Querying a Matrix through Matrix-Vector Products

XIAOMING SUN, Institute of Computing Technology, Chinese Academy of Sciences, China and University of Chinese Academy of Sciences, China
DAVID P. WOODRUFF, Carnegie Mellon University, USA
GUANG YANG, Institute of Computing Technology, Chinese Academy of Sciences, China and Conflux, China
JIALIN ZHANG, Institute of Computing Technology, Chinese Academy of Sciences, China and University of Chinese Academy of Sciences, China

We consider algorithms with access to an unknown matrix $\mathbf{M} \in \mathbb{F}^{n \times d}$ via *matrix-vector products*, namely, the algorithm chooses vectors $\mathbf{v}^1, \ldots, \mathbf{v}^q$, and observes $\mathbf{M}\mathbf{v}^1, \ldots, \mathbf{M}\mathbf{v}^q$. Here the $\mathbf{v}^i$ can be randomized as well as chosen adaptively as a function of $\mathbf{M}\mathbf{v}^1, \ldots, \mathbf{M}\mathbf{v}^{i-1}$. Motivated by applications of sketching in distributed computation, linear algebra, and streaming models, as well as connections to areas such as communication complexity and property testing, we initiate the study of the number q of queries needed to solve various fundamental problems. We study problems in three broad categories, including linear algebra, statistics problems, and graph problems. For example, we consider the number of queries required to approximate the rank, trace, maximum eigenvalue, and norms of a matrix $\mathbf{M}$; to compute the AND/OR/Parity of each column or row of $\mathbf{M}$, to decide whether there are identical columns or rows in $\mathbf{M}$ or whether $\mathbf{M}$ is symmetric, diagonal, or unitary; or to compute whether a graph defined by $\mathbf{M}$ is connected or triangle-free. We also show separations for algorithms that are allowed to obtain matrix-vector products only by querying vectors on the right, versus algorithms that can query vectors on both the left and the right. We also show separations depending on the underlying field the matrix-vector product occurs in. For graph problems, we show separations depending on the form of the matrix (bipartite adjacency versus signed edge-vertex incidence matrix) to represent the graph.

Surprisingly, very few works discuss this fundamental model, and we believe a thorough investigation of problems in this model would be beneficial to a number of different application areas.

CCS Concepts: • **Theory of computation → Complexity classes**; **Communication complexity;**

Additional Key Words and Phrases: Communication complexity, linear algebra, sketching

An extended abstract of this manuscript appeared at ICALP 2019 [36].
All authors contributed equally to this research.
This work was supported in part by the National Natural Science Foundation of China under Grant No. 61832003, 61872334, the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000 and K. C. Wong Education Foundation. D. Woodruff would also like to thank the National Science Foundation grant No. CCF-181584 for partial support, as well as the Chinese Academy of Sciences and the Simons Institute for the Theory of Computing, where part of this work was done.
Authors' addresses: X. Sun, and J. Zhang, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, University of Chinese Academy of Sciences, Beijing, China; emails: {sunxiaoming, zhangjialin}@ict.ac.cn; D. P. Woodruff, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA; email: dwoodruf@andrew.cmu.edu; G. Yang, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, Conflux, Beijing, China; email: guang.research@gmail.com.

1549-6325/2021/10-ART31 $15.00
https://doi.org/10.1145/3470566

ACM Reference format:
Xiaoming Sun, David P. Woodruff, Guang Yang, and Jialin Zhang. 2021. Querying a Matrix through Matrix-Vector Products. *ACM Trans. Algorithms* 17, 4, Article 31 (October 2021), 19 pages.
https://doi.org/10.1145/3470566

1 INTRODUCTION

Suppose there is an unknown matrix $\mathbf{M} \in \mathbb{F}^{n\times d}$ that you can only access via a sequence of *matrix-vector products* $\mathbf{M}\cdot\mathbf{v}^1, \ldots, \mathbf{M}\cdot\mathbf{v}^q$, where we call the vectors $\mathbf{v}^1, \ldots, \mathbf{v}^q$ the *query vectors*, which can be chosen in a randomized, possibly adaptive way. By adaptive, we mean that $\mathbf{v}^i$ can depend on $\mathbf{v}^1, \ldots, \mathbf{v}^{i-1}$ as well as $\mathbf{M}\mathbf{v}^1, \ldots, \mathbf{M}\mathbf{v}^{i-1}$. Here $\mathbb{F}$ is a field, and we study different fields for different applications. Suppose our goal is to determine if $\mathbf{M}$ satisfies a specific property $\mathcal{P}$, such as having approximately full rank, or, for example, whether $\mathbf{M}$ has two identical columns. A natural question is the following:

Question 1: How many queries q are necessary to determine if $\mathbf{M}$ has property $\mathcal{P}$?

