R}| + \ln 1/\delta + \ln 1/\varepsilon + \ln \ln n}{\varepsilon^2}$ is (ε, δ)-continuously robust.*

Moreover, if only continuous robustness against a static adversary is desired, then the $\ln |\mathcal{R}|$ term can be replaced with the VC-dimension of $(U, \mathcal{R})$.

We are not aware of a previous analysis of continuous robustness, even in the static setting. The proof, appearing in Section 6, follows by applying Theorem 1.2 (or its static analogue) in carefully picked "checkpoints" $k = i_1 \leq i_2 \leq \ldots \leq i_t = n$ along the stream, where $t = O(\varepsilon^{-1} \ln n)$. It shows that if the sample S_i is representative of the stream X_i in any of the points $i = i_1, \ldots, i_{t-1}$, then with high probability, the sample is also representative in any other point along the stream. (We remark that a similar statement with weaker dependence on n can be obtained from Theorem 1.2 by a straightforward union bound.) The proof can be found in Section 6.

Comparison to deterministic sampling algorithms. Our results show that sampling algorithms provide an ε-approximation in the adversarial model. One advantage of using the notion of ε-approximation is its wide array of applications, where for each such task we get a streaming algorithm in the adversarial model as described in the following subsection. We stress that for any specific task a *deterministic* algorithm that works in the static setting will also automatically be robust in the adversarial setting. However, deterministic algorithms tend to be more complicated, and in some cases they require larger memory. Here, we focus on showing that the most simple and generic sampling algorithms "as is" are robust in our adaptive model and yield a representative sample of the data that can be used for many different applications.

The best known *deterministic* algorithm for computing an ε-approximating sample in the streaming model is that of Bagchi et al. [BCEG07]. The sample size they obtain is $O(\varepsilon^{-2} \ln 1/\varepsilon)$;

the working space of their algorithm and the processing time per element are of the form $\varepsilon^{-2d-O(1)}(\ln n)^{O(d)}$, where d is the *scaffold dimension*[2] of the set system. The exact bounds are rather intricate, see Corollary 4.2 in [BCEG07]. While the space requirement of their approach does not have a dependence on $\ln|\mathcal{R}|$, its dependence on ε and $\ln n$ is generally worse than ours, making their bounds somewhat incomparable to ours. Finally, we note that there exist more efficient methods to generate an ε-approximation in some special cases, e.g., when the set system constitutes of rectangles or halfspaces [STZ04].

1.2 Applications of Our Results

We next describe several representative applications and usages of ε-approximations (see also [BCEG07] for more applications in the area of robust statistics). For some of these applications, there exist deterministic algorithms known to require less memory than the simple random sampling models discuss in this paper. However, one area where our generic random sampling approach shines compared to deterministic approaches is the *query complexity* or *running time* (under a suitable computational model). Indeed, while deterministic algorithms must inherently query *all* elements in the stream in order to run correctly, our random sampling methods query just a small sublinear portion of the elements in the stream.

Consequently, to the best of our knowledge, Bernoulli and reservoir sampling are the first two methods known to compute an ε-approximation (and as a byproduct, solve the tasks described in this subsection) in adversarial situations where it is unrealistic or too costly to query all elements in the stream. The last part of this subsection exhibits an example of one such situation.

Quantile approximation. As was previously mentioned, ε-approximations have a deep connection to approximate median (and more generally, quantile estimation). Assume the universe U is well-ordered. We say that a streaming algorithm is an (ε,δ)-*robust quantile sketch* if, in our *adversarial model*, it provides a sample that allows to approximate the rank[3] of any element in the stream up to additive error εn with probability at least $1-\delta$. Observe that this is achieved with an ε-approximation with respect to the set system $(U,\mathcal{R})$ where $\mathcal{R}=\{[1,b] : b\in U\}$. For example, set b to be the median of the stream. Since the density of the range $[1,b]$ is preserved in the sample, we know that the median of the sample will be ε-close to the median of the stream. This works for any other quantile simultaneously. The sample size is $\Theta(\frac{\ln|U|+\ln(1/\delta)}{\varepsilon^2})$.

Corollary 1.5. *For any $0<\varepsilon,\delta<1$, well-ordered universe U, and stream length n,* BernoulliSample *with parameter $p\geq 10\cdot\frac{\ln|U|+\ln(4/\delta)}{\varepsilon^2 n}$ is an (ε,δ)-robust quantile sketch. The same holds for the* ReservoirSample *algorithm with parameter $k\geq 2\cdot\frac{\ln|U|+\ln(2/\delta)}{\varepsilon^2}$.*

A corollary in the same spirit regarding *continuously* robust quantile sketches can be derived from Theorem 1.4.

Range queries. Suppose that the universe is of the form $U=[m]^d$ for some parameters m and d. One basic problem is that of *range queries*: one is given a set of ranges $\mathcal{R}$ and each query consists of a range $R\in\mathcal{R}$ where the desired answer is the number of points in the stream that

[2]The scaffold dimension is a variant of the VC-dimension equal to $\lceil\ln|\mathcal{R}|/\ln|U|\rceil$.

[3]The *rank* of an element x_i in a stream $x_1,\ldots,x_n$ is the total amount of elements x_j in the stream so that $x_j\leq x_i$.

are in this range. Popular choices of such ranges are axis-aligned or rotated boxes, spherical ranges and simplicial ranges. An ε-approximation allows us to answer such range queries up to an additive error of εn. Suppose the sampled set is S, then an answer is given by computing $d_R(S) \cdot n/|S|$. For example, when $\mathcal{R}$ consists of all axis-parallel boxes, $\ln|\mathcal{R}| = O(d \ln m)$ and thus the sample size required to answer range queries that are robust against adversarial streams is $|S| = O\left(\frac{d \ln |m| + \ln(1/\delta)}{\varepsilon^2}\right)$; for rotated boxes, one should replace d with d^2 in this expression. See [BCEG07] for more details on the connection between ε-approximations and range queries.

Center points. Our result is also useful for computing β-center points. A point c in the stream is a β-center point if every closed halfspace containing c in fact contains at least βn points of the stream. In [CEM+96, Lemma 6.1] it has been shown that an ε-approximation (with respect to half-spaces) can be used to get a β-center point for suitable choices of the parameters. For example, setting $\varepsilon = \beta/5$ we get that a $6\beta/5$-center of the sample S is a β-center of the stream X. Thus, we can compute a β-center of a stream in the adversarial model. See also [BCEG07].

Heavy hitters. Finding those elements that appear many times in a stream is a fundamental problem in data mining, with a myriad of practical applications. In the *heavy hitters* problem, there is a threshold α and an error parameter ε. The goal is to output a list of elements such that if an element x appears more than αn times in the stream (i.e., $d_x(X) \geq \alpha$) it must be included in the list, and if an element appears less than $(\alpha - \varepsilon)n$ times in the stream (i.e., $d_x(X) \leq \alpha - \varepsilon$ it cannot be included in the list.

Our results yield a simple and efficient heavy hitters streaming algorithm in the adversarial model. For any universe U let $\mathcal{R} = \{\{a\} : a \in U\}$ be the set of all singletons. Now, pick $\varepsilon' = \varepsilon/3$ and use either Bernoulli or reservoir sampling to compute an ε'-approximation S of the stream X, outputting all elements $x \in S$ with $d_{\{x\}}(S) \geq \alpha - \varepsilon'$. Indeed, if $d_a(X) \geq \alpha$ then $d_x(S) \geq \alpha - \varepsilon'$. On the other hand, if $d_x(X) \leq \alpha - \varepsilon$ then $d_x(S) \leq \alpha - \varepsilon + \varepsilon' < \alpha - \varepsilon'$.

Corollary 1.6. *There exists $c > 0$ such that for any $0 < \varepsilon, \delta < 1/2$, universe U, and stream length n,* BernoulliSample *with parameter $p \geq c \cdot \frac{\ln|U| + \ln(1/\delta)}{\varepsilon^2 n}$ solves the heavy hitters problem with error ε in the adversarial model. The same holds for* ReservoirSample *with parameter $k \geq c \cdot \frac{\ln|U| + \ln(1/\delta)}{\varepsilon^2}$.*

Clustering. The task of partitioning data elements into separate groups, where the elements in each group are "similar" and elements in different groups are "dissimilar" is fundamental and useful for numerous applications across computer science. There has been lots of interest on clustering in a streaming setting, see e.g. [GLA16] for a survey on recent results. Our results suggest a generic framework to accelerate clustering algorithms in the adversarial model: Instead of running clustering on the full data, one can simply sample the data to obtain (with high probability, even against an adversary) an ε-approximation of it, run the clustering algorithm on the sample, and then extrapolate the results to the full dataset.

Sampling in modern data-processing systems. It is very common to use random sampling (sometimes "in disguise") in modern data-intensive systems that operate on streaming data, arriving in an online manner. As an illustrative example, consider the following *distributed database* [OV11] setting. Suppose that a database system must receive and process a huge amount of

queries per second. It is unrealistic for a single server to handle all the queries, and hence, for load balancing purposes, each incoming query is randomly assigned to one of K query-processing servers. Seeing that the set of queries that each such server receives is essentially a Bernoulli random sample (with parameter $p = 1/K$) of the full stream, one hopes that the portion of the stream sampled by each of these servers would truthfully represent the whole data stream (e.g., for query optimization purposes), even if the stream changes with time (either unintentionally or by a malicious adversary). Such "representation guarantees" are also desirable in distributed machine learning systems [GDG+17, SKYL17], where each processing unit learns a model according to the portion of the data it received, and the models are then aggregated, with the hope that each of the units processed "similar" data.

