ROBUST ALGORITHMS ON ADAPTIVE INPUTS FROM BOUNDED ADVERSARIES

Anonymous authors
Paper under double-blind review

ABSTRACT

We study dynamic algorithms robust to adaptive inputs generated from sources with bounded capabilities, such as sparsity or limited interaction. For example, we consider robust linear algebraic algorithms when the updates to the inputs are sparse but given by an adversary with access to a query oracle. We also study robust algorithms in the standard centralized setting, where an adversary queries an algorithm in an adaptive manner, but the number of interactions between the adversary and the algorithm is bounded. Together, we provide a unified framework for answering Q adaptive queries that incurs $\widetilde{\mathcal{O}}(\sqrt{Q})$ overhead in space, which is roughly a quadratic improvement over the naïve implementation, and only incurs a logarithmic overhead in query time. Our general framework has diverse applications in machine learning and data science, such as adaptive distance estimation, kernel density estimation, linear regression, range queries, and point queries. Surprisingly, we show that these novel subroutines for each of these problems can be generally combined with the elegant use of differential privacy to hide the internal randomness of various subroutines, leading to robust algorithms across these different settings. In addition, we demonstrate even better algorithmic improvements for (1) reducing the pre-processing time for adaptive distance estimation and (2) permitting an unlimited number of adaptive queries for kernel density estimation.

1 INTRODUCTION

Robustness to adaptive inputs or adversarial attacks has recently emerged as an important desirable characteristic for algorithm design. An adversarial input can be created using knowledge of the model to induce incorrect outputs on widely used models, such as neural networks (Biggio et al., 2013; Szegedy et al., 2014; Goodfellow et al., 2015; Carlini & Wagner, 2017; Madry et al., 2018). Adversarial attacks against machine learning algorithms in practice have also been documented in applications such as network monitoring (Chandola et al., 2009), strategic classification (Hardt et al., 2016), and autonomous navigation (Papernot et al., 2016; Liu et al., 2017; Papernot et al., 2017). The need for sound theoretical understanding of adversarial robustness is also salient in situations where successive inputs to an algorithm can be possibly correlated; even if the input is not adversarially generated, a user may need to repeatedly interact with a mechanism in a way such that future updates may depend on the outcomes of previous interactions (Mironov et al., 2011; Gilbert et al., 2012; Bogunovic et al., 2017; Naor & Yogev, 2019; Avdiukhin et al., 2019). Motivated by both practical needs and a lack of theoretical understanding, there has been a recent flurry of theoretical studies of adversarial robustness. The streaming model of computation has especially received significant attention (Ben-Eliezer et al., 2021; Hassidim et al., 2020; Woodruff & Zhou, 2021; Kaplan et al., 2021; Braverman et al., 2021; Attias et al., 2021; Chakrabarti et al., 2022; Ajtai et al., 2022; Chakrabarti et al., 2022; Ben-Eliezer et al., 2022). More recently, there have also been a few initial results for dynamic algorithms on adaptive inputs for graph algorithms (Wajc, 2020; Beimel et al., 2021; Bernstein et al., 2022). These works explored the capabilities and limits of algorithms for adversaries that were freely able to choose the input based on previous outputs by the algorithm.

However, in many realistic settings, adversarial input is limited in its abilities. For example, adversarial attacks in machine learning are often permitted to only alter the "true" input by a small amount bounded in norm. For the L_0 norm, this restriction means that the adversary can only add a sparse noise to the true input. More generally, it seems reasonable to assume that adversarial input is

generated from a source that has bounded computation time or bounded interactions with an honest algorithm.

1.1 Our Contributions

In this paper, we study algorithms robust to adaptive/adversarial input generated from sources with bounded capabilities. We first study dynamic algorithms for adaptive inputs from a source that is restricted in sparsity, where the adversary can only alter a few elements. Namely, we consider robust linear algebraic algorithms in this setting. We then study robust algorithms in the standard centralized setting, where an adversary queries an algorithm in an adaptive manner, but the number of interactions between the adversary and the algorithm is bounded. Surprisingly, we show that combining novel subroutines for each of these problems in conjunction with a simple but elegant idea of using differential privacy to hide the internal randomness of various subroutines, e.g., Hassidim et al. (2020); Beimel et al. (2021) suffices to achieve robust algorithms across these different settings.

Dynamic algorithms on adaptive input for regression. Motivated by the problem of label shift in machine learning, we consider a dynamic version of least-squares regression, where the labels get updated. In this model, we are given a fixed design matrix and a target label that receives a sequence of updates. After each one, the algorithm is asked to output an estimate of the optimal least-squares objective. The goal of the algorithm is to maintain the objective value within a multiplicative factor $(1+\varepsilon)$ to the optimal.

More specifically, the algorithm is given a fixed design matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $n \geq d$ and an initial response vector (i.e., label) $\mathbf{b}^{(1)}$, which receives updates over time. We are interested in estimating the least-squares objective value $F(\mathbf{A}, \mathbf{b}) = \min_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ as the target label $\mathbf{b}$ undergoes updates. The updates to $\mathbf{b}$ are adaptively chosen by an adversary but can only affect at most K entries of $\mathbf{b}$ per step. Formally, on the i-th round:

(1) The adversary provides an update to K entries of $\mathbf{b}^{(i-1)}$, possibly depending on all previous outputs of the algorithm.

(2) The algorithm updates its data structure and outputs an estimate $\widehat{F}_i$ of $F_i = F\left(\mathbf{A}, \mathbf{b}^{(i)}\right)$.

(3) The adversary observes and records the output $\widehat{F}_i$.

The goal of the adversary is to create a sequence of labels $\left(\mathbf{b}^{(i)}\right)_{i=1}^{T}$ that induces the algorithm to output an inaccurate estimate. To deal with adaptivity, a naïve idea is to treat each step as an independent least-squares regression problem. However, this approach uses a completely new approximation of the objective value for each update, which seems potentially wasteful. On the other hand, any randomness that is shared by computations over multiple updates can potentially be leveraged by the adversary to induce an incorrect output.

Our main result is an algorithm that beats the naïve algorithm in this challenging, adaptively adversarial setting. We provide a general result with run-time dependence on n, d, K, and the number of nonzero entries in $\mathbf{A}$, $\mathrm{nnz}(\mathbf{A})$.