A number of well-studied problems are special cases of this question, i.e., compressed sensing or sparse recovery, for which $\mathbf{M} \in \mathbb{R}^{1\times d}$ is an approximately k-sparse vector, and one would like a number q of queries close to k. It is known that if the query sequence is non-adaptive, meaning $\mathbf{v}^1, \ldots, \mathbf{v}^q$ are chosen before making any queries, then $q = \Theta(k\log(n/k))$ is necessary and sufficient [7, 14] to recover an approximately k-sparse vector.[1] However, if the queries can be adaptive, then $q = O(k\log\log n)$ queries suffice [19], while there is a lower bound of $\Omega(k+\log\log n)$ [33] (see also recent work [20, 32]). There is also work on compressed sensing of matrices, where one recovers sparse matrices via queries of the form $\mathbf{M}\mathbf{v}^1, \mathbf{M}\mathbf{v}^2, \ldots, \mathbf{M}\mathbf{v}^{i-1}$ [15].

The above problem is representative of an emerging field called *linear sketching* which is the underlying technique behind a number of algorithmic advances the past two decades. In this model, one queries $\mathbf{M}\cdot\mathbf{v}^1, \ldots, \mathbf{M}\cdot\mathbf{v}^r$ for non-adaptive queries $\mathbf{v}^1, \ldots, \mathbf{v}^r$. For brevity, we write this as $\mathbf{M}\cdot\mathbf{V}$, where $\mathbf{V} \in \mathbb{F}^{d\times r}$ has ith column equal to $\mathbf{v}^i$. Linear sketching has played a central role in the development of streaming algorithms [3]. Perhaps more surprisingly, linear sketches are also known to achieve the minimal space necessary *of any, possibly non-linear, algorithm* for processing dynamic data streams under certain general conditions [2, 22, 27], which is an essential result for proving a number of lower bounds for approximating matchings in a stream [5, 25]. Linear sketching has also led to the fastest known algorithms for problems in numerical linear algebra, such as least squares regression and low rank approximation; for a survey, see [40]. Note that given $\mathbf{M}\cdot\mathbf{V}$ and $\mathbf{M}'\cdot\mathbf{V}$, by linearity one can compute $(\mathbf{M}+\mathbf{M}')\cdot\mathbf{V} = \mathbf{M}\cdot\mathbf{V} + \mathbf{M}'\cdot\mathbf{V}$. This basic versatility property allows for fast updates in a data stream and mergeability in environments such as MapReduce and other distributed models of computation.

Given the applications above, we consider Question 1 an important question to understand for many different properties $\mathcal{P}$ of interest, which we describe in more detail below. A central goal of this work is to answer Question 1 for such properties, and we believe the query model through matrix-vector products is a natural model of study in its own right.

One notable difference with our model and a number of applications of linear sketching is that we will allow for adaptive query sequences. In fact, our upper bounds will be non-adaptive, and our nearly matching lower bounds for each problem we consider will hold even for adaptive query sequences. The adaptive query model is common in practice. For the problems like iterative

[1]Here the goal is to output a vector $\mathbf{M}'$ for which $\|\mathbf{M}-\mathbf{M}'\|_2 \le (1+\varepsilon)\|\mathbf{M}-\mathbf{M}_k\|_2$, where $\mathbf{M}_k$ is the best k-sparse approximation to $\mathbf{M}$, and ε is a constant.

linear system solving, eigenvalue finding, and so on, many algorithms used in practice are based on gradient descent methods, Krylov subspace methods, and the like. Those algorithms are essentially adaptive query algorithms designed with the primary goal of minimizing the number of queries.

Our model is also related to property testing, where one tries to infer properties of a large unknown object by (possibly adaptively) sampling a sublinear number of locations of that object. We argue that linear queries are a natural extension of sampling locations of an object, and that this is a natural "sampling model" not only because of the desired properties of the distributed, linear algebra, and streaming applications above, but sometimes also for physical constraints, e.g., in compressed sensing, where optical devices naturally capture linear measurements.