In general, modern data-intensive systems like those described above become more and more complicated with time, consisting of a large number of different components. Making these systems *robust* against environmental changes in the data, let alone *adversarial* changes, is one of the greatest challenges in modern computer science. From our perspective, the following question naturally emerges:

Is random sampling a risk in modern data processing systems?

Fortunately, our results indicate that the answer to this question is largely *negative*. Our upper bounds, Theorems 1.2 and 1.4, show that a sufficiently large sample suffices to circumvent adversarial changes of the environment.

1.3 Related Work

Online learning. One related field to our work is *online learning*, which was introduced for settings where the data is given in a sequential online manner or where it is necessary for the learning algorithm to adapt to changes in the data. Examples include stock price predictions, ad click prediction, and more (see [Sha12] for an overview and more examples).

Similar to our model, online learning is viewed as a repeated game between a learning algorithm (or a predictor) and the environment (i.e., the adversary). It considers n rounds where in each round the environment submits an instance x_i, the learning algorithm then makes a prediction for x_i, the environment, in turn, chooses a loss for this prediction and sends it as feedback to the algorithm. The goal in this model is usually to minimize regret (the sum of losses) compared to the best fixed prediction in hindsight. This is the typical setting (e.g., [HAK07, SST10]), however, many different variants exist (e.g., [DGS15, ZLZ18]).

PAC learning. In the PAC-learning framework [Val84], the learner algorithm receives samples generated from an unknown distribution and must choose a hypothesis function from a family of hypotheses that best predicts the data with respect to the given distribution. It is known that the number of samples required for a class to be learnable in this model depends on the VC-dimension of the class.

A recent work of Cullina et al. [CBM18] investigates the effect of evasion adversaries on the PAC-learning framework, coining the term of *adversarial VC-dimension* for the parameter governing the sample complexity. Despite the name similarity, their context is seemingly unrelated to ours (in particular, it is not a streaming setting), and correspondingly, their notion of adversarial VC-dimension does not seem to relate to our work.

Adversarial examples in deep learning. A very popular line of research in modern deep learning proposes methods to attack neural networks, and countermeasures to these attacks. In such a setting, an adversary performs adaptive queries to the learned model in order to fool the model via a malicious input. The learning algorithms usually have an underlying assumption that the training and test data are generated from the same statistical distribution. However, in practice, the presence of an adaptive adversary violates this assumption. There are many devastating examples of attacks on learning models [SZS+14, BCM+13, PMG+17, BR18, MHS19] and we stress that currently, the understanding of techniques to defend against such adversaries is rather limited [GMP18, MW18, MM19, MHS19].

Maintaining random samples. Reservoir sampling is a simple and elegant algorithm for maintaining a random sample of a stream [Vit85], and since its proposal, many flavors have been introduced. Chung, Tirthapura, Woodruff [CTW16] generalized reservoir sampling to the setting of multiple distributed streams, which need to coordinate in order to continuously respond to queries over the union of all streams observed so far (see also Cormode et al. [CMYZ12]). Another variant is weighted reservoir sampling where the probability of sampling an element is proportional to a weight associated with the element in the stream [ES06, BOV15]. A distributed version as above was recently considered for the weighted case as well [JSTW19].

1.4 Paper Organization

Section 2 contains an overview of our adversarial model and a more precise and detailed definition than the one given in the introduction. In Section 3 we mention several concentration inequalities required for our analysis. In Section 4 we present and prove our main technical Lemma, from which we derive Theorem 1.2. This includes analysis of both BernoulliSample and ReservoirSample. In Section 5 we present our "attack", i.e., our lower bound showing the tightness of our result. Finally, in Section 6, we prove our upper bounds in the continuous setting.

2 The Adversarial Model for Sampling

In this section, we formally define the online adversarial model discussed in this paper. Roughly speaking, we say that Sampler is an (ε, δ)-robust sampling algorithm for a set system $(U, \mathcal{R})$ if for any adversary choosing an adaptive stream of elements $X = (x_1, \ldots, x_n)$, the final state of the sampling algorithm σ_n is an ε-approximation of the stream with probability $1 - \delta$. This is formulated using a game, AdaptiveGame, between two players, Sampler and Adversary.

Rules of the game:

1. Sampler is a streaming algorithm, which gets a sequence of n elements one by one $x_1, \ldots, x_n$ in an online manner (the sampling algorithms we discuss in this paper do not need to know n in advance). Upon receiving an element x_i, Sampler can perform an arbitrary computation (the running time can be unbounded) and update a local state σ. We denote the local state after i steps by σ_i, and write $\sigma_i \leftarrow \mathsf{Sampler}(\sigma_{i-1}, x_i)$.

2. The stream is chosen adaptively by Adversary: a probabilistic (unbounded) player that, given all previously sent elements $x_1, \ldots, x_{i-1}$ and the current state σ_{i-1}, chooses the next element

x_i to submit. The *strategy* that Adversary employs along the way, that is, the probability distribution over the choice of x_i given any possible set of values $x_1, \ldots, x_{i-1}$ and σ_{i-1}, is fixed in advance. The underlying (finite or infinite) set from which Adversary is allowed to choose elements during the game is called the *universe*, and denoted by U. We assume that U does not change along the game.

3. Once all n rounds of the game have ended, Sampler outputs σ_n. For the sampling algorithms discussed in this paper, $S := \sigma_n$ is a subsequence of the stream $X = (x_1, \ldots, x_n)$. S is usually called the *sample* obtained by Sampler in the game.

For an illustration on the rules of the game see Figure 1.

The game AdaptiveGame

Parameters: n, ε, $(U, \mathcal{R})$.

1. Set $\sigma_0 = \perp$.
2. For $i = 1 \ldots n$ do:
   (a) Adversary$(\sigma_{i-1}, x_1, \ldots, x_{i-1})$ submits the query x_i.
   (b) Set $\sigma_i \leftarrow$ Sampler(σ_{i-1}, x_i).
3. Let $S = \sigma_n$, and output 1 if S is an ε-representative sample of $X = x_1, \ldots, x_n$ with respect to $(U, \mathcal{R})$, and 0 otherwise.

Figure 1: The definition of the game AdaptiveGame$_\varepsilon$ between a streaming algorithm Sampler and Adversary. Here the adversary chooses the next element to the stream while given the state (memory) of the streaming algorithm thus-far. In the beginning of the game, Adversary receives the parameters $n, \varepsilon, (U, \mathcal{R})$ and knows exactly which sampling algorithm is employed by Sampler.

Using the game defined above, we now describe what it means for a sampling algorithm to be (adversarially) *robust*.

Definition 2.1 (Robust sampling algorithm)**.** *We say that a sampling algorithm* Sampler *is* (ε, δ)*-robust with respect to the set system* $(U, \mathcal{R})$ *and the stream length* n *if for and any (even unbounded) strategy of* Adversary*, it holds that*

$$\Pr[\text{AdaptiveGame}(\text{Sampler}, \text{Adversary}) = 1] \geq 1 - \delta$$

The memory size used by Sampler *is defined to be the maximal size of* σ *throughout the process of* AdaptiveGame*.*

A stronger requirement that one can impose on the sampling algorithm is to hold an ε-approximation of the stream at *any* step during the game. To handle this, we define a continuous variant of AdaptiveGame which we denote ContinuousAdaptiveGame, presented in Figure 2.

For the sampling algorithms that we consider, the state at any time σ_i is essentially equal to the sample S_i. In any case, the definition of the framework given in Figure 2 generally allows σ_i

The game ContinuousAdaptiveGame

Parameters: n, ε, $(U, \mathcal{R})$.

1. Set $\sigma_0 = \perp$.
2. For $i = 1 \ldots n$ do:
   (a) Adversary$((\sigma_{i-1}, S_{i-1}), x_1, \ldots, x_{i-1})$ submits the query x_i.
   (b) Set $(\sigma_i, S_i) \leftarrow$ Sampler(σ_{i-1}, x_i).
   (c) If S_i is not an ε-approximation of $X_i = x_1, \ldots, x_i$ with respect to $(U, \mathcal{R})$ then output 0 and halt.
3. Output 1.

Figure 2: The game corresponding to the continuous variant, ContinuousAdaptiveGame between a streaming algorithm Sampler and an Adversary. Here, Sampler is required to hold an ε-approximating sampled set S_i after each step.

to contain additional information, if needed. A sampling algorithm is called (ε, δ)-*continuously robust* if the following holds with probability at least $1 - \delta$: for any strategy of Adversary, and all $i \in [n]$, the sample S_i is an ε-approximation of the stream at time i.