Theorem 1.1 (Informal; see Theorem 2.1)**.** *Let $\kappa(\mathbf{A}) = O(1)$ and $\varepsilon \in (0,1)$. There exists a dynamic algorithm that given adaptively chosen K-sparse updates to $\mathbf{b}$ and a fixed design matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, outputs a $(1+\varepsilon)$ approximation to the least-squares objective $F(\mathbf{A}, \mathbf{b}^{(i)})$ every round with high probability. The algorithm uses $\widetilde{\mathcal{O}}\left(\sqrt{K\,\mathrm{nnz}(\mathbf{A})}/\varepsilon^{2.5}\right)$ amortized time per step of update.*

Specifically, the update time is $d^{1.5}$ when $K \leq d$ and $n = O(d)$ and square root of the input sparsity when $K = O(1)$. Notice that this significantly betters the naïve approach of treating each step independently and solving for the least-square objective, which requires $\mathrm{nnz}(\mathbf{A}) + \mathrm{poly}(d)$ time by sketching (Woodruff (2014)).

We mention that a recent work by Jiang et al. (2022) considers a row-arrival model for dynamic linear regression. Our setting is different since we allow arbitrary updates to the target label, whereas in their setting the design matrix undertakes incremental change. We note that their algorithm maintains a solution vector, while we focus on the cost only.

Adaptive query framework. We then consider robust algorithms in the standard *centralized* setting, where an adversary queries an algorithm in an adaptive manner. In many key algorithmic applications, randomization is necessary to achieve fast query time and efficient storage. This necessitates the need for robust versions of these algorithm which can efficiently employ the power of randomness while also being accurate across multiple possibly correlated inputs. Our main parameters of interest are query time and the space used by a robust algorithm compared to their naïve, non-robust, counterparts.

Formally, we define the model as a two-player game between an algorithm HonestAlg over a data set X and an adversary $\mathcal{A}$ that makes adversarial queries about X to HonestAlg. At the beginning of the game, HonestAlg uses pre-processing time to compute a data structure $\mathcal{D}$ from X to answer future queries from $\mathcal{A}$. The game then proceeds in at most Q rounds for some predetermined Q, so that in the t-th round, where $t \in [Q]$:

(1) $\mathcal{A}$ computes a query q_t on X, which depends on all previous responses from HonestAlg.

(2) HonestAlg uses $\mathcal{D}$ to output a response d_t to query q_t.

(3) $\mathcal{A}$ observes and records the response d_t.

The goal of $\mathcal{A}$ is to formulate a query q_t for which the algorithm HonestAlg produces an incorrect response d_t. We remark that the algorithm may not have access to X, after constructing $\mathcal{D}$, to respond to the query q_t. On the other hand, $\mathcal{A}$ can use previous outputs to possibly determine the internal randomness of the data structure $\mathcal{D}$ and make future queries accordingly. In this case, the analysis of many randomized algorithms fails because it assumes that the randomness of the algorithm is independent of the input. Consequently, it does not seem evident how to handle Q adaptive queries without implementing Q instances of a non-adaptive data structure, i.e., each instance handles a separate query. Thus, a natural question to ask is whether a space overhead of $\Omega(Q)$ is necessary.

We show that a space overhead of $\Omega(Q)$ is unnecessary by giving a unified framework with only an $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ space overhead.

Theorem 1.2. *Given a data structure $\mathcal{D}$ that answers a query q with probability at least $\frac{3}{4}$ using space S and query time T, there exists a data structure that answers Q adaptive queries, with high probability, i.e., $1 - \frac{1}{\mathrm{poly}(n,Q)}$, using space $\mathcal{O}\left(S\sqrt{Q}\log(nQ)\right)$ and query time $\widetilde{\mathcal{O}}\left(T\log(nQ) + \log^3(nQ)\right)$.*

To concretely instantiate our framework and state an example, we consider the adaptive distance estimation problem defined as follows. In the adaptive distance estimation problem, there exists a set $X = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ of n points in $\mathbb{R}^d$. We are also given an accuracy parameter $\varepsilon > 0$. A query is of the form $\mathbf{q}$, and the algorithm must output a $(1+\varepsilon)$-approximation to $\|\mathbf{x}^{(i)} - \mathbf{q}\|_p$ for all i. The trivial solution of storing all n points and computing all n distances to a query point uses space and query time $\mathcal{O}(nd)$. Cherapanamjeri & Nelson (2020) improved the query time to $\widetilde{\mathcal{O}}\left(\frac{n+d}{\varepsilon^2}\right)$ at the cost of using $\widetilde{\mathcal{O}}\left(\frac{(n+d)d}{\varepsilon^2}\right)$ space and $\widetilde{\mathcal{O}}\left(\frac{nd^2}{\varepsilon^2}\right)$ pre-processing time, while permitting an arbitrary number of queries. By comparison, our data structure handles Q queries of approximate distances from a *specified point in* X, using query time $\widetilde{\mathcal{O}}\left(\frac{n+d}{\varepsilon^2}\right)$, pre-processing time $\widetilde{\mathcal{O}}\left(\frac{nd\sqrt{Q}}{\varepsilon^2}\right)$, and space $\widetilde{\mathcal{O}}\left(\frac{(n+d)\sqrt{Q}}{\varepsilon^2}\right)$. Thus, in the regime where $d \gg n\sqrt{Q}$, our data structure already improves on the work of Cherapanamjeri & Nelson (2020).

A noticeable weakness of our construction is that the Q queries return only the approximate distance between a query point and a single point in X, whereas Cherapanamjeri & Nelson (2020) outputs approximate distances to all points in X. Moreover, Cherapanamjeri & Nelson (2022) subsequently improve the pre-processing time to $\widetilde{\mathcal{O}}\left(\frac{nd}{\varepsilon^2}\right)$. Thus we open up our framework to (1) show that it can be further improved to handle the case where we return the approximate distances of all points in X from Q adaptive query points and (2) achieve pre-processing time $\widetilde{\mathcal{O}}\left(\frac{nd}{\varepsilon^2}\right)$.