From a theoretical standpoint, any property testing algorithm, i.e., one that samples q entries of $\mathbf{M}$, can be implemented in our model with q linear queries. However, our model gives the algorithm much more flexibility. From a lower-bound perspective, as in the case of property testing [11], some of our lower bounds will be derived from communication complexity. However, not all of our bounds can be proved this way. For example, one notable result we show is an optimal lower bound on the number of queries needed to approximate the rank of $\mathbf{M} \in \mathbb{R}^{n\times n}$ up to a factor t by randomized, possibly adaptive algorithms; we show that $\frac{n}{t}+1$ queries are necessary and sufficient. A natural alternative way to prove this would be to give part of the matrix to Alice, part of the matrix to Bob, and have the players exchange the $\mathbf{M}^L\mathbf{v}^i$ and $\mathbf{M}^R\mathbf{v}^i$, where $\mathbf{M} = \mathbf{M}^L + \mathbf{M}^R$ and $\mathbf{M}^L$ is Alice's part and $\mathbf{M}^R$ is Bob's part. Then, if the 2-player randomized communication complexity of approximating the rank of $\mathbf{M}$ up to a factor of t were known to be $\Omega(n^2/t)$, we would obtain a nearly matching query lower bound of $\Omega(n/(t(b+\log n)))$, where b is the number of bits needed to specify the entries of $\mathbf{M}$ and the queries. However, it is unknown what the 2-player communication complexity of approximating the rank of $\mathbf{M}$ up to a factor t is over $\mathbb{R}$! We are not aware of any lower bound better than $\Omega(1)$ for constant t for this problem for adaptive queries. We note that for non-adaptive queries, there is an $\Omega(n^2)$ sketching lower bound over the reals given in [26], and an $\Omega(n^2/\log p)$ lower bound for finite fields (of size p) in [4]. There is also a property testing lower bound in [8], though such a lower bound makes additional assumptions on the input. Thus, our model gives a new lens to study this problem from, from which we are able to derive strong lower bounds for adaptive queries. Our techniques could be helpful for proving lower bounds in existing models, such as two-party communication complexity.

Our model is also related to linear decision tree complexity, see, e.g., [10] and [21], though such lower bounds typically involve just seeing a threshold applied to $\mathbf{M}\mathbf{v}^i$, and typically $\mathbf{M}$ is a vector. In our case, we observe the entire output vector $\mathbf{M}\mathbf{v}^i$.

An interesting twist in our model is that, in our formulation above, we are only allowed to query $\mathbf{M}$ via matrix-vector products *on the right*, i.e., of the form $\mathbf{M}\cdot\mathbf{v}^i$. One could ask if there are natural properties $\mathcal{P}$ of $\mathbf{M}$ for which the number q_L of queries one would need to make if querying $\mathbf{M}$ via queries of the form $(\mathbf{u}^1)^T\mathbf{M}, (\mathbf{u}^2)^T\mathbf{M}, \ldots, (\mathbf{u}^{q_L})^T\mathbf{M}$ can be significantly smaller than the number q_R of queries one would need to make if querying $\mathbf{M}$ via queries of the form $\mathbf{M}\mathbf{u}^1, \mathbf{M}\mathbf{u}^2, \ldots, \mathbf{M}\mathbf{u}^{q_R}$:

Question 2: Are there natural problems for which $q_L \ll q_R$?

We show that this is in fact the case, namely, if we can only multiply on the right, then it takes $\Omega(n/\log n)$ queries to determine if there is a *column* of a matrix $\mathbf{M} \in \{0,1\}^{n\times n}$ which is all 1s. However, if we can multiply on the left, then the single query $(1,1,\ldots,1)$ can determine this.

We study a few problems around Question 2, which is motivated from several perspectives. First, matrices might be stored on computers in a specific encoding, e.g., a sparse row format, from which it may be much easier to multiply on the right than on the left. Also, in compressed sensing, it may be natural for physical reasons to obtain linear combinations of columns rather than rows.

Another important question is how the query complexity depends on the *underlying field* for which matrix-vector products are performed. For example, might the query complexity be significantly different when the products are performed modulo 2 or over the reals?

Question 3: Is there a natural problem for which the query complexity in our model over $\mathbb{F}[2]$ is much larger than that over the reals?

Yet another important application of this model is to querying graphs. A natural question is which representation to use for the graph. For example, a natural representation of a graph on n vertices is through its adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, where $\mathbf{A}_{i,j} = 1$ if and only if $\{i, j\}$ occurs as an edge. A natural representation for a bipartite graph with n vertices in each part could be an $n \times n$ matrix $\mathbf{A}$ where $\mathbf{A}_{i,j} = 1$ iff there is an edge from the ith left vertex to the jth right vertex. Yet another representation could be the $\binom{n}{2} \times n$ edge-vertex incidence matrix, where the $\{i, j\}$-th row is either 0, or has exactly two ones, one in location i and one in location j. One often considers a signed edge-vertex incidence matrix, where one first arbitrarily fixes an ordering on the vertices and then the $\{i, j\}$-th entry has a 1 in the ith position and a -1 in the jth position if $i > j$; otherwise, positions i and j are swapped. Yet another possible representation of a graph is through its Laplacian.

Question 4: Do some natural representations of graphs admit much more efficient query algorithms for certain problems than other natural representations?

We note that in the data stream model, where one sees a long sequence of insertions and deletions to the edges of a graph, each of the matrix representations above can be simulated and so they lead to the same complexity. We will show, perhaps surprisingly, that in this model there can be an exponential difference in the query complexity for two different natural representations of a graph for the same problem.