Definition 2.2 (Continuously robust sampling algorithm)**.** *We say that a sampling algorithm* Sampler *is* (ε, δ)*-continuously robust with respect to the set system* $(U, \mathcal{R})$ *and the stream length* n *if for and any (even unbounded) strategy of* Adversary*, it holds that*

$$\Pr[\mathsf{ContinuousAdaptiveGame}(\mathsf{Sampler}, \mathsf{Adversary}) = 1] \geq 1 - \delta$$

The memory size used by Sampler *is defined to be the maximal size of* σ *throughout the process of* ContinuousAdaptiveGame*.*

Reservoir sampling. For completeness, we provide the pseudocode of the reservoir sampling algorithm [Vit85, Knu97]. Here, k denotes the (fixed) memory size of the algorithm, i denotes the current round number, and x_i is the currently received element.

ReservoirSample$(k, i, \sigma_{i-1}, x_i)$:

1. If $i < k$ then parse $\sigma_{i-1} = x_1, \ldots, x_{i-1}$ and output $\sigma_i = x_1, \ldots, x_i$.
2. Otherwise, parse $\sigma_{i-1} = s_1, \ldots, s_k$.
3. With probability k/i do:
   choose $j \in [k]$ uniformly at random and output $\sigma_i = s_1, \ldots, s_{j-1}, x_i, s_{j+1}, \ldots, s_k$.
4. Otherwise, output $\sigma_i = \sigma_{i-1}$.

3 Technical Preliminaries

The logarithms in this paper are usually of base e, and denoted by ln. The exponential function $\exp(x)$ is e^x. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. We state some concentration inequalities, useful for our analysis in later sections. We start with the well-known Chernoff's inequality for sums of independent random variables.

Theorem 3.1 (Chernoff Bound [Che52]; see Theorem 3.2 in [CL06]). *Let $X_1, \ldots, X_m$ be independent random variables that take the value 1 with probability p_i and 0 otherwise, $X = \sum_{i=1}^{m} X_i$, and $\mu = \mathbb{E}[X]$. Then for any $0 < \delta < 1$,*

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2 \mu}{2}\right)$$

and

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2 \mu}{2+2\delta/3}\right).$$

Our analysis of adversarial strategies crucially makes use of *martingale inequalities*. We thus provide the definition of a martingale.

Definition 3.2. *A* martingale *is a sequence $X = (X_0, \ldots, X_m)$ of random variables with finite means, so that for $0 \leq i < m$, it holds that $\mathbb{E}[X_{i+1} \mid X_0, \ldots, X_i] = X_i$.*

The most basic and well-known martingale inequality, Azuma's (or Hoeffding's) inequality, asserts that martingales with bounded differences $|X_{i+1} - X_i|$ are well-concentrated around their mean. For our purposes, this inequality does not suffice, and we need a generalized variant of it, due to McDiarmid [McD98, Theorem 3.15]; see also Theorem 4.1 in [Fre75]. The formulation that we shall use is given as Theorem 6.1 in the survey of Chung and Lu [CL06].

Lemma 3.3 (See [CL06], Theorem 6.1). *Let $X = (X_0, X_1, \ldots, X_n)$ be a martingale. Suppose further that for any $1 \leq i \leq n$, the variance satisfies $Var(X_i | X_0, \ldots, X_{i-1}) \leq \sigma_i^2$ for some values $\sigma_1, \ldots, \sigma_n \geq 0$, and there exists some $M \geq 0$ so that $|X_i - X_{i-1}| \leq M$ always holds. Then, for any $\lambda \geq 0$, we have*

$$\Pr(X - X_0 \geq \lambda) \leq \exp\left(-\frac{\lambda^2}{2\sum_{i=1}^{n}(\sigma_i^2) + M\lambda/3}\right).$$

In particular,

$$\Pr(|X - X_0| \geq \lambda) \leq 2\exp\left(-\frac{\lambda^2}{2\sum_{i=1}^{n}(\sigma_i^2) + M\lambda/3}\right).$$

Unlike Azuma's inequality, Lemma 3.3 is well-suited to deal with martingales where the maximum value M of $|X_{i+1} - X_i|$ is large, but the maximum is rarely attained (making the variance much smaller than M^2). The martingales we investigate in this paper depict this behavior.

4 Adaptive Robustness of Sampling: Main Technical Result

In this section, we prove the main technical lemma underlying our upper bounds for Bernoulli sampling and reservoir sampling. The lemma asserts that for both sampling methods, and any given subset R of the universe U, the fraction of elements from R within the sample typically does not differ by much from the corresponding fraction among the whole stream.

Lemma 4.1. *Fix $\varepsilon, \delta > 0$, a universe U and a subset $R \subseteq U$, and let $X = (x_1, x_2, \ldots, x_n)$ be the sequence chosen by* Adversary *in* $\mathsf{AdaptiveGame}_\varepsilon$ *against either* BernoulliSample *or* ReservoirSample.

1. *For* BernoulliSample *with parameter $p \geq 10 \cdot \frac{\ln(4/\delta)}{\varepsilon^2 n}$, we have $\Pr(|d_R(X) - d_R(S)| \geq \varepsilon) \leq \delta$.*
2. *For* ReservoirSample *with memory size $k \geq 2 \cdot \frac{\ln(2/\delta)}{\varepsilon^2}$, it holds that $\Pr(|d_R(X) - d_R(S)| \geq \varepsilon) \leq \delta$.*

Both of these bounds are tight up to an absolute multiplicative constant, even for a static adversary (that has to submit all elements in advance); see Section 6 for more details.

The proof of Theorem 1.2 follows immediately from Lemma 4.1, and is given below. The proof of Theorem 1.4 requires slightly more effort, and is given in Section 6.

Proof of Theorem 1.2. Let $(U, \mathcal{R})$, ε, δ, n be as in the statement of the theorem, and let X and S denote the stream and sample, respectively. We start with the Bernoulli sampling case, and assume that $p \geq 10 \cdot \frac{\ln(4/\delta)+\ln|\mathcal{R}|}{\varepsilon^2 n} = 10 \cdot \frac{\ln(4|\mathcal{R}|/\delta)}{\varepsilon^2 n}$. For each $R \in \mathcal{R}$, we apply the first part of Lemma 4.1 with parameters ε and $\delta/|\mathcal{R}|$, concluding that

$$\Pr(|d_R(X) - d_R(S)| \geq \varepsilon) \leq \delta/|\mathcal{R}|.$$

In the event that $|d_R(X) - d_R(S)| \leq \varepsilon$ for any R, by definition S is an ε-approximation of X. Taking a union bound over all $R\mathcal{R}$, we conclude that the probability of this event *not* to hold is bounded by $|\mathcal{R}| \cdot (\delta/|\mathcal{R}|) = \delta$, meaning that BernoulliSample with p as above is (ε, δ)-robust.

The proof for ReservoirSample is identical, except that we replace the condition on p with the condition that $k \geq 2 \cdot \frac{\ln(2/\delta)+\ln|\mathcal{R}|}{\varepsilon^2}$, and apply the second part of Lemma 4.1. □

It is important to note that the typical proofs given for statements of this type in the static setting (i.e., when Adversary submits all elements in advance, and cannot act adaptively) do not apply for our adaptive setting. Indeed, the usual proof of the static analogue of the above lemma goes along the following lines: Adversary chooses which elements to submit in advance, and in particular, determines the number of elements from A sent, call it n_A. Then, the number of sampled elements from A is distributed according to the binomial distribution $\mathsf{Bin}(n_A, p)$ for Bernoulli sampling, and $\mathsf{Bin}(n_A, k/n)$ for reservoir sampling. One can then employ Chernoff bound to conclude the proof. This kind of analysis crucially relies on the adversary being static.

Here, we need to deal with an adaptive adversary. Recall that Adversary at any given point is modeled as a probabilistic process, that given the sequence $X_{i-1} = (x_1, \ldots, x_{i-1})$ of elements sent until now, and the current state σ_{i-1} of Sampler, probabilistically decides which element x_i to submit next. Importantly, this makes for a well-defined probability space, and allows us to analyze Adversary's behavior with probabilistic tools, specifically with concentration inequalities.

Chernoff bound cannot be used here, as it requires the choices made by the adversary along the process to be independent of each other, which is clearly not the case. In contrast, martingale inequalities are suitable for this setting. We shall thus employ these, specifically Lemma 3.3, to prove both parts of our main result in this section.

4.1 The Bernoulli Sampling Case

We start by proving the Bernoulli sampling case (first statement of Lemma 4.1). Recall that here each element is sampled, independently, with probability p. At any given point $0 \leq i \leq n$ along

the process, let $X_i = (x_1, \ldots, x_i)$ denote the sequence of elements submitted by the adversary until round i, and let $S_i \subseteq X_i$ denote the subsequence of sampled elements from X_i. Note that $X_n = X$ and $S_n = S$, and hence, to prove the lemma, we need to show that $|d_R(X_n) - d_R(S_n)| \leq \varepsilon$.