Theorem 1.3. *There is a data structure which, when instantiated with dataset $X = \{x_i\}_{i\in[n]} \subset \mathbb{R}^d$ and query bound $Q \leq d$, answers any sequence of Q adaptively chosen distance estimation queries correctly with probability at least 0.99. Furthermore, the space complexity of the data structure is $\widetilde{O}(\varepsilon^{-2} \cdot n\sqrt{Q})$ and the setup and query times are $\widetilde{O}(\varepsilon^{-2} \cdot nd)$ and $\widetilde{O}(\varepsilon^{-2} \cdot (n+d))$, respectively.*

Another application of our framework is the adaptive kernel density estimation problem, where there exists a set $X = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ of n points in $\mathbb{R}^d$ and the goal is to output a $(1+\varepsilon)$-approximation to the quantity $\frac{1}{n}\sum_{i\in[n]} k(\mathbf{x}^{(i)}, \mathbf{q})$, for an accuracy parameter $\varepsilon > 0$, a query $\mathbf{q}$, and a kernel function k, under the promise that the output is at least some threshold $\tau > 0$. Backurs et al. (2019) give an algorithm for kernel density estimation that uses $\mathcal{O}\left(\frac{1}{\tau\varepsilon^2}\right)$ space and $\mathcal{O}\left(\frac{d}{\sqrt{\tau}\varepsilon^2}\right)$ query time, improving over the standard algorithm that samples $\mathcal{O}\left(\frac{1}{\tau\varepsilon^2}\right)$ points and then uses $\mathcal{O}\left(\frac{d}{\tau\varepsilon^2}\right)$ query time to output the empirical kernel density. However, the analysis for both of these algorithms fails for the adaptive setting, where there can be dependencies between the query and the data structure. By using the data structure of Backurs et al. (2019) as a subroutine, our framework immediately implies an algorithm for adaptive kernel density estimation that uses $\widetilde{\mathcal{O}}\left(\frac{\sqrt{Q}}{\tau\varepsilon^2}\right)$ space and $\mathcal{O}\left(\frac{d\log Q}{\sqrt{\tau}\varepsilon^2}\right)$ query time to answer each of Q adaptive queries. In this case, we are again able to go beyond our framework and give a data structure that handles an unlimited number of adaptive kernel density queries:

Theorem 1.4. *Suppose the kernel function k is L-Lipschitz in the second variable for some $L > 0$, i.e., $|k(\mathbf{x},\mathbf{y}) - k(\mathbf{x},\mathbf{z})| \leq L\|\mathbf{y}-\mathbf{z}\|_2$ for all $\mathbf{x},\mathbf{y},\mathbf{z} \in \mathbb{R}^d$. Moreover, suppose that for all $\|\mathbf{x}-\mathbf{y}\|_2 \leq \rho$, we have $k(\mathbf{x},\mathbf{y}) \leq \frac{\tau}{3}$. Then an algorithm that produces a kernel density estimation data structure D that is L-Lipschitz over a set X of points with diameter at most Δ and outputs a $(1+\varepsilon)$-approximation to KDE queries with value at least τ with probability at least $1-\delta$ using space $S(n,\varepsilon,\tau,\log\delta)$ and query time $T(n,\varepsilon,\tau,\log\delta)$, then there exists a KDE data structure that with probability at least 0.99, outputs a $(1+\varepsilon)$-approximation to any number of KDE queries with value at least τ using space $S\left(n, \mathcal{O}(\varepsilon), \mathcal{O}(\tau), \mathcal{O}\left(d\log\frac{(\Delta+\rho)L}{\varepsilon\tau}\right)\right)$ and query time $T\left(n, \mathcal{O}(\varepsilon), \mathcal{O}(\tau), \mathcal{O}\left(d\log\frac{(\Delta+\rho)L}{\varepsilon\tau}\right)\right)$.*

Additionally, we show that our framework guarantees adversarial robustness for a number of other important problems such as nearest neighbor search, range queries, point queries, matrix-vector norm queries, and linear regression. Finally, we supplement our theoretical results with a number of empirical evaluations, which are in the appendix.

1.2 Our Techniques

Dynamic regression on adaptive inputs. Our dynamic algorithm for dynamic maintenance of least-squares objective exploits two main ideas. First, standard results in sketching and sampling show that it suffices to solve for the sketched objective of $\min_{\mathbf{x}\in\mathbb{R}^d}\|\mathbf{SAx}-\mathbf{Sb}\|_2^2$, where $\mathbf{S}$ is an ℓ_2 subspace embedding for $\mathbf{A}$. Here, we exploit several techniques from numerical linear algebra and in particular use leverage score sampling to obtain a subspace embedding $\mathbf{S}$ of $\mathbf{A}$. By standard results in sketching, a $(1+\varepsilon)$ optimal solution is given by $\mathbf{x}^* = (\mathbf{SA})^\dagger\mathbf{Sb}$. Moreover, since the goal is to output the objective value instead of the solution vector, we may take a Johnson-Lindenstrauss (JL) sketch to further reduce dimensionality and run-time. This allows us to focus on $\|\mathbf{GAx}^*-\mathbf{Gb}\|_2^2$, where $\mathbf{G} \in \mathbb{R}^{O(\log d)\times n}$ is a JL sketch.

As a result, our algorithm dynamically maintains a solution $\mathbf{GA}(\mathbf{SA})^\dagger\mathbf{b}$ in this sketched space. To achieve that, we first explicitly solve $\mathbf{GA}(\mathbf{SA})^\dagger$ in pre-processing. Since $\mathbf{GA}$ has few rows, this reduces to a small number of linear solves and can be computed fast via conjugate gradient-type methods. To handle the updates, we leverage their sparsity to efficiently maintain the solution and show that each round takes roughly $\mathcal{O}(K)$ time. Amortizing the pre-processing with the update costs over all iterations yields our desired run-time.

Finally, we apply techniques from differential privacy to ensure adversarial robustness, by aggregating independent copies of the algorithm via a private median mechanism. Intuitively, the private mechanism hides the internal randomness of the algorithm and therefore prevents the adversary from otherwise choosing a "bad" input based on knowledge of internal parameters.

Adaptive query framework. Our framework maintains $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ instances of the non-adaptive data structure and crucially uses differential privacy (DP) to protect the internal randomness of the data structures. In addition to our previous results for dynamic regression, the technique of using DP to hide randomness has recently been used in the streaming model (Hassidim et al., 2020; Kaplan et al., 2021) and the dynamic model (Beimel et al., 2021). These works elegantly use the advanced composition property of DP to bound the number of simultaneous algorithms that must be used in terms of the number of times the output changes "significantly" over the course of the stream. In the

streaming model, the robust algorithms proceed by instantiating many "hidden" copies of a standard randomized algorithm. As the stream arrives, the algorithms are updated and an answer, aggregated using DP, is reported. Crucially, many of these results exploit the fact that the output answer is monotonic in the stream so that there is a known upper bound on the final output. Thus, the reported answers can only increase by a multiplicative factor at most a logarithmic number of times, which is used to bound the initial number of algorithms which are initialized. In our centralized setting, this can be imagined as setting the parameter Q. The main parameter of interest in the streaming literature is the space used by the streaming algorithms, whereas we are concerned with both space usage and query times. Furthermore, stream elements are only accessed one at a time and cannot be processed together unless memory is used. In our case, the dataset is given to us upfront and we can pre-process it to construct a data structure towards solving a centralized problem.