We next get into the details of our results. We would like to stress that in this model, even for basic problems, it is not immediately obvious how to tackle them. As a puzzle for the reader, what is the query complexity of determining if a matrix $\mathbf{M} \in \mathbb{F}^{n \times n}$ is symmetric if one can only query vectors on the right? We will answer this later in the paper.

1.1 Formal Model and Our Results

We now describe our model and results formally in terms of an oracle. The oracle has a matrix $\mathbf{M} \in \mathbb{F}^{m \times n}$, for some underlying field $\mathbb{F}$ that we specify in each application. We can only query this matrix via matrix-vector products, i.e., we pick an arbitrary vector $\mathbf{x}$ and send it to the oracle, and the oracle will respond with a vector $\mathbf{y} = \mathbf{M} \cdot \mathbf{x}$. We focus our attention when the queries only occur on the right. Our goal is to approximate or test a number of properties of $\mathbf{M}$ with a minimal number of queries, i.e., to answer Question 1 for a large number of different application areas.

We study a number of problems as summarized in the Table 1. We assume $\mathbf{M}$ is an $m \times n$ matrix and $\varepsilon > 0$ is a parameter of the problem. The bounds hold for constant probability algorithms. In some problems, such as testing whether the matrix is a diagonal matrix, we always assume $m = n$, and in the graph testing problems we explicitly describe how the graph is represented using $\mathbf{M}$. Interestingly, we are able to prove very strong lower bounds for approximating the rank, which as described above, are unknown to hold for randomized communication complexity.

Motivated be streaming and statistics questions, we next study the query complexity of approximating the norm of each row of $\mathbf{M}$. We also study the computation of the majority or parity of each column or row of $\mathbf{M}$, the AND/OR of each column or row of $\mathbf{M}$, or equivalently, whether $\mathbf{M}$ has an all ones column or row, whether $\mathbf{M}$ has two identical columns or rows, and whether $\mathbf{M}$ contains an unusually large-normed row, i.e., a "heavy hitter". Here we show there are natural problems,

Table 1. Our Results

Problem	Query Complexity
Linear Algebra Problems	
Approximate Rank (for any $p' > p$ distinguishing Rank $\leq p$ from Rank p')	$p+1$ (Section 3.1)
Trace Estimation	$\Omega(n/\log n)$ (Section 3.2)
Symmetric Matrix/Diagonal Matrix	$O(1)$ (Sections 3.3 and 3.4)
Unitary Matrix	1 (Section 3.5)
Approximate Maximum Eigenvalue	$\Theta(\varepsilon^{-0.5}\log n)$ for adaptive queries, $\Theta(n)$ for non-adaptive queries ([29, 31, 35], Section 3.6)
Streaming and Statistics Problems	
All Ones Column	$\Theta(n)$ over $\mathbb{F}[2]$, $\Omega(n/\log n)$ over $\mathbb{R}$ (Section 4.1)
Two Identical Columns Two Identical Rows	$\Omega(n/m)$ $(m = \Omega(\log(n/\varepsilon)))$ $O(\log m)$ (Section 4.2)
Approximate Row Norms/Heavy Hitters	$O\left(\varepsilon^{-2}\log m\right)$ (Section 4.3)
Majority of Columns Majority of Rows	$\Omega(n/\log n)$ over $\mathbb{R}$ $O(1)$ over $\mathbb{R}$ (Section 4.4)
Parity of Columns Parity of Rows	$\Theta(n)$ $O(1)$ (Section 4.5)
Graph Problems	
Connectivity given Bipartite Adjacency Matrix Connectivity given Signed Edge-Vertex Matrix	$\Omega(n/\log n)$ (Section 5.1) $O(\text{polylog}(n))$ ([23], noted in Section 5.1)
Triangle Detection	$\Omega(n/\log n)$ (Section 5.2)

such as computing the parity of all columns, which can be solved with 1 query if sketching on the left, but require $\Omega(n)$ queries if sketching on the right, thus answering Question 2. We also answer Question 3, observing for the natural problem of testing if a row is all ones, a single deterministic query suffices over the reals but over $\mathbb{F}[2]$ this deterministically requires $\Omega(n)$ queries.

For graph problems, we first argue if the graph is presented as an $n \times n$ bipartite adjacency matrix $\mathbf{M}$, then it requires $\Omega(n/\log n)$ possibly adaptive queries to determine if the graph is connected. In contrast, if the graph is presented as an $n \times \binom{n}{2}$ signed vertex-edge incidence matrix, then polylog (n) non-adaptive queries suffices. This answers Question 4, showing that the type of representation of the graph is critical in this model. Motivated by a large body of recent work on triangle counting (see, e.g., [16] and the references therein), we also give strong negative results for this problem in our model, which, as with all of our lower bounds unless explicitly stated otherwise, hold even for algorithms which perform adaptive queries.