As a first attempt, it might make sense to try applying a martingale concentration inequality on the sequence of random variables $(Y_0, Y_1, \ldots, Y_n)$, where we define $Y_i = d_R(X_i) - d_R(S_i)$. Indeed, our end-goal is to bound the probability that Y_n significantly deviates from zero. However, a straightforward calculation shows that this is not a martingale, since the condition that $E[Y_i | Y_0, \ldots, Y_{i-1}] = 0$ does not hold in general. To overcome this, we show that a slightly different formulation of the random variables at hand does yield a martingale. Given the above $R \subseteq U$, for any $0 \leq i \leq n$ we define the random variables

$$A_i^R = \frac{i}{n} \cdot d_R(X_i) = \frac{|R \cap X_i|}{n} \quad ; \quad B_i^R = \frac{|R \cap S_i|}{np} \quad ; \quad Z_i^R = B_i^R - A_i^R, \tag{1}$$

where, as before, the intersection between a set R and a sequence X_i is the subsequence of X_i consisting of all elements that also belong to R.

Importantly, as is described in the next claim, the sequence of random variables $Z^R = (Z_0^R, \ldots, Z_n^R)$ defined above forms a martingale. The claim also demonstrates several useful properties of these random variables, to be used later in combination with Lemma 3.3.

Claim 4.2. *The sequence $(Z_0^R, Z_1^R, \ldots, Z_n^R)$ is a martingale. Furthermore, the variance of Z_i^R conditioned on $Z_0^R, \ldots, Z_{i-1}^R$ is bounded by $1/n^2p$, and it always holds that $|Z_i^R - Z_{i-1}^R| \leq 1/np$.*

We shall prove Claim 4.2 later on; first we use it to complete the proof of the main result.

Proof of Lemma 4.1, Bernoulli sampling case. It suffices to prove the following two inequalities for any p satisfying the conditions of the lemma for the Bernoulli sampling case:

$$\Pr(|A_n^R - B_n^R| \geq \varepsilon/2) \leq \delta/2 \quad ; \quad \Pr(|B_n^R - d_R(S_n)| \geq \varepsilon/2) \leq \delta/2. \tag{2}$$

Indeed, taking a union bound over these two inequalities, applying the triangle inequality, and observing that $A_n^R = d_R(X_n)$, we conclude that $\Pr(|d_R(X_n) - d_R(S_n)| \geq \varepsilon) \leq \delta$, as desired.

The first inequality follows from Claim 4.2 and Lemma 3.3. Indeed, in view of Claim 4.2, we can apply Lemma 3.3 on $(Z_0^R, \ldots, Z_n^R)$ with parameters $\lambda = \varepsilon/2$, $\sigma_i^2 = 1/n^2p$, and $M = 1/np$. As $Z_0^R = 0$, we have $|A_n^R - B_n^R| = |Z_n^R - Z_0^R|$, and so

$$\Pr(|A_n^R - B_n^R| \geq \varepsilon/2) \leq 2\exp\left(-\frac{(\varepsilon/2)^2}{2n \cdot \frac{1}{n^2p} + \frac{\varepsilon}{6np}}\right) < 2\exp\left(-\frac{\varepsilon^2 np}{9}\right).$$

The right hand side is bounded by $\delta/2$ when $np \geq \frac{9}{\varepsilon^2}\ln(\delta/4)$, settling the first inequality of (2).

We next prove the second inequality of (2). Observe that $B_n^R = d_R(S_n) \cdot \frac{|S_n|}{np}$. Since each element is added to the sample with probability p, independently of other elements, the size of S_n is distributed according to the binomial distribution $\mathsf{Bin}(n, p)$, regardless of the adversary's strategy. Applying Chernoff inequality with $\delta = \varepsilon/2$, we get that

$$\Pr(\big||S_n| - np\big| \geq \varepsilon np/2) \leq 2\exp\left(-\frac{(\varepsilon/2)^2 np}{2 + \varepsilon/3}\right) < 2\exp\left(-\frac{\varepsilon^2 np}{10}\right).$$

This probability is bounded by $\delta/2$ provided that $np \geq \frac{10\ln(4/\delta)}{\varepsilon^2}$. Conditioning on this event not occurring, we have that

$$\left|d_R(S_n) - B_n^R\right| = \left|1 - \frac{|S_n|}{np}\right| \cdot d_R(S_n) \leq \left|1 - \frac{|S_n|}{np}\right| \leq \frac{\varepsilon}{2}\,,$$

where the first inequality follows from the fact that densities (in this case, $d_R(S_n)$) are always bounded from above by one, and the second inequality follows from our conditioning. This completes the proof of the second inequality in (2). □

The proof of Claim 4.2 is given next.

Proof of Claim 4.2. We first show that $(Z_0^R, Z_1^R, \ldots, Z_n^R)$ is a martingale. Fix $1 \leq i \leq n$, and suppose that the first $i-1$ rounds of $\mathsf{AdaptiveGame}_\varepsilon$ have just ended (so the values of $Z_0^R, \ldots, Z_{i-1}^R$ are already fixed), and that Adversary now picks an element x_i to submit in round i of the game.

If $x_i \notin R$ then $A_i^R = A_{i-1}^R$ and $B_i^R = B_{i-1}^R$ and so $Z_i^R = Z_{i-1}^R$, which trivially means that $\mathbb{E}\left[Z_i^R \mid Z_0^R, \ldots, Z_{i-1}^R\,;\, x_i \notin R\right] = Z_{i-1}^R$ as desired.

When $x_i \in R$, we have

$$A_i^R = A_{i-1}^R + \frac{1}{n} \quad ; \quad B_i^R = \begin{cases} B_{i-1}^R & \text{if } x_i \text{ is not sampled.} \\ B_{i-1}^R + \frac{1}{np} & \text{if } x_i \text{ is sampled.} \end{cases}$$

$$\Rightarrow \quad Z_i^R = \begin{cases} Z_{i-1}^R - 1/n & \text{if } x_i \text{ is not sampled.} \\ Z_{i-1}^R + 1/np - 1/n & \text{if } x_i \text{ is sampled.} \end{cases}$$

Recall that Sampler uses Bernoulli sampling with probability p, that is, x_i is sampled with probability p (regardless of the outcome of the previous rounds). Therefore, we have that

$$\mathbb{E}\left[Z_i^R \mid Z_0^R, \ldots, Z_{i-1}^R\,;\, x_i \in R\right] = Z_{i-1}^R + p \cdot (\frac{1}{np} - \frac{1}{n}) + (1-p) \cdot (-\frac{1}{n}) = Z_{i-1}^R.$$

The analysis of both cases $x_i \notin R$ and $x_i \in R$ implies that $E[Z_i^R | Z_0^R, \ldots, Z_{i-1}^R] = Z_{i-1}^R$, as desired.

We now turn to prove the other two statements of Claim 4.2. The maximum of the expression $|Z_i^R - Z_{i-1}^R|$ is $\max\{\frac{1}{n}, \frac{1}{np} - \frac{1}{n}\} \leq \frac{1}{np}$, obtained when $x_i \in R$. The variance of Z_i^R given $Z_0^R, \ldots, Z_{i-1}^R$ is zero given the additional assumption that $x_i \notin R$; assuming that $x_i \in R$, the variance satisfies

$$\mathsf{Var}(Z_i^R \mid Z_0^R, \ldots, Z_{i-1}^R\,;\, x_i \in R) = (1-p) \cdot \left(\frac{1}{n}\right)^2 + p \cdot \left(\frac{1}{np} - \frac{1}{n}\right)^2 = \frac{1}{n^2}\left(\frac{1}{p} - 1\right) \leq \frac{1}{n^2 p}.$$

Combining both cases, we conclude that $\mathsf{Var}(Z_i^R \mid Z_0^R, \ldots, Z_{i-1}^R) \leq \frac{1}{n^2 p}$, completing the proof. □

4.2 The Reservoir Sampling Case

We continue to the proof of the second statement of Lemma 4.1, which considers reservoir sampling. In high level, the proof goes along the same lines, except that we work with a different martingale. Specifically, for $k < i \leq n$ we define

$$A_i^R = i \cdot d_R(X_i) = |R \cap X_i|,$$
$$B_i^R = i \cdot d_R(S_i) = \frac{i}{k} \cdot |R \cap S_i|,$$
$$Z_i^R = B_i^R - A_i^R,$$

whereas for $i \leq k$ we simply define $A_i^R = B_i^R = |R \cap X_i|$. (This is a natural extension of the definition for $i > k$; specifically, in view of the definition of B_i^R, note that as long as no more than k elements appear in the stream, the reservoir simply keeps all of the stream's elements.)

The following claim is the analogue of Claim 4.2 for the setting of reservoir sampling.

Claim 4.3. *The sequence $(Z_0^R, Z_1^R, \ldots, Z_n^R)$ is a martingale. Furthermore, the variance of Z_i^R conditioned on $Z_0^R, \ldots, Z_{i-1}^R$ is bounded by i/k, and it always holds that $|Z_i^R - Z_{i-1}^R| \leq i/k$.*

Proof. We follow the same kind of analysis as in Claim 4.2. Fix $i > k$ (for $i \leq k$ the claim holds trivially), and suppose that the first $i-1$ rounds have ended, so $Z_0^R, \ldots, Z_{i-1}^R$ are already fixed. Denote the next element that the adversary submits by x_i. First, it is easy to verify that

$$A_i^R = \begin{cases} A_{i-1}^R & x_i \notin R \\ A_{i-1}^R + 1 & x_i \in R \end{cases}$$

The calculation of B_i^R requires a more subtle case analysis. Given $B_0^R, \ldots, B_{i-1}^R$ and x_i, the value of B_i^R is determined by three factors: (i) is $x_i \in R$ or not? (ii) is x_i sampled or not? and (iii) conditioning on x_i being sampled, does it replace an element from R in the sample, or an element not in R? We separate the analysis into several cases; in cases where x_i is sampled, we denote the element removed from the sample to make room for x_i by r_i.