The work by Beimel et al. (2021) shares many of these ideas: the authors are concerned with dynamic graph algorithms where an adversary can update the graph in an adaptive fashion. Similar tools such as multiple randomized initialization and aggregated responses using DP are utilized. The main difference is their parameters of interest: the goal of Beimel et al. (2021) is to have a fast *amortized* update time across many queries. This necessitates the need to "throw away" existing algorithms and start with fresh randomness at intermittent points. In contrast, we study a centralized setting where the underlying dataset is not updated but we wish to answer Q adaptive queries on the dataset.

Inspired by these works, our main framework also uses advanced composition to show the sufficiency of maintaining $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ data structures to answer Q adaptive queries in the centralized setting, which gives a rich set of applications. Moreover, to improve the query time of our algorithms, we further invoke the privacy amplification of sampling to show that it suffices to output the private median of a small subset, i.e., a subset of size $\mathcal{O}\left(\log Q\right)$, of these $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ data structures. Thus our framework only incurs a logarithmic overhead in query time and an $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ overhead in space. Surprisingly, our simple framework gives diverse applications for adaptive algorithms on a number of important problems, including estimating matrix-vector norms, adaptive range query search, adaptive nearest neighbor search, and adaptive kernel density estimation, to name a few. These applications are discussed in depth in Section 3.

Adaptive distance estimation. To achieve better pre-processing time for adaptive distance estimation, our main technique is to sample groups of rows from a Hadamard transform and argue that an interaction with a separate group should be considered in separate privacy budgets, effectively arguing that outputting n approximate distances to a single adaptive query only uses one unit of privacy budget. By contrast, our black-box framework charges one unit of privacy budget per approximate distance, so that outputting n approximate distances would use n units of privacy budget.

Adaptive kernel density estimation. Theorem 1.4 is based on showing that with constant probability, our data structure is accurate on all possible queries in $\mathbb{R}^d$. In particular, we first show that our data structure is accurate on a sufficiently fine net of points through a standard union bound argument, which incurs the d overhead compared to the space required to handle a single query. We then show that if the algorithm and the kernel function are both Lipschitz, which is true for sampling-based algorithms and a number of standard kernel functions, then accuracy on the net implies accuracy on all possible points in $\mathbb{R}^d$.

2 Dynamic Regression under Label Updates

In this section, we consider the dynamic problem of maintaining the cost of the least-squares regression, where the labels receive adaptively chosen updates. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be the design matrix and $\mathbf{b} \in \mathbb{R}^n$ be the target label. A classic problem in numerical linear algebra and optimization is to solve the ℓ_2 least-squares regression objective

$$F(\mathbf{A}, \mathbf{b}) = \min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \left\|\mathbf{A}\mathbf{A}^\dagger \mathbf{b} - \mathbf{b}\right\|_2^2. \tag{2.1}$$

We consider a dynamic version of the problem, where the label receives adaptively chosen updates. We assume that each update can only affect K entries of the label vector. In this setting, we show:

Theorem 2.1 (Main theorem; dynamic maintenance of regression cost)**.** *Let $\varepsilon \in (0, 1/4)$ be an error parameter and $\mathbf{b}^{(1)}$ be the initial target label. Given $\varepsilon, \mathbf{A}, \mathbf{b}^{(1)}$, a stream of T adaptively chosen,*

K-sparse updates to the label, Algorithm 5 outputs an estimate $\widehat{F}_i$ such that $\widehat{F}_i = (1 \pm \varepsilon) F(\mathbf{A}, \mathbf{b}^{(i)})$ for all i with high probability.

Furthermore, the algorithm requires a preprocessing step in time $\widetilde{\mathcal{O}}\left(\operatorname{nnz}(\mathbf{A}) + \operatorname{poly}(d)\right)$. The amortized update time of the algorithm is

$$\widetilde{\mathcal{O}}\left(\sqrt{K \operatorname{nnz}(\mathbf{A})}\left(\sqrt{\kappa(\mathbf{A})} + \varepsilon^{-3}\right)\right).$$

We defer the technical details to Appendix B. Here, we describe the main ideas of the algorithm.

At a high-level, our algorithm implements a *sketch-and-solve* strategy. First, in pre-processing, the algorithm samples a leverage score sketching matrix $\mathbf{S} \in \mathbb{R}^{k \times n}$, where $k = \mathcal{O}\left(d \log d / \varepsilon^2\right)$. This provides a $(1+\varepsilon)$ ℓ_2 subspace embedding for $\mathbf{A}$. Standard results in sketching imply that it suffices to solve for the sketched objective of $\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{SAx} - \mathbf{Sb}\|_2^2$ (Sarlos (2006); Clarkson & Woodruff (2013; 2017); Woodruff (2014)). Let $\widehat{\mathbf{A}} = \mathbf{SA}$. A $(1+\varepsilon)$ optimal solution is thus given by $\widehat{\mathbf{A}}^{\dagger}\mathbf{b}$. Moreover, our goal is to maintain the regression cost, rather than this solution vector. Hence, we can apply the Johnson–Lindenstrauss lemma and focus on

$$\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{SAx} - \mathbf{Sb}\|_2^2 \approx \left\|\mathbf{GA}(\mathbf{SA})^{\dagger}\mathbf{Sb} - \mathbf{Gb}\right\|_2^2, \tag{2.2}$$

where $\mathbf{G} \in \mathbb{R}^{\mathcal{O}\left(\log n / \varepsilon^2\right) \times n}$ is a JL sketch.