Followup Work. In [12], the authors considered other related problems in this model, including the important problem of linear regression and top eigenvalue estimation.

2 PRELIMINARIES

We use capital bold letters, e.g., $\mathbf{A}, \mathbf{B}, \mathbf{M}$, to denote matrices, and use lowercase bold letters, e.g., $\mathbf{x}, \mathbf{y}$, to denote column vectors. Sometimes we write a matrix as a list of column vectors in square brackets, e.g., $\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_n]$. We use calligraphic letters, e.g., $\mathcal{D}$, to denote probability distributions, and use $\mathbf{M} \leftarrow \mathcal{D}$ to denote that $\mathbf{M}$ is sampled from distribution $\mathcal{D}$. In particular, we

use $\mathcal{G}$ to denote a Gaussian distribution and $\mathbf{G}$ for a matrix whose entries are sampled from an independently and identically distributed (denoted as i.i.d. in the following) Gaussian distribution.

We call a matrix $\mathbf{M}$ i.i.d. Gaussian if each element is i.i.d. Gaussian. Let matrix $\mathbf{G}$ be a $p \times n$ i.i.d. Gaussian matrix, and $\mathbf{R}$ be an $n \times n$ rotation matrix, that is, $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. It is easy to check that $\mathbf{G} \times \mathbf{R}$ is still i.i.d. Gaussian, and has the same probability distribution of $\mathbf{G}$.

The *total variation distance*, sometimes called the *statistical distance*, between two probability measures P and Q is defined as

$$\mathrm{D}_{\mathrm{TV}}(P, Q) \stackrel{\text{def}}{=} \sup_A |P(A) - Q(A)|.$$

Let $\mathbf{X}$ be an $n \times m$ matrix with each row i.i.d. drawn from an m-variate normal distribution $N(0, \Sigma)$. Then the distribution of the $m \times m$ random matrix $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ is called the *Wishart distribution* with n degrees of freedom and covariance matrix Σ, denoted by $W_m(n, \Sigma)$. The distribution of eigenvalues of $\mathbf{A}$ is characterized in the following lemma:

Lemma 1 (Corollary 3.2.19 in [24]). *If $\mathbf{A}$ is $W_m(n, \lambda I_m)$, with $n > m-1$, the joint density function of the eigenvalues $\Lambda = (\lambda_1, \ldots, \lambda_m)$ of $\mathbf{A}$ (in descending order) is*

$$f(\Lambda) = \frac{\pi^{m^2/2}}{(2\lambda)^{mn/2}\Gamma_m(m/2)\Gamma_n(n/2)} \exp\left(-\frac{1}{2\lambda}\sum_{i=1}^m \lambda_i\right) \prod_{i=1}^m \lambda_i^{(n-m-1)/2} \prod_{1 \le i < j \le m} (\lambda_i - \lambda_j).$$

In particular, for $\lambda = 1$ and $n = m$, $\exists$ a constant Z_m independent from $\lambda_1, \ldots, \lambda_m$, such that

$$f(\Lambda) = \frac{1}{Z_m} \exp\left(-\frac{1}{2}\sum_{i=1}^m \lambda_i\right) \prod_{i=1}^m \lambda_i^{-1/2} \prod_{1 \le i < j \le m} (\lambda_i - \lambda_j).$$

3 LINEAR ALGEBRA PROBLEMS

In this part, we present our lower bound for rank approximation in Section 3.1. After that, we provide our results about trace estimation in Section 3.2, testing symmetric matrices in Section 3.3, testing diagonal matrices in Section 3.4, testing unitary matrices in Section 3.5, and approximating the maximum eigenvalue in Section 3.6.

3.1 Lower Bound for Rank Approximation

In this section, we discuss how to approximate the rank of a given matrix $\mathbf{M}$ over the reals when the queries consist of right multiplication by vectors. A naïve algorithm to learn the rank is to pick random Gaussian query vectors non-adaptively. In order to approximate the rank, that is, to distinguish whether $\mathrm{rank}(\mathbf{M}) \le p$ or $\mathrm{rank}(\mathbf{M}) \ge p + 1$, this algorithm needs at least $p + 1$ queries, and it is not hard to see that the algorithm succeeds with probability 1. Indeed, if $\mathbf{M}$ is the unknown $n \times n$ matrix, and $\mathbf{H} \in \mathbb{R}^{n \times (p+1)}$ is the random Gaussian query matrix, then we can write $\mathbf{M}$ in its thin singular value decomposition as $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V} \in \mathbb{R}^{n \times k}$ have orthonormal columns, and $\Sigma \in \mathbb{R}^{k \times k}$ has positive diagonal entries. Here, $k = \mathrm{rank}(\mathbf{M})$. We have that $\mathrm{rank}(\mathbf{M} \cdot \mathbf{H}) = \mathrm{rank}(\mathbf{V}^T\mathbf{H})$. Since $\mathbf{V}$ has orthonormal columns, it can be extended to a rotation matrix $\mathbf{V}' \in \mathbb{R}^{n \times n}$ by adding $n - k$ orthonormal columns. We have $\mathbf{V}'^T\mathbf{H}$ is a random Gaussian matrix by rotational invariance of the Gaussian distribution. $\mathbf{V}^T\mathbf{H}$ is actually the first k rows of $\mathbf{V}'^T\mathbf{H}$, so it is also a random Gaussian matrix. Thus, $\mathrm{rank}(\mathbf{V}^T\mathbf{H})$ is the minimum of $p + 1$ and the rank of $\mathbf{M}$ with probability 1.