Case 1: $x_i \notin R$. In the cases where x_i is either not sampled, or sampled but with $r_i \notin R$, elements from R are neither added nor removed from the sample. That is, $R \cap S_i = R \cap S_{i-1}$. Hence,

$$B_i^R = \frac{i}{k} \cdot |R \cap S_i| = \frac{i-1}{k} \cdot |R \cap S_{i-1}| + \frac{1}{k} \cdot |R \cap S_{i-1}| = B_{i-1}^R + d_R(S_{i-1}),$$

where the first equality is by definition, and the third equality follows again by definition and since $|S_{i-1}| = k$ for $i > k$.

It remains to consider the event where x_i is sampled and $r_i \in R$. The probability that x_i is sampled equals k/i, and conditioning on this occurring, the probability that r_i belongs to R is $d_R(S_{i-1})$, so the above event holds with probability $(k/i) \cdot d_R(S_{i-1})$. In this case, one element from R is removed from the sample, that is, $|R \cap S_i| = |R \cap S_{i-1}| - 1$, and therefore

$$B_i^R = \frac{i}{k} \cdot |R \cap S_i| = \frac{i}{k} \cdot |R \cap S_{i-1}| - \frac{i}{k} = B_{i-1}^R + d_R(S_{i-1}) - \frac{i}{k}.$$

Thus, conditioned on $x_i \notin R$, the expectation of B_i^R is

$$\left(1 - \frac{k}{i} \cdot d_R(S_{i-1})\right) \cdot \left(B_{i-1}^R + d_R(S_{i-1})\right) + \frac{k}{i} \cdot d_R(S_{i-1}) \cdot \left(B_{i-1}^R + d_R(S_{i-1}) - \frac{i}{k}\right) = B_{i-1}^R.$$

Since $A_i^R = A_{i-1}^R$ when $x_i \notin R$, we deduce that

$$E[Z_i^R | Z_0^R, \ldots, Z_{i-1}^R \,;\, x_i \notin R] = Z_{i-1}^R.$$

Case 2: $x_i \in R$**.** Similarly, whenever $S_i = S_{i-1}$ we have that $B_i^R = B_{i-1}^R + d_R(S_{i-1})$. The only case where this does not hold is when x_i is sampled and $r_i \notin R$, which has probability $(k/i) \cdot (1 - d_R(S_{i-1}))$. In this case, $|R \cap S_i| = |R \cap S_{i-1}| + 1$, implying that

$$B_i^R = \frac{i}{k} \cdot |R \cap S_i| = \frac{i}{k} \cdot |R \cap S_{i-1}| + \frac{i}{k} = B_{i-1}^R + d_R(S_{i-1}) + \frac{i}{k}.$$

Combining these two we get, conditioned on $x_i \in R$, that the expectation of B_i^R is

$$B_{i-1}^R + d_R(S_{i-1}) + \left(\frac{k}{i} \cdot (1 - d_R(S_{i-1}))\right) \cdot \frac{i}{k} = B_{i-1}^R + 1.$$

Finally, since $A_i^R = A_{i-1}^R + 1$ when $x_i \in R$, we have that

$$E[Z_i^R | Z_0^R, \ldots, Z_{i-1}^R \; ; \; x_i \in R] = Z_{i-1}^R.$$

The analysis of these two cases implies that $(Z_0^R, \ldots, Z_n^R)$ is indeed a martingale.

It remains to obtain the bounds on the difference $|Z_i^R - Z_{i-1}^R|$ and the variance of Z_i^R given $Z_0^R, \ldots, Z_{i-1}^R$. This follows rather easily as a byproduct of the above analysis (and the fact that the density d_R is always bounded between zero and one). When $x_i \notin R$, we know from the analysis that $A_i^R = A_{i-1}^R$ and $B_{i-1}^R - i/k \leq B_i^R \leq B_{i-1}^R + 1$, whereas if $x_i \in R$, we have $A_i^R = A_{i-1}^R + 1$ and $B_{i-1}^R \leq B_i^R \leq B_{i-1}^R + 1 + i/k$. In both cases, we conclude that $|Z_i^R - Z_{i-1}^R| \leq i/k$.

We next bound the variance of Z_i^R conditioned on the values of $Z_0^R, \ldots, Z_{i-1}^R$ (the analysis also implicitly conditions on the value $d_R(S_{i-1})$; the bound we shall eventually derive holds regardless of this value). We start with the case that $x_i \notin R$, and revisit Case 1 above: with probability $(k/i) \cdot d_R(S_{i-1})$, the value of Z_i^R is smaller than its expectation by $i/k - d_R(S_{i-1})$; and otherwise (with probability $1 - (k/i) \cdot d_R(S_{i-1})$), the value of Z_i^R is larger than its expectation by $d_R(S_{i-1})$. Thus, we have that

$$\begin{aligned}
&\mathsf{Var}(Z_i^R \mid Z_0^R, \ldots, Z_{i-1}^R, \; x_i \notin R, \; d_R(S_{i-1})) \\
&\quad = \frac{k}{i} \cdot d_R(S_{i-1}) \cdot \left(\frac{i}{k} - d_R(S_{i-1})\right)^2 + \left(1 - \frac{k}{i} \cdot d_R(S_{i-1})\right) \cdot (d_R(S_{i-1}))^2 \\
&\quad = \frac{i}{k} \cdot d_R(S_{i-1}) - (d_R(S_{i-1}))^2 \leq \frac{i}{k}.
\end{aligned}$$

We next address the case where $x_i \in R$, which correspond to Case 2 above. Here, with probability $(k/i) \cdot (1 - d_R(S_{i-1}))$, the value of Z_i^R is larger than its conditional expectation by $i/k + d_R(S_{i-1}) - 1$; otherwise, Z_i^R is smaller than the expectation by $1 - d_R(S_{i-1})$. Thus,

$$\begin{aligned}
&\mathsf{Var}(Z_i^R \mid Z_0^R, \ldots, Z_{i-1}^R, \; x_i \in R, \; d_R(S_{i-1})) \\
&\quad = \frac{k}{i} \cdot (1 - d_R(S_{i-1})) \cdot \left(\frac{i}{k} + d_R(S_{i-1}) - 1\right)^2 + \left(1 - \frac{k}{i} \cdot (1 - d_R(S_{i-1}))\right) \cdot (1 - d_R(S_{i-1}))^2 \\
&\quad = \frac{i}{k} \cdot (1 - d_R(S_{i-1})) - (1 - d_R(S_{i-1}))^2 \leq \frac{i}{k}.
\end{aligned}$$

As the conditional variance is always bounded by i/k, the bound remains intact if we remove the conditioning on the value of $d_R(S_{i-1})$ and the predicate assessing whether $x_i \in R$ or not. In other words, $\mathsf{Var}(Z_i^R | Z_0^R, \ldots, Z_{i-1}^R) \leq i/k$, completing the proof. □

The proof of the second part of Lemma 4.1 now follows from the last claim.

Proof of Lemma 4.1, reservoir sampling case. Observe that

$$\begin{aligned}\Pr(|d_R(X) - d_R(S)| \geq \varepsilon) &= \Pr(|B_n^R - A_n^R| \geq \varepsilon n) \\ &= \Pr(|Z_n^R - Z_0^R| \geq \varepsilon n).\end{aligned}$$

In view of Claim 4.3, we apply Lemma 3.3 on the martingale $Z^R = (Z_0^R, \ldots, Z_n^R)$ with $\lambda = \varepsilon n$, $\sigma_i^2 = i/k$ for any $i \geq k$ (for $i \leq k$, we can set $\sigma_i^2 = 0$), and $M = n/k$. We get that

$$\begin{aligned}\Pr(|Z_n^R - Z_0^R| \geq \lambda) &\leq 2\exp\left(-\frac{\lambda^2}{2\sum_{i=1}^n \sigma_i^2 + M\lambda/3}\right) \\ &= 2\exp\left(-\frac{\varepsilon^2 n^2}{2\sum_{i=1}^n (i/k) + (n/k)\cdot \varepsilon n/3}\right) \\ &= 2\exp\left(-\frac{\varepsilon^2 k n^2}{n(n+1) + \varepsilon n^2/3}\right) \\ &\leq 2\exp\left(-\frac{\varepsilon^2 k n^2}{2n^2}\right) = 2\exp\left(-\frac{\varepsilon^2 k}{2}\right),\end{aligned}$$

where the second inequality holds for $n \geq 2$. Therefore, it suffices to require $k \geq \frac{2}{\varepsilon^2}\ln\left(\frac{2}{\delta}\right)$ to get the bound $\Pr(|d_R(X) - d_R(S)| \geq \varepsilon) \leq \delta$. □

5 An Adaptive Attack on Sampling

In this section, we present our lower bounds. Specifically, we show that the sample size cannot depend solely on the VC-dimension, but rather that the dependency on the cardinality is necessary. This is done by describing a set system $(U, \mathcal{R})$ with large $|U|$ and VC-dimension of one, together with a strategy for the adversary that will make the sampled set unrepresentative with respect to $(U, \mathcal{R})$. That is, the sampled set will not be an ε-approximation of $(U, \mathcal{R})$ with high probability. This is in contrast to the static setting where the same sample size suffices to an ε-approximation with high probability. Moreover, in the case of the BernoulliSample algorithm, the sampled set under attack is extremely unrepresentative, consisting precisely of the k smallest elements in the stream (where k is the total sample size at the end of the stream).