To track the cost value dynamically, the algorithm first computes and stores $\mathbf{M} = \mathbf{GA}(\mathbf{SA})^{\dagger}$ in pre-processing. In the 1-st step, given the initial target label $\mathbf{b}^{(1)}$, the algorithm computes $\mathbf{Sb}^{(i)}$, $\mathbf{M}\left(\mathbf{Sb}^{(1)}\right)$ and $\mathbf{Gb}^{(1)}$. Then it outputs $\widehat{F}_1 = \left\|\mathbf{MSb}^{(1)} - \mathbf{Gb}^{(1)}\right\|_2^2$ as an estimate of the regression cost. For the later steps, we show how to maintain $\mathbf{Gb}^{(i)}, \mathbf{Sb}^{(i)}$ efficiently, by exploiting the sparsity of the updates. Finally, to remain robust under adaptive inputs, we aggregate multiple copies using private median and carefully balance the parameters to achieve the run-time guarantee.

3 A Framework for Adversarial Robustness

In this section, we describe our main framework, which enables Q adaptive queries to a data structure by using $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ copies of a non-adaptive data structure. We show that through advanced composition of differential privacy, the private median of $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ copies protects the internal randomness of each non-adaptive data structure while still adding sufficiently small noise to guarantee accuracy. Moreover, we use amplification of privacy by sampling to only consider a small subset of the $\widetilde{\mathcal{O}}\left(\sqrt{Q}\right)$ non-adaptive data structures to further improve the runtime.

Algorithm 1 Adaptive Algorithm Interaction

1: $r \leftarrow \mathcal{O}\left(\sqrt{Q}\log^2(nQ)\right)$, $k \leftarrow \mathcal{O}\left(\log(nQ)\right)$
2: **for** $i \in [r]$ **do**
3: Implement data structure $\mathcal{D}_i$ on the input
4: **for** each query q_i, $i \in [Q]$ **do**
5: Let S be a set of k indices sampled (with replacement) from $[r]$
6: For each $j \in [k]$, let $d_{i,j}$ be the output of $\mathcal{D}_{S_j}$ on query q_i
7: $d_i \leftarrow \mathsf{PrivMed}(\{d_{i,j}\}_{j \in [k]})$, where $\mathsf{PrivMed}$ is $(1,0)$-DP

We first argue that Algorithm 1 maintains accuracy against Q rounds of interaction with an adaptive adversary. Let $R = \{R^{(0)}, R^{(1)}, \ldots, R^{(r)}\}$, where $R^{(1)}, \ldots, R^{(r)}$ denotes the random strings used by the oblivious data structures $\mathcal{D}_1, \ldots, \mathcal{D}_r$ and $R^{(0)}$ denotes the additional randomness used by Algorithm 1, such as in the private median subroutine $\mathsf{PrivMed}$. Consider a transcript $T(R) = \{T_1, \ldots, T_Q\}$ such that for each $i \in [Q]$, we define $T_i = (q_i, d_i)$ to be the ordered pair consisting of the query q_i and the corresponding answer d_i by Algorithm 1 using the random string $R^{(0)}$, as well as the oblivious data structures $\mathcal{D}_1, \ldots, \mathcal{D}_r$ with random strings $R^{(1)}, \ldots, R^{(r)}$. We remark that d_i is a random variable due to the randomness of each data structure, as well as the randomness of the private median subroutine $\mathsf{PrivMed}$.

We will first argue that the transcript T_R is differentially private with respect to R. Similar arguments were made in the streaming model by Hassidim et al. (2020) and in the dynamic model Attias et al. (2021); Beimel et al. (2021).

Lemma 3.1. *For a fixed iteration, T_i is $\left(\mathcal{O}\left(\frac{1}{\sqrt{Q}\log(nQ)}\right), 0\right)$-differentially private with respect to R.*

Proof. We first observe that PrivMed is $(1, 0)$-differentially private on the outputs of the $r = \mathcal{O}\left(\sqrt{Q}\log^2(nQ)\right)$ data structures. Algorithm 1 samples $k = \mathcal{O}\left(\log(nQ)\right)$ groups of data structures from the r total data structures. Thus by amplification via sampling, i.e., Theorem A.2, PrivMed is $\left(\mathcal{O}\left(\frac{1}{\sqrt{Q}\log(nQ)}\right), 0\right)$-differentially private. Therefore, T_i is $\left(\mathcal{O}\left(\frac{1}{\sqrt{Q}\log(nQ)}\right), 0\right)$-differentially private with respect to R. □

We next argue that the entire transcript is differentially private with respect to the randomness R.

Lemma 3.2. *T is $\left(\mathcal{O}\left(1\right), \frac{1}{\mathrm{poly}(nQ)}\right)$-differentially private with respect to R.*

Proof. By Lemma 3.1, for each fixed iteration $i \in [Q]$, the transcript T_i is $\left(\mathcal{O}\left(\frac{1}{\sqrt{Q}\log(nQ)}\right), 0\right)$-differentially private with respect to R. Note that the transcript T is an adaptive composition of the transcripts $T_1, \ldots, T_Q$. Thus, by the advanced composition of differential privacy, i.e., Theorem A.4, the transcript T is $\left(\mathcal{O}\left(1\right), \frac{1}{\mathrm{poly}(nQ)}\right)$-differentially private with respect to R. □

We now prove the correctness of our unifying framework.

Proof of Theorem 1.2: For a fixed query q_i with $i \in [Q]$, let S be the corresponding set of k indices sampled from $[r]$. Let $\mathcal{V}$ be the set of valid answers on query q_i. Let I_j be an indicator variable for whether the output $d_{i,j}$ on query q_i by $\mathcal{D}_{S_j}$ is correct, so that $I_j = 1$ if $d_{i,j} \in \mathcal{V}$ and $I_j = 0$ if $d_{i,j} \notin \mathcal{V}$. By assumption, we have that for each $j \in [k]$,

$$\mathbf{Pr}\left[I_j = 1\right] \geq \frac{3}{4},$$

so that $\mathbb{E}\left[I_j\right] \geq \frac{3}{4}$. We define the random variable $I = \frac{1}{k}\sum_{j\in[k]} I_j$ so that by linearity of expectation, $\mathbb{E}\left[I\right] = \frac{1}{k}\sum_{j\in[k]} \mathbb{E}\left[I_j\right] \geq \frac{3}{4}$.