In the following, we will show that we cannot expect anything better. We will first show for non-adaptive queries, at least $p + 1$ queries are necessary to learn the approximate rank. Then we generalize our results to adaptive queries. Our results hold for randomized algorithms by applying Yao's minimax principle.

3.1.1 Non-Adaptive Query Protocols.

Theorem 1. *Let constant $\varepsilon > 0$ be the error tolerance and let $\mathbf{M}$ be an $n \times n$ oracle matrix and suppose to start that we make non-adaptive queries. For integer $p < p' \le n$, at least $p+1$ queries are necessary to distinguish* $\mathrm{rank}(\mathbf{M}) \le p$ *from* $\mathrm{rank}(\mathbf{M}) \ge p'$ *with advantage* $\ge \varepsilon$.

Proof. Given any algorithm distinguishing $\mathrm{rank}(\mathbf{M}) \le p$ from $\mathrm{rank}(\mathbf{M}) \ge p'$ for some $p' < n$, we can determine whether a $p' \times p'$ matrix $\mathbf{M}'$ has full rank p' or $\mathrm{rank}(\mathbf{M}') \le p$, by padding $\mathbf{M}'$ to an $n \times n$ matrix $\mathbf{M}$. Therefore, in what follows, it suffices to prove the lower bound for two $n \times n$ matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ where $\mathrm{rank}(\mathbf{M}_1) \le p$ and $\mathrm{rank}(\mathbf{M}_2) = n$:

(1) $\mathbf{M}_1 = \mathbf{U} \times \mathbf{G}^T$;
(2) $\mathbf{M}_2 = \mathbf{U} \times \mathbf{G}^T + \frac{1}{Z(n)} \cdot \mathbf{U}^\perp \times \mathbf{H}^T$.

Here $\mathbf{U}$ has p columns and $\mathbf{U}^\perp$ has $(n-p)$ columns such that $[\mathbf{U}, \mathbf{U}^\perp]$ forms an $n \times n$ random orthonormal basis, $\mathbf{G}^T$ and $\mathbf{H}^T$ are $p \times n$ and $(n-p) \times n$ matrices whose entries are sampled i.i.d. from the standard Gaussian distribution, and $Z(n)$ is a function in n which will be specified later. Since $\mathbf{M}_2 = [\mathbf{U}, \mathbf{U}^\perp] \times [\begin{smallmatrix} \mathbf{G}^T \\ \frac{1}{Z(n)}\mathbf{H}^T \end{smallmatrix}]$, it immediately follows that $\mathrm{rank}(\mathbf{M}_1) \le p$ and $\mathrm{rank}(\mathbf{M}_2) = n$ with probability 1. Then we assume $\mathrm{rank}(\mathbf{M}_2) = n$ and discuss the query lower bound for distinguishing $\mathbf{M}_1$ from $\mathbf{M}_2$.

Given $\mathbf{M} \in \{\mathbf{M}_1, \mathbf{M}_2\}$, without loss of generality we denote the q non-adaptive queries with an $n \times q$ orthonormal[2] matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_q]$, where $q \le p$ and each $n \times 1$ column vector $\mathbf{v}_i$ is a query to the oracle of matrix $\mathbf{M}$ which gets response $\mathbf{M} \cdot \mathbf{v}_i$, for $i \in [q]$. Then, it suffices to show that the following two distributions are hard to distinguish:

(1) $\mathbf{M}_1 \times \mathbf{V} \equiv \mathbf{UW}$, where $\mathbf{W} = \mathbf{G}^T\mathbf{V}$;
(2) $\mathbf{M}_2 \times \mathbf{V} \equiv \mathbf{UW} + \frac{1}{Z(n)} \cdot \mathbf{U}^\perp\mathbf{W}'$, where $\mathbf{W}' = \mathbf{H}^T\mathbf{V}$.

Note that $[\mathbf{U}, \mathbf{U}^\perp]$ is orthonormal, and hence $\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$, $(\mathbf{U}^\perp)^T\mathbf{U}^\perp = \mathbf{I}_{n-p}$, $\mathbf{U}^T\mathbf{U}^\perp = \mathbf{0}_{p\times(n-p)}$. We introduce Lemma 2 to eliminate $\mathbf{U}, \mathbf{U}^\perp$ in the representation of $\mathbf{M} \times \mathbf{V}$.