Proof of Theorem 1.3. Set the universe to be the well-ordered set $U = \{1, 2, \ldots, N\}$ for an arbitrary $n^{6\ln n} \leq N \leq 2^{n/2}$ and let $\mathcal{R} = \{[1, b] : b \in U\}$. Clearly, $(U, \mathcal{R})$ has VC-dimension 1. Adversary's strategy (for both sampling algorithms BernoulliSample and ReservoirSample) is described in Figure 3.

Let S denote the subsequence of elements sampled by the algorithm BernoulliSample along the stream. The expected size of S is $np \leq np'$, and it follows from the well-known Markov inequality (see e.g. [AS16], Appendix A) that $\Pr(|S| \geq 2np') < 1/2$ (in fact the probability is much smaller, by Chernoff inequality, but we will not need the stronger bound). From here on, we condition on the complementary event: we assume that $|S| < 2np'$. The next claim asserts that for S of this size, Adversary's strategy does not fail, in the sense that it never runs out of elements (i.e., $a_i < b_i$ for all $i \in [n]$).

The adversarial strategy

1. Set $a_1 = 1$ and $b_1 = N$.
2. Let $p' = \max\{p, \ln n/n\}$.
3. For $i = 1 \dots n$ do:
   (a) Set $x_i = \lfloor a_i + (1-p')(b_i - a_i) \rfloor$.
   (b) If x_i is sampled then set $a_{i+1} = x_i$ and $b_{i+1} = b_i$.
   (c) Otherwise set $a_{i+1} = a_i$ and $b_{i+1} = x_i$.
4. The final stream is $X = x_1, \dots, x_n$.

Figure 3: The description of *Adversary*'s strategy for making the sample unrepresentative.

Claim 5.1. *If $|S| < 2np'$ then $b_i - a_i \geq n$ for any $i \in [n]$.*

Proof. For any $i \in [n]$, set $\ell_i = b_i - a_i$. We prove by induction that $\ell_i \geq n$. If x_i is sampled, then we have that $\ell_{i+1} \geq p'\ell_i$ and otherwise we have that $\ell_{i+1} = (1-p')\ell_i - 2 \geq (1-2p')\ell_i$, where the inequality follows from the induction assumption. Since $|S| < 2np'$, we get that

$$\begin{aligned}
\ell_i &\geq p'^{|S|}(1-2p')^{n-|S|} \cdot N \\
&\geq p'^{|S|}(1-2p')^{n} \cdot N \\
&= e^{-(|S| \ln \frac{1}{p'} + n \ln(\frac{1}{1-2p'}))} \cdot N \\
&> e^{-(|S| \ln \frac{1}{p'} + 3np')} \cdot N \\
&> e^{-(2np' \ln \frac{1}{p'} + 3np')} \cdot N \\
&\geq e^{\ln n - \ln N} \cdot N = n \,,
\end{aligned}$$

where the third inequality holds since $3p \geq \ln(\frac{1}{1-2p})$ for small enough $p > 0$, and the last inequality follows since $p' \leq \frac{\ln N}{6n \ln n}$ and $p' \geq \ln n/n$, which means that

$$\ln N \geq 6np' \ln n \geq 2np' \ln(1/p') + 3np' + \ln n.$$

This proves the induction step, and completes the proof of the claim. □

The last claim means that if $|S| < 2np'$, then the attack in Figure 3 successfully generates a stream of n elements. We now show that the sampled set is not an ε-approximation. We begin by analyzing the BernoulliSample algorithm.

Claim 5.2. *Consider* Adversary*'s attack on* BernoulliSample *described in Figure 3. At round i of the game,*

- *All elements that were previously submitted by* Adversary *and* sampled *are no bigger than a_i.*
- *All elements that were previously submitted but* not sampled *are no smaller than b_i.*

- *The element submitted during round i is between a_i and b_i.*

Proof. By induction, where the base case $i = 1$ is trivial. Suppose that the claim holds for the first $i-1$ rounds; we now prove it for round i. By definition of the attack, and from Claim 5.1 it holds that $a_{i-1} \leq a_i < b_i \leq b_{i-1}$ and so any of the elements x_j for $j < i-1$ satisfies the desired condition, by the induction assumption. It remains to address the case where $j = i-1$. If x_{i-1} was sampled, then the attack sets $a_i = x_{i-1}$, that is, x_{i-1} is a sampled element and satisfies $x_{i-1} \leq a_i$. Otherwise, the attack sets $b_i = x_{i-1}$ and so x_{i-1} is a non-sampled element and satisfies $x_{i-1} \geq b_i$. Finally, $a_i < x_i < b_i$ always holds. Thus, the three desired conditions are retained. □

As the last claim depicts, all *sampled* elements are smaller than all *non-sampled* ones at any point along the stream. This, of course, suffices for the sampled set to not be an ε-approximation of $(U, \mathcal{R})$. Denote the sampled set by S, and let s be the maximal element in S (if S is empty, we are done). Consider now the range $[1, s] \in \mathcal{R}$: its density in the sampled set is 1, namely, $d_{[1,s]}(S) = 1$, while its density in the stream is $d_{[1,s]}(X) = |S|/n$. To summarize,

$$|d_{[1,s]}(S) - d_{[1,s]}(X)| \geq 1 - |S|/n \geq 1 - 2p' > 1/2 \geq \varepsilon \,.$$

Altogether, the attack does not fail provided that $|S| < 2np'$, which holds with probability at least $1/2$. Thus, BernoulliSample with parameter p as in the theorem's statement is not (ε, δ)-robust.

The analysis of the ReservoirSample algorithm is very similar. Recall that k denotes the sample size, and let k' be the total number of elements that were sampled during the reservoir sampling process. That is, k' counts sampled elements that were evicted at a future iteration. We bound k' as follows. $\mathbb{E}[k'] = k + \sum_{i=1}^{n} k/n \leq 2k \ln n$. Again, Markov inequality shows that with probability at least $1/2$, we will have $k' \leq 4k \ln n$. Using the previous analysis, we know that all k' elements are the smallest elements in the stream. The sample set S consists of some k elements among these k' elements (in other words, the sample set is not necessarily the set of k smallest element, but it is still a subset of the k' smallest elements). Thus, taking the interval $[1, s]$ where s is the maximal element among the k' elements, we have that the density of $[1, s]$ in the sample is $d_{[1,s]}(S) = \frac{k}{k} = 1$. On the other hand, the density of $[1, s]$ is the stream is

$$d_{[1,s]}(X) = \frac{k'}{n} \leq \frac{4k \ln n}{n} \leq \frac{\ln N}{n} \leq 1/2.$$

Together, we entail that

$$|d_{[1,s]}(S) - d_{[1,s]}(X)| > 1 - 1/2 \geq \varepsilon \,,$$

meaning that ReservoirSample with k as in the statement of the theorem is not (ε, δ)-robust. □

6 Continuous Robustness

In this section, we prove that the ReservoirSample algorithm is (ε, δ)-*continuous* robust against static and adaptive adversaries. Recall that a sampling algorithm is (ε, δ)-continuously robust if the following holds with probability at least $1 - \delta$: at *any* point throughout the stream, the current sample held by Sampler is an ε-approximation of the current stream (i.e., of the set of all elements submitted by Adversary until now).

With this definition in hand, BernoulliSample cannot possibly be continuously robust in general (even in the static setting)[4]. We thus restrict our discussion to ReservoirSample from here on, and turn to the proof of Theorem 1.4. The proof examines $O(\varepsilon^{-1} \ln n)$ carefully picked points along the stream, applying Theorem 1.2 on each of the points. It then shows that if the sample is a good approximation of the stream at all of these points, then continuous robustness is guaranteed with high probability.

Proof of Theorem 1.4. We provide the proof for the setting of an *adaptive* adversary. The proof for the static setting is essentially identical, with the only difference being that, instead of making black-box applications of Theorem 1.2, we apply the static analogue of it; Recall that the bound in the static analogue is of the form $\Theta\left(\frac{d+\ln 1/\delta}{\varepsilon^2}\right)$, compared to the $\Theta\left(\frac{\ln|\mathcal{R}|+\ln 1/\delta}{\varepsilon^2}\right)$ bound appearing in the statement of Theorem 1.2.