To handle a sequence of Q adaptive queries, we consider the transcript $T(R) = \{T_1, \ldots, T_Q\}$ for the randomness $R = \{R^{(0)}, R^{(1)}, \ldots, R^{(r)}\}$ previously defined, i.e., for each $i \in [Q]$, $T_i = (q_i, d_i)$ is the ordered pair consisting of the query q_i and the corresponding answer d_i by Algorithm 1 using the random string $R^{(0)}$, as well as the oblivious data structures $\mathcal{D}_1, \ldots, \mathcal{D}_r$ with random strings $R^{(1)}, \ldots, R^{(r)}$. By Lemma 3.2, we have that T is $\left(\mathcal{O}\left(1\right), \frac{1}{\mathrm{poly}(nQ)}\right)$-differentially private with respect to R.

For $j \in [k]$, we define the function $\mathrm{success}(R^{(S_j)})$ to be the indicator variable for whether the output d_{i,S_j} by data structure D_{S_j} is successful on query q_i. For example, if D is supposed to answer queries within $(1+\alpha)$-approximation, then we define $\mathrm{success}(R^{(S_j)})$ to be one if d_{i,S_j} is within a $(1+\alpha)$-approximation to the true answer on query q_i, and zero otherwise. From the generalization properties of differential privacy, i.e., Theorem A.5, we have

$$\mathbf{Pr}\left[\left|\frac{1}{k}\sum_{j\in[k]} \mathrm{success}(R^{(S_j)}) - \mathop{\mathbb{E}}_{\overline{R}}\left[\mathrm{success}(\overline{R})\right]\right| \geq \frac{1}{10}\right] < \frac{1}{\mathrm{poly}(n, Q)},$$

for sufficiently small $\mathcal{O}\left(1\right)$. Therefore, by a union bound over all Q queries, we have

$$\mathbf{Pr}\left[\frac{1}{k}\sum_{i\in[k]} I_i > 0.6\right] > 1 - \frac{1}{\mathrm{poly}(n, Q)},$$

which implies that d_i is correct on query q_i. Then by a union bound over $i \in [Q]$ for all Q adaptive queries, we have that the data structure answers all Q adaptive queries with high probability. $\square$

4 Applications of Our Framework

Theorem 1.2 has applications to a number of central problems in data science and machine learning. In this section, we formally describe the range queries, point queries, matrix-vector norm queries, and linear regression problems; we defer discussion of adaptive distance estimation, kernel density estimation, and nearest neighbor search to the the appendix.

4.1 Application: Matrix-Vector Norm Queries

In the matrix-vector norm query problem, we are given a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and we would like to handle Q adaptive queries $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(Q)}$ for an approximation parameter $\varepsilon > 0$ by outputting a $(1+\varepsilon)$-approximation to $\|\mathbf{A}\mathbf{x}^{(i)}\|_p$ for each query $\mathbf{x}^{(i)} \in \mathbb{R}^d$ with $i \in [Q]$. Here we define $\|\mathbf{v}\|_p^p = \sum_{i \in [d]} |v_i|^p$ for a vector $\mathbf{v} \in \mathbb{R}^d$. Observe that computing $\mathbf{A}\mathbf{x}^{(i)}$ explicitly and then computing its p-norm requires $\mathcal{O}(nd)$ time. Thus for $n \gg d$, a much faster approach is to produce a subspace embedding, i.e., to compute a matrix $\mathbf{M} \in \mathbb{R}^{m \times d}$ with $m \ll n$, such that for all $\mathbf{x} \in \mathbb{R}^d$,

$$(1-\varepsilon)\|\mathbf{A}\mathbf{x}\|_p \leq \|\mathbf{M}\mathbf{x}\|_p \leq (1+\varepsilon)\|\mathbf{A}\mathbf{x}\|_p.$$

However, because subspace embeddings must be correct over all possible queries, the number of rows of $\mathbf{M}$ is usually $m = \Omega\left(\frac{d}{\varepsilon^2}\right)$ due to requiring correctness over an ε-net.

Theorem 4.1 (Indyk (2006); Li (2008))**.** *Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, $p \in (0, 2]$, and an accuracy parameter $\varepsilon > 0$, there exists an algorithm that creates a data structure that uses $\mathcal{O}\left(\frac{1}{\varepsilon^2} \log n\right)$ bits of space and outputs a $(1+\varepsilon)$-approximation to $\|\mathbf{A}\mathbf{x}\|_p$ for a query $\mathbf{x} \in \mathbb{R}^d$, with high probability, in time $\mathcal{O}\left(\frac{d}{\varepsilon^2} \log n\right)$.*

Theorem 4.1 essentially creates a matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ of random variables sampled from p-stable distribution (Zolotarev, 1986) and then stores the matrix $\mathbf{R}\mathbf{A}$. Once the query $\mathbf{x}$ arrives, the data structure then outputs a $(1+\varepsilon)$-approximation to $\|\mathbf{A}\mathbf{x}\|_p$ by computing a predetermined function on $\mathbf{R}\mathbf{A}\mathbf{x}$. The restriction on $p \in (0, 2]$ is due to the fact that p-stable distributions only exist for $p \in (0, 2]$. From Theorem 4.1 and Theorem 1.2, we have the following:

Theorem 4.2. *Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, $p \in (0, 2]$, and an accuracy parameter $\varepsilon > 0$, there exists an algorithm that creates a data structure that uses $\mathcal{O}\left(\frac{\sqrt{Q}}{\varepsilon^2} \log^2(nQ)\right)$ bits of space and outputs a $(1+\varepsilon)$-approximation to $\|\mathbf{A}\mathbf{x}^{(i)}\|_p$ with $i \in [Q]$ for Q adaptive queries $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(Q)} \in \mathbb{R}^d$, with high probability, in time $\widetilde{\mathcal{O}}\left(\frac{d}{\varepsilon^2} \log^2(nQ) + \log^3(nQ)\right)$.*

4.2 Application: Linear Regression

In the linear regression problem, we are given a fixed matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and we would like to handle Q adaptive queries $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(Q)}$, for an approximation parameter $\varepsilon > 0$, by outputting a $(1+\varepsilon)$-approximation to $\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}^{(i)}\|_2$ for each query $\mathbf{b}^{(i)} \in \mathbb{R}^n$ with $i \in [Q]$. For linear regression, we can again compute a subspace embedding $\mathbf{M} = \mathbf{S}\mathbf{A} \in \mathbb{R}^{m \times n}$ and answer a query $\mathbf{b}^{(i)}$ by approximately solving $\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{S}\mathbf{A}\mathbf{x} - \mathbf{S}\mathbf{b}^{(i)}\|_2$, where $\mathbf{S}$ is a sketching matrix (Clarkson & Woodruff, 2013).