Lemma 2. *For $\mathbf{M}_1, \mathbf{M}_2$ and $\mathbf{V}$ defined as above,*

$$\mathsf{D}_{\mathsf{TV}}(\mathbf{M}_1\mathbf{V}, \mathbf{M}_2\mathbf{V}) = \mathsf{D}_{\mathsf{TV}}\left((\mathbf{M}_1\mathbf{V})^T\mathbf{M}_1\mathbf{V}, (\mathbf{M}_2\mathbf{V})^T\mathbf{M}_2\mathbf{V}\right)$$

Proof. The direction $\mathsf{D}_{\mathsf{TV}}(\mathbf{M}_1\mathbf{V}, \mathbf{M}_2\mathbf{V}) \ge \mathsf{D}_{\mathsf{TV}}((\mathbf{M}_1\mathbf{V})^T\mathbf{M}_1\mathbf{V}, (\mathbf{M}_2\mathbf{V})^T\mathbf{M}_2\mathbf{V})$ is trivial by the data-processing inequality (i.e., for every $\mathbf{X}, \mathbf{Y}$ and random function f, $\mathsf{D}_{\mathsf{TV}}(\mathbf{X}, \mathbf{Y}) \ge \mathsf{D}_{\mathsf{TV}}(f(\mathbf{X}), f(\mathbf{Y}))$). In what follows, we only prove the other direction.

For a random matrix $\mathbf{M}$ sampled as $\mathbf{M}_1$ or $\mathbf{M}_2$, we will design a random transformation process where the input is a random sample from $\mathbf{V}^T\mathbf{M}^T\mathbf{MV}$, and the output has the same distribution as $\mathbf{MV}$. If we have such a transformation, we have $\mathsf{D}_{\mathsf{TV}}(\mathbf{M}_1\mathbf{V}, \mathbf{M}_2\mathbf{V}) \le \mathsf{D}_{\mathsf{TV}}((\mathbf{M}_1\mathbf{V})^T\mathbf{M}_1\mathbf{V}, (\mathbf{M}_2\mathbf{V})^T\mathbf{M}_2\mathbf{V})$.

The transformation process works as follow: given a $q \times q$ sample matrix $\mathbf{X}$ from $\mathbf{V}^T\mathbf{M}^T\mathbf{MV}$, let its singular value decomposition be $\mathbf{X} = \mathbf{B}\mathbf{\Lambda}\mathbf{B}^T$. We generate an $n \times q$ random orthonormal matrix $\mathbf{A}$, and output an $n \times q$ matrix $\mathbf{Y} = \mathbf{A}\mathbf{\Lambda}^{1/2}\mathbf{B}^T$. We then show the matrix $\mathbf{Y}$ follows the same distribution of $\mathbf{MV}$.

First, let us consider the case $\mathbf{M} = \mathbf{M}_1 = \mathbf{U} \times \mathbf{G}^T$. Let the singular value decomposition of $\mathbf{W} = \mathbf{G}^T\mathbf{V}$ be $\mathbf{W} = \mathbf{C}\mathbf{\Sigma}\mathbf{D}^T$ where $\mathbf{C}$ is a $p \times q$ matrix, and $\mathbf{\Sigma}, \mathbf{D}$ are $q \times q$ matrices. Then, we have $\mathbf{V}^T\mathbf{M}_1^T\mathbf{M}_1\mathbf{V} = \mathbf{D}\mathbf{\Sigma}^2\mathbf{D}^T$. Thus, we know $\mathbf{\Sigma} = \mathbf{\Lambda}^{1/2}$ and $\mathbf{D} = \mathbf{B}$. We do not know the left matrix $\mathbf{C}$, but $\mathbf{UC}$ is an $n \times q$ random orthonormal matrix since the columns of $\mathbf{U}$ come from a basis for a

[2]Non-orthonormal queries can be made orthonormal using a change of basis in post-processing.

random p dimensional subspace. In addition, the random matrix $\mathbf{UC}$ is independent of $\mathbf{D}$. Thus, the matrix $\mathbf{A}$ generated by the transformation process has the same distribution as $\mathbf{UC}$ which means our output matrix $\mathbf{Y}$ follows the distribution of $\mathbf{M}_1\mathbf{V}$ if the input matrix is sampled from $\mathbf{V}^T\mathbf{M}_1^T\mathbf{M}_1\mathbf{V}$.