Let $(U, \mathcal{R})$, n, ε, δ be as in the statement of the theorem. As a warmup, let us analyze a simple yet non-optimal proof based on a naïve union bound. Denote the stream and sample after i rounds by X_i and S_i, respectively. Consider for a moment the first i rounds of the game as a "standalone" game where the stream length is i. Applying the second part of Theorem 1.2 with parameters $(U, \mathcal{R}), \varepsilon, \delta', i$, where $\delta' = \delta/n$, we get that if the memory size k of ReservoirSample satisfies

$$k \geq 2 \cdot \frac{\ln|\mathcal{R}| + \ln(2n/\delta)}{\varepsilon^2} = 2 \cdot \frac{\ln|\mathcal{R}| + \ln(2/\delta) + \ln n}{\varepsilon^2}, \tag{3}$$

then regardless of Adversary's strategy,

$$\Pr(S_i \text{ is not an } \varepsilon\text{-approximation of } X_i) \leq \delta/n.$$

Taking a union bound, the probability that S_i is an ε-approximation of X_i for all $i \in [n]$ is at least $1 - n \cdot (\delta/n) = 1 - \delta$. Thus, it follows that ReservoirSample whose parameter k satisfies the condition of (3) is (ε, δ)-continuously robust.

We now continue to the proof of the improved bound, appearing in the statement of the theorem. The proof is also based, at its core, on a union bound argument, albeit a more efficient one. The key idea is to take a sparse set of "checkpoints" $i_1, \ldots, i_t$ along the stream, where $i_{j+1} = (1 + \Theta(\varepsilon))i_j$, apply Theorem 1.2 at any of the times $i_1, \ldots, i_t$ to make sure the sample is an $(\varepsilon/2)$-approximation of the stream in any of these times. Finally, we show that with high probability, for any $j \in [t-1]$, the approximation is preserved (the approximation factor might become slightly worse, but no worse than ε) in the "gaps" between any couple of such neighboring points.

For this, we first need the following simple claims.

Claim 6.1. *Let T, T' be two sequences of length k over U, which differ in up to v values. Then $|d_R(T) - d_R(T')| \leq v/k$ for any $R \subseteq U$. In particular, if T is an α-approximation of some sequence $X \supseteq T, T'$, then T' is an $(\alpha + v/k)$-approximation of X.*

Proof. For any subset $R \subseteq U$ we have $-v \leq |R \cap T| - |R \cap T'| \leq v$. Dividing by $k = |T| = |T'|$, and recalling that $d_R(T) = |R \cap T|/|T|$ and $d_R(T') = |R \cap T'|/|T'|$, we conclude that $-v/k \leq$

[4] To see this, consider any set system $(U, \mathcal{R})$ where $\mathcal{R}$ contains a singleton $\{u\}$ for some $u \in U$, which is the first element of the stream. With probability $1 - p$ this element is not sampled and the density of $\{u\}$ in the sample at the current point is 0, while its density in the stream is 1. This violates the ε-approximation requirement (unless $p \geq 1 - \delta$).

$d_R(T) - d_R(T') \leq v/k$, that is, $|d_R(T) - d_R(T')| \leq v/k$. To prove the second part, note that

$$|d_R(T') - d_R(X)| \leq |d_R(T') - d_R(T)| + |d_R(T) - d_R(X)| \leq v/k + \alpha$$

for any $R \subseteq U$. □

Claim 6.2. *Suppose that $T \subseteq X \subseteq X'$ are three sequences over U, where T is an α-approximation of X, and $|X'| \leq (1+\beta)|X|$. Then T is an $(\alpha+\beta)$-approximation of X'.*

Proof. For any subset $R \subseteq U$, we have that $|R \cap X| \leq |R \cap X'| \leq |R \cap X| + \beta|X|$. We also know that $|d_R(T) - d_R(X)| \leq \alpha$, since T is an α-approximation of X. On the one hand, it follows that

$$\begin{aligned} d_R(T) &\geq d_R(X) - \alpha = \frac{|R \cap X|}{|X|} - \alpha \geq \frac{|R \cap X'| - \beta|X|}{|X|} - \alpha \\ &\geq \frac{|R \cap X'|}{|X'|} - \beta - \alpha = d_R(X') - (\alpha + \beta). \end{aligned}$$

On the other hand,

$$\begin{aligned} d_R(T) &\leq d_R(X) + \alpha = \frac{|R \cap X|}{|X|} + \alpha \leq \frac{|R \cap X'|}{|X'|/(1+\beta)} + \alpha \\ &= (1+\beta)d_R(X') + \alpha \leq d_R(X') + (\alpha + \beta). \end{aligned}$$

As these inequalities hold for any $R \subseteq U$, the claim follows. □

As a consequence of the above two claims, we get the following useful claim. (Recall that for any $i \in [n]$, the sample and stream after i rounds are denoted by S_i and X_i, respectively.)

Claim 6.3. *Consider* ReservoirSample *with memory size k, and suppose that exactly v elements were sampled in rounds $l+1, l+2, \ldots, m$ of the game, where $k \leq l < m \leq (1+\beta)l$. If S_l is an α-approximation of X_l, then S_m is an $(\alpha + \beta + v/k)$-approximation of X_m.*

Proof. By Claim 6.2, S_l is an $(\alpha+\beta)$-approximation of X_m. As S_m differs from S_l by at most v elements, we conclude from Claim 6.1 that S_m is an $(\alpha + \beta + v/k)$-approximation of X_m. □

The last claim equips us with an approach to ensure continuous robustness, which is more efficient compared to the simple union bound approach. Suppose that there exists a set of integers $k = i_1 < i_2 < \ldots < i_t = n$ satisfying the following for any $j \in [t-1]$.

1. S_{i_j} is an α-approximation of X_{i_j}, where $\alpha = \varepsilon/4$.
2. $i_{j+1} \leq (1+\beta)i_j$, where $\beta = \varepsilon/4$.
3. The number of elements sampled in rounds $i_j + 1, i_j + 2, \ldots, i_{j+1}$ is bounded by $v = \varepsilon k/2$.

We claim that the above three conditions suffice to ensure that S_i is an ε-approximation of X_i for any $i \in [n]$. Indeed, for $i \leq k$, $S_i = X_i$ is trivially an ε-approximation. When $i > k$, consider the maximum $j < t$ for which $i_j \leq i$, and apply Claim 6.3 with $l = i_j$, $m = i$, and α, β, v as dictated above. Since $\alpha + \beta + v/k = \varepsilon$, the claim implies that S_i is an ε-approximation of X_i, as desired.

Specifically, given k satisfying the assumption of Theorem 1.4, we pick $i_1, i_2, \ldots, i_t$ recursively as follows: we start with $i_1 = k$; and given i_j we set $i_{j+1} \leq n$ as the largest integer satisfying that

$i_{j+1} \leq (1+\beta)i_j = (1+\varepsilon/4)i_j$. It is not hard to verify that $i_j = k \cdot (1+\theta(\varepsilon))^{j-1}$ (this implicitly relies on the fact that $k \geq 4/\varepsilon$, ensured by the assumption of the theorem). Note that $t = O(\ln_{1+\varepsilon} n) = O(\varepsilon^{-1} \ln n)$. We next show that for this choice of $i_1, \ldots, i_t$, the above three conditions are satisfied simultaneously for all $j \in [t-1]$ with probability at least $1-\delta$. This shall conclude the proof.

For the first condition, apply Theorem 1.2 for any $j \in [t-1]$ with parameters $(U, \mathcal{R}), \varepsilon/42, \delta', i_j$ where $\delta' = \delta/2t$, concluding that if the memory size k satisfies

$$k \geq 2 \cdot \frac{\ln|\mathcal{R}| + \ln(4t/\delta)}{(\varepsilon/4)^2} = \Theta\left(\frac{\ln|\mathcal{R}| + \ln(1/\delta) + \ln(1/\varepsilon) + \ln\ln n}{\varepsilon^2}\right)$$

then for any $j \in [t-1]$,

$$\Pr(S_{i_j} \text{ is an } \varepsilon/2\text{-approximation of } X_{i_j}) \geq 1 - \delta/2t.$$

Taking a union bound, with probability at least $1-\delta/2$ the first condition holds for all $j \in [t-1]$.

The second condition, regarding the boundedness of i_{j+1} as a function of i_j, holds trivially (and deterministically) for our choice of $i_1 \leq i_2 \leq \ldots \leq i_t$.

Finally, it remains to address the third condition. For any $j \in [t-1]$, let A_j denote the total number of sampled elements in rounds $i_j+1, i_j+2, \ldots, i_{j+1}$ of the game. Note that each such A_j is a random variable. We wish to show that

$$\Pr(A_j > \varepsilon k/2) \leq \delta/2t. \tag{4}$$

Indeed, if (4) is true for any $j \in [t-1]$, then the probability that the third condition holds for any j is at least $1-\delta/2$, which (in combination with our analysis of the other two conditions) completes the proof. Thus, it remains to prove (4).

Recall that the probability of an element to be sampled in round i is exactly k/i, and that $i_{j+1} \leq (1+\varepsilon/4)i_j$. Hence, A_j is a sum of up to $\lfloor \varepsilon i_j/4 \rfloor$ independent random variables, each of which has probability less than k/i_j to be sampled. In particular, the mean of A_j is less than $(\varepsilon i_j/4) \cdot (k/i_j) = \varepsilon k/4$. From Chernoff bound (Theorem 3.1), we get the desired bound:

$$\Pr(A_j \geq \varepsilon k/2) < \exp\left(-\frac{2^2 \cdot \varepsilon k/4}{2 + 2 \cdot 2/3}\right) \leq \exp\left(-\frac{\varepsilon k}{4}\right) \leq \frac{\delta}{2t},$$

where the last inequality holds for $k \geq c \cdot \varepsilon^{-1}(\ln 1/\delta + \ln 1/\varepsilon + \ln\ln n)$, for a sufficiently large constant $c > 0$; note that k in the theorem's statement indeed satisfies this inequality. □

Acknowledgments

We are grateful to Moni Naor for suggesting the study of streaming algorithms in the adversarial setting and for helpful and informative discussions about it. We additionally thank Noga Alon, Nati Linial, and Ohad Shamir for invaluable comments and suggestions for the paper.