Theorem 4.3 (Clarkson & Woodruff (2013))**.** *Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, $\mathbf{b} \in \mathbb{R}^n$, and an accuracy parameter $\varepsilon > 0$, there exists an algorithm that creates a data structure that uses $\mathcal{O}\left(\frac{d^2}{\varepsilon^2} \log^2(nQ)\right)$ bits of space and outputs a $(1+\varepsilon)$-approximation to $\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ with high probability.*

However, this may fail for multiple interactions with the data structure. For example, suppose the adversary learns the kernel of $\mathbf{S}$. Then the adversary could query some vector $\mathbf{b}^{(i)}$ in the kernel of $\mathbf{S}$ so that $\mathbf{S}\mathbf{b}^{(i)}$ is the all zeros vector, so that the output is the all zeros vector of dimension d, which could be arbitrarily bad compared to the actual minimizer. Thus the naïve approach is to maintain

Q subspace embeddings, one for each query, resulting in a data structure with space $\widetilde{\mathcal{O}}\left(\frac{Qd}{\varepsilon^2}\right)$. By comparison, Theorem 4.3 and Theorem 1.2 yield the following:

Theorem 4.4. *Given $\mathbf{A} \in \mathbb{R}^{n\times d}$ and an accuracy parameter $\varepsilon > 0$, there exists an algorithm that creates a data structure that uses $\mathcal{O}\left(\frac{\sqrt{Q}d^2}{\varepsilon^2}\log^3(nQ)\right)$ bits of space and with high probability, outputs $(1+\varepsilon)$-approximations to $\min_{\mathbf{x}\in\mathbb{R}^d}\|\mathbf{A}\mathbf{x}-\mathbf{b}^{(i)}\|_2$ for Q adaptive queries $\mathbf{b}^{(1)},\ldots,\mathbf{b}^{(Q)}$.*

4.3 Application: Half-Space Queries

Given a set P of n points in $\mathbb{R}^d$, the range query or search problem asks us to pre-process P so that given a region R, chosen from a predetermined family, one can quickly count or return the points in $P \cap R$. This is an extremely well-studied class of problems in computational geometry Toth et al. (2017) and the case where the regions R are hyperplanes (also called half-spaces) is of special interest since many algebraic constraints can be "lifted" to be hyperplanes in a higher dimension.

Unfortunately, exact versions of the problem are known to have the "curse of dimensionality" and suffer from exponential dependence on d in the query time (Brönnimann et al., 1993; Chazelle, 2000). Nonetheless, Chazelle et al. (2008) gave a data structure capable of answering hyperplane queries *approximately* with polynomial query time. Their notion of approximation is as follows: given a set of points P in the unit ℓ_2 ball, hyperplane R, and $\varepsilon > 0$, we return the number of points that are on a given side of the hyperplane R up to additive error equal to the number of points in P which lie within distance ε of the boundary of R. We will refer to this query as an ε-approximate hyperplane query. Chazelle et al. (2008) proved the following theorem.

Theorem 4.5 (Chazelle et al. (2008))**.** *Given a set of points P that lie in the unit ℓ_2 ball, there exists a data structure that pre-processes P using space $\widetilde{\mathcal{O}}\left(dn^{O(\varepsilon^{-2})}\right)$ such that any ε-approximate hyperplane range query is answered correctly with high probability. The query time is $\widetilde{\mathcal{O}}\left(d/\varepsilon^2\right)$.*

The data structure of Chazelle et al. (2008) is randomized and in particular employs randomized dimensionality reduction. Thus, it is feasible that queries might fail for multiple adaptive interactions with the data structure. By utilizing our framework of Section 3 and Theorem 1.2, we can obtain the following robust guarantee.

Theorem 4.6. *Given a set of points P that lie in the unit ℓ_2 ball, there exists a data structure which pre-processes P using space $\widetilde{\mathcal{O}}\left(\sqrt{Q}dn^{O(\varepsilon^{-2})}\right)$ such that Q adaptive ε-hyperplane range queries are answered correctly with high probability. The query time is $\widetilde{\mathcal{O}}\left(d/\varepsilon^2\right)$.*

4.4 Application: Point Queries on Turnstile Streams

In the problem of point queries on turnstile streams, there exists a stream of m updates. Each update specifies a coordinate $i \in [n]$ of an underlying frequency vector $f \in \mathbb{R}^n$ and changes f_i by some amount between $\Delta_i \in [-\Delta, \Delta]$, where $\Delta = \mathrm{poly}(n)$. Given any *constant* accuracy parameter $\varepsilon > 0$ any time $t \in [m]$, we define $f^{(t)}$ to be the frequency vector implicitly defined after the first t updates. Then the point query problem is to output $f_i^{(t)}$ for various choices of $t \in [m]$ and $i \in [n]$ within an additive error of $\varepsilon\|f^{(t)}\|_1$.

Theorem 4.7 (Alman & Yu (2020))**.** *There exists an algorithm that uses space $\mathcal{O}\left(\log^2 n\right)$ bits, worst-case update time $\mathcal{O}\left(\log^{0.582} n\right)$, and query time $\mathcal{O}\left(\log^{1.582} n\right)$, that supports point queries with $\varepsilon = 0.1$ with high probability.*

An important quality of Theorem 4.7 is that it significantly improves the update time over previous data structures, e.g., Charikar et al. (2004), at a cost in query time. By applying Theorem 1.2, we can avoid a blow-up in query time while still enjoying the update time improvements:

Theorem 4.8. *There exists an algorithm that uses space $\mathcal{O}\left(\sqrt{Q}\log^3(nQ)\right)$ bits, has worst-case update time $\mathcal{O}\left(\sqrt{Q}\log^{1.582}(nQ)\right)$ and query time $\widetilde{\mathcal{O}}\left(\log^3(nQ)\right)$, and supports Q adaptive point queries with $\varepsilon = 0.1$ and with high probability.*

REFERENCES

Miklós Ajtai, Vladimir Braverman, T. S. Jayram, Sandeep Silwal, Alec Sun, David P. Woodruff, and Samson Zhou. The white-box adversarial data stream model. *CoRR*, abs/2204.09136, 2022. 1

Josh Alman and Huacheng Yu. Faster update time for turnstile streaming algorithms. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 1803–1813, 2020. 9

Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. *Advances in neural information processing systems*, 28, 2015. 25

Idan Attias, Edith Cohen, Moshe Shechner, and Uri Stemmer. A framework for adversarial streaming via differential privacy and difference estimators. *CoRR*, abs/2107.14527, 2021. 1, 7

Dmitrii Avdiukhin, Slobodan Mitrovic, Grigory Yaroslavtsev, and Samson Zhou. Adversarially robust submodular maximization under knapsack constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, pp. 148–156, 2019. 1

Arturs Backurs, Piotr Indyk, and Tal Wagner. Space and time efficient kernel density estimation in high dimensions. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pp. 15773–15782, 2019. 4, 23, 24

Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan R. Ullman. Algorithmic stability for adaptive data analysis. *SIAM J. Comput.*, 50(3), 2021. 13

Amos Beimel, Haim Kaplan, Yishay Mansour, Kobbi Nissim, Thatchaphol Saranurak, and Uri Stemmer. Dynamic algorithms against an adaptive adversary: Generic constructions and lower bounds. *CoRR*, abs/2111.03980, 2021. 1, 2, 4, 5, 7

Omri Ben-Eliezer, Rajesh Jayaram, David P. Woodruff, and Eylon Yogev. A framework for adversarially robust streaming algorithms. *SIGMOD Rec.*, 50(1):6–13, 2021. 1

Omri Ben-Eliezer, Talya Eden, and Krzysztof Onak. Adversarially robust streaming via dense-sparse trade-offs. In *5th Symposium on Simplicity in Algorithms, SOSA@SODA*, pp. 214–227, 2022. 1

Aaron Bernstein, Jan van den Brand, Maximilian Probst Gutenberg, Danupon Nanongkai, Thatchaphol Saranurak, Aaron Sidford, and He Sun. Fully-dynamic graph sparsifiers against an adaptive adversary. In *49th International Colloquium on Automata, Languages, and Programming, ICALP*, pp. 20:1–20:20, 2022. 1

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 Proceedings,*, 2013. 1

Ilija Bogunovic, Slobodan Mitrovic, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, pp. 508–516, 2017. 1

Vladimir Braverman, Avinatan Hassidim, Yossi Matias, Mariano Schain, Sandeep Silwal, and Samson Zhou. Adversarial robustness of streaming algorithms through importance sampling. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS*, pp. 3544–3557, 2021. 1

Hervé Brönnimann, Bernard Chazelle, and János Pach. How hard is half-space range searching? *Discrete & Computational Geometry*, 10(2):143–155, 1993. 9

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS*, pp. 634–649, 2015. 13

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP*, pp. 39–57. IEEE Computer Society, 2017. 1

Amit Chakrabarti, Prantar Ghosh, and Manuel Stoeckl. Adversarially robust coloring for graph streams. In *13th Innovations in Theoretical Computer Science Conference, ITCS*, pp. 37:1–37:23, 2022. 1

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, 2009. 1

Moses Charikar and Paris Siminelakis. Hashing-based-estimators for kernel density in high dimensions. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pp. 1032–1043, 2017. 24

Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004. 9

Bernard Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, 2000. 9

Bernard Chazelle, Ding Liu, and Avner Magen. Approximate range searching in higher dimension. *Computational Geometry*, 39(1):24–29, 2008. 9

Nadiia Chepurko, Kenneth Clarkson, Lior Horesh, Honghao Lin, and David Woodruff. Quantum-inspired algorithms from randomized numerical linear algebra. In *International Conference on Machine Learning (ICML)*, 2022. 14

Yeshwanth Cherapanamjeri and Jelani Nelson. On adaptive distance estimation. In *Advances in Neural Information Processing Systems 33: NeurIPS*, 2020. 3, 19, 20, 25, 26

Yeshwanth Cherapanamjeri and Jelani Nelson. Uniform approximations for randomized hadamard transforms with applications. *CoRR*, abs/2203.01599, 2022. 3, 19, 21, 26

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Symposium on Theory of Computing Conference, STOC*, pp. 81–90, 2013. 6, 8, 15

Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017. 6, 15

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC, Proceedings*, pp. 265–284, 2006. 13

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 51–60, 2010. 13

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pp. 117–126. ACM, 2015. 13

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*, pp. 536–539, 2012. 1

Gene H Golub and Charles F Van Loan. *Matrix computations*. Johns Hopkins University Press, 2013. 16

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR, Conference Track Proceedings*, 2015. 1

Moritz Hardt, Nimrod Megiddo, Christos H. Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pp. 111–122, 2016. 1

Avinatan Hassidim, Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Adversarially robust streaming algorithms via differential privacy. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020. 1, 2, 4, 7, 13

Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006. 8

Shunhua Jiang, Binghui Peng, and Omri Weinstein. Dynamic least-squares regression. *arXiv preprint arXiv:2201.00228*, 2022. 2

Haim Kaplan, Yishay Mansour, Kobbi Nissim, and Uri Stemmer. Separating adaptive streaming from oblivious streaming using the bounded storage model. In *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO, Proceedings, Part III*, pp. 94–121, 2021. 1, 4

Ping Li. Estimators and tail bounds for dimension reduction in ℓ_α $(0 < \alpha \leq 2)$ using stable random projections. In Shang-Hua Teng (ed.), *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 10–19, 2008. 8

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR, Conference Track Proceedings*, 2017. 1

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR*, 2018. 1

Michael W Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011. 14

Ilya Mironov, Moni Naor, and Gil Segev. Sketching in adversarial environments. *SIAM J. Comput.*, 40(6):1845–1870, 2011. 1

Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. *ACM Trans. Algorithms*, 15 (3):35:1–35:30, 2019. 1

Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016. 1

Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS*, pp. 506–519, 2017. 1

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *47th annual IEEE symposium on foundations of computer science (FOCS)*, pp. 143–152, 2006. 6, 15

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, Conference Track Proceedings*, 2014. 1

Csaba D Toth, Joseph O'Rourke, and Jacob E Goodman. *Handbook of discrete and computational geometry*. CRC press, 2017. 9

David Wajc. Rounding dynamic matchings against an adaptive adversary. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pp. 194–207, 2020. 1

David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014. 2, 6, 14, 15

David P. Woodruff and Samson Zhou. Tight bounds for adversarially robust streams and sliding windows via difference estimators. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pp. 1183–1196, 2021. 1

Vladimir M Zolotarev. *One-dimensional stable distributions*, volume 65. American Mathematical Soc., 1986. 8