References

[AS16] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley Publishing, 4th edition, 2016.

[BCEG07] Amitabha Bagchi, Amitabh Chaudhary, David Eppstein, and Michael T. Goodrich. Deterministic sampling and range counting in geometric data streams. *ACM Transactions on Algorithms*, 3(2):16, 2007.

[BCM+13] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD*, pages 387–402, 2013.

[BOV15] Vladimir Braverman, Rafail Ostrovsky, and Gregory Vorsanger. Weighted sampling without replacement from data streams. *Information Processing Letters*, 115(12):923–926, 2015.

[BR18] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[CBM18] Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. PAC-learning in the presence of evasion adversaries. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS, pages 228–239, 2018.

[CDK+11] Edith Cohen, Nick G. Duffield, Haim Kaplan, Carsten Lund, and Mikkel Thorup. Efficient stream sampling for variance-optimal estimation of subset sums. *SIAM Journal on Computing*, 40(5):1402–1431, 2011.

[CEM+96] Kenneth L. Clarkson, David Eppstein, Gary L. Miller, Carl Sturtivant, and Shang-Hua Teng. Approximating center points with iterative radon points. *International Journal of Computational Geometry and Applications*, 6(3):357–377, 1996.

[CG05] Graham Cormode and Minos N. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In *Proceedings of the 31st International Conference on Very Large Data Bases, VLDB*, pages 13–24, 2005.

[CGP+18] Timothy Chu, Yu Gao, Richard Peng, Sushant Sachdeva, Saurabh Sawlani, and Junxing Wang. Graph sparsification, spectral sketches, and faster resistance computation, via short cycle decompositions. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 361–372, 2018.

[Cha01] Bernard Chazelle. *The discrepancy method - randomness and complexity*. Cambridge University Press, 2001.

[Che52] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[CJSS03] Charles D. Cranor, Theodore Johnson, Oliver Spatscheck, and Vladislav Shkapenyuk. The gigascope stream database. *IEEE Data Engineering Bulletin*, 26(1):27–32, 2003.

[CL06] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79–127, 2006.

[CMY11] Graham Cormode, S. Muthukrishnan, and Ke Yi. Algorithms for distributed functional monitoring. *ACM Transactions on Algorithms*, 7(2):21:1–21:20, 2011.

[CMYZ12] Graham Cormode, S. Muthukrishnan, Ke Yi, and Qin Zhang. Continuous sampling from distributed streams. *Journal of the ACM*, 59:10:1–10:25, 2012.

[CTW16] Yung-Yu Chung, Srikanta Tirthapura, and David P. Woodruff. A simple message-optimal algorithm for random sampling from a distributed stream. *IEEE Transactions on Knowledge and Data Engineering*, 28:1356–1368, 2016.

[DFH+15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[DGS15] Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, pages 1405–1411, 2015.

[DLT05] Nick G. Duffield, Carsten Lund, and Mikkel Thorup. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Transactions on Networking*, 13(5):933–946, 2005.

[ES06] Pavlos S. Efraimidis and Paul G. Spirakis. Weighted random sampling with a reservoir. *Information Processing Letters*, 97(5):181–185, 2006.

[Fre75] David A. Freedman. On tail probabilities for martingales. *The Annals of Probability*, 3(1):100–118, 1975.

[GDG+17] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[GHR+12] Anna C. Gilbert, Brett Hemenway, Atri Rudra, Martin J. Strauss, and Mary Wootters. Recovering simple signals. In *Information Theory and Applications Workshop, ITA*, pages 382–391, 2012.

[GHS+12] Anna C. Gilbert, Brett Hemenway, Martin J. Strauss, David P. Woodruff, and Mary Wootters. Reusable low-error compressive sampling schemes through privacy. In *IEEE Statistical Signal Processing Workshop, SSP*, pages 536–539, 2012.

[GK01] Michael Greenwald and Sanjeev Khanna. Space-efficient online computation of quantile summaries. *SIGMOD Record*, 30(2):58–66, 2001.

[GK16] Michael B. Greenwald and Sanjeev Khanna. Quantiles and equi-depth histograms over streams. In Minos Garofalakis, Johannes Gehrke, and Rajeev Rastogi, editors, *Data Stream Management: Processing High-Speed Data Streams*, pages 45–86. Springer Berlin Heidelberg, 2016.

[GLA16] Mohammed Ghesmoune, Mustapha Lebbah, and Hanene Azzag. State-of-the-art on clustering data streams. *Big Data Analytics*, 1(1):13, 2016.

[GMP18] Ian J. Goodfellow, Patrick D. McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7):56–66, 2018.

[HAK07] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

[HW13] Moritz Hardt and David P. Woodruff. How robust are linear sketches to adaptive inputs? In *Symposium on Theory of Computing Conference*, pages 121–130, 2013.

[JMR05] Theodore Johnson, S. Muthukrishnan, and Irina Rozenbaum. Sampling algorithms in a stream operator. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2005.

[JPA04] Chris Jermaine, Abhijit Pol, and Subramanian Arumugam. Online maintenance of very large random samples. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 299–310, 2004.

[JSTW19] Rajesh Jayaram, Gokarna Sharma, Srikanta Tirthapura, and David P. Woodruff. Weighted reservoir sampling from distributed streams. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS, pages 218–235, 2019.

[KLL16] Zohar Karnin, Kevin Lang, and Edo Liberty. Optimal quantile approximation in streams. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 71–78, 2016.

[Knu97] Donald E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[LLS01] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *Journal of Computer and System Sciences*, 62(3):516–527, 2001.

[LMPL18] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 114–122, 2018.

[LN93] Richard J. Lipton and Jeffrey F. Naughton. Clocked adversaries for hashing. *Algorithmica*, 9(3):239–252, 1993.

[McD98] Colin McDiarmid. Concentration. In Michel Habib, Colin McDiarmid, Jorge Ramirez-Alfonsin, and Bruce Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248. Springer Berlin Heidelberg, 1998.

[MHS19] Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory, COLT*, volume 99 of *Proceedings of Machine Learning Research*, pages 2512–2530, 2019.

[MM19] Saeed Mahloujifar and Mohammad Mahmoody. Can adversarially robust learning leverage computational hardness? In *Algorithmic Learning Theory, ALT*, pages 581–609, 2019.

[MNS11] Ilya Mironov, Moni Naor, and Gil Segev. Sketching in adversarial environments. *SIAM Journal on Computing*, 40(6):1845–1870, 2011.

[MRL99] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. *SIGMOD Record*, 28(2):251–262, 1999.

[MV17] Nabil H. Mustafa and Kasturi R. Varadarajan. Epsilon-approximations and epsilon-nets. In Csaba D. Toth, Joseph O'Rourke, and Jacob E. Goodman, editors, *Handbook of Discrete and Computational Geometry, 3rd Edition*, chapter 47, page 27. Chapman and Hall/CRC, New York, 2017.

[MW18] Michael McCoyd and David A. Wagner. Background class defense against adversarial examples. In *2018 IEEE Security and Privacy Workshops, SP Workshops*, pages 96–102, 2018.

[NY15] Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference*, pages 565–584, 2015.

[OV11] M. Tamer Özsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Springer Publishing Company, Incorporated, 3rd edition, 2011.

[PMG+17] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM Asia Conference on Computer and Communications Security, AsiaCCS*, pages 506–519, 2017.

[Sau72] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145 – 147, 1972.

[SBM+18] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. DARTS: deceiving autonomous cars with toxic signs. *CoRR*, abs/1802.06430, 2018.

[Sha12] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[She72] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

[SKYL17] Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size, 2017. Published as a conference paper at ICLR 2018.

[SS17] Ohad Shamir and Liran Szlak. Online learning with local permutations and delayed feedback. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, pages 3086–3094, 2017.

[SST10] Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems 23: Annual Conference on Neural Information Processing Systems, NIPS*, pages 2199–2207, 2010.

[STZ04] Subhash Suri, Csaba D. Tóth, and Yunhong Zhou. Range counting over multidimensional data streams. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, SCG '04, pages 160–169, 2004.

[SZS+14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR*, 2014.

[Tal94] Michel Talagrand. Sharper bounds for gaussian and empirical processes. *The Annals of Probability*, 22(1):28–76, 1994.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[VC71] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[Vit85] Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985.

[WFRS18] Blake E. Woodworth, Vitaly Feldman, Saharon Rosset, and Nati Srebro. The everlasting database: Statistical validity at a fair price. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 6532–6541, 2018.

[WLYC13] Lu Wang, Ge Luo, Ke Yi, and Graham Cormode. Quantiles over data streams: An experimental study. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 737–748, 2013.

[ZLZ18] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 1330–1340, 2018.