Second, let us consider the case $\mathbf{M} = \mathbf{M}_2 = \mathbf{U} \times \mathbf{G}^T + \frac{1}{Z(n)} \cdot \mathbf{U}^\perp \times \mathbf{H}^T$. The idea is similar as the first case. Let the singular value decomposition of $\mathbf{M}_2\mathbf{V}$ be $\mathbf{M}_2\mathbf{V} = \mathbf{C}\Sigma\mathbf{D}^T$ where $\mathbf{C}$ is an $n \times q$ matrix, and Σ, $\mathbf{D}$ are $q \times q$ matrices. Then, we have $\mathbf{V}^T\mathbf{M}_2^T\mathbf{M}_2\mathbf{V} = \mathbf{D}\Sigma^2\mathbf{D}^T$. Thus, we know $\Sigma = \mathbf{\Lambda}^{1/2}$ and $\mathbf{D} = \mathbf{B}$. Consider an $n \times n$ uniformly random orthonormal matrix $\mathbf{R}$, we have $\mathbf{RM}_2\mathbf{V}$ has the same distribution as $\mathbf{M}_2\mathbf{V}$ since $[\mathbf{RU}, \mathbf{RU}^\perp]$ also forms an $n \times n$ random orthonormal basis. And $\mathbf{RC}$ is an $n \times q$ uniformly random orthonormal matrix which has the same distribution of $\mathbf{A}$, and also independent of $\mathbf{D}$. So our output matrix $\mathbf{Y}$ follows the distribution of $\mathbf{M}_2\mathbf{V}$ if the input matrix sampled from $\mathbf{V}^T\mathbf{M}_2^T\mathbf{M}_2\mathbf{V}$. □

Let $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_q), \mathbf{\Lambda}' = \mathrm{diag}(\lambda'_1, \ldots, \lambda'_q)$ be diagonal matrices such that $\mathbf{W}^T\mathbf{W} = \mathbf{A}^T\mathbf{\Lambda}\mathbf{A}$ and $\mathbf{W}^T\mathbf{W} + \frac{(\mathbf{W}')^T\mathbf{W}'}{Z^2(n)} = \mathbf{B}^T\mathbf{\Lambda}'\mathbf{B}$ for orthonormal matrices $\mathbf{A}$ and $\mathbf{B}$. We assume both $\lambda_1, \ldots, \lambda_q$ and $\lambda'_1, \ldots, \lambda'_q$ are sorted in decreasing order. Using Lemma 2, it suffices to prove an upper bound for $\mathsf{D}_{\mathsf{TV}}(\mathbf{\Lambda}, \mathbf{\Lambda}')$ as follows:

$$\begin{aligned}
&\mathsf{D}_{\mathsf{TV}}\left(\mathbf{UW}, \mathbf{UW} + \frac{\mathbf{U}^\perp\mathbf{W}'}{Z(n)}\right) \\
&= \mathsf{D}_{\mathsf{TV}}\left((\mathbf{UW})^T(\mathbf{UW}), \left(\mathbf{UW} + \frac{\mathbf{U}^\perp\mathbf{W}'}{Z(n)}\right)^T\left(\mathbf{UW} + \frac{\mathbf{U}^\perp\mathbf{W}'}{Z(n)}\right)\right) \\
&= \mathsf{D}_{\mathsf{TV}}\left(\mathbf{W}^T\mathbf{W}, \mathbf{W}^T\mathbf{W} + \frac{(\mathbf{W}')^T\mathbf{W}'}{Z^2(n)}\right) \le \mathsf{D}_{\mathsf{TV}}(\mathbf{\Lambda}, \mathbf{\Lambda}').
\end{aligned}$$

The last inequality uses the similar idea to Lemma 2. Given any sample $\mathbf{X}$ from $\mathbf{\Lambda}$ or $\mathbf{\Lambda}'$, we can generate a $q \times q$ uniformly random orthonormal matrix $\mathbf{C}$, and the matrix $\mathbf{C}^T\mathbf{XC}$ follows the same distribution of either $\mathbf{W}^T\mathbf{W}$ or $\mathbf{W}^T\mathbf{W} + \frac{(\mathbf{W}')^T\mathbf{W}'}{Z^2(n)}$, which depends on whether $\mathbf{X}$ comes from $\mathbf{\Lambda}$ or $\mathbf{\Lambda}'$.

We then apply Weyl's inequality as follows:

Lemma 3 (Weyl's Inequality, [39, 42]). *Let $\mathbf{A} = \mathbf{B} + \mathbf{C}$ where $\mathbf{B}, \mathbf{C}$ are $n \times n$ Hermitian matrices, with their respective eigenvalues μ_i, ν_i, ρ_i ordered as follows:*

$$\mathbf{A}: \mu_1 \ge \cdots \ge \mu_n,$$
$$\mathbf{B}: \nu_1 \ge \cdots \ge \nu_n,$$
$$\mathbf{C}: \rho_1 \ge \cdots \ge \rho_n.$$

Then the following inequalities hold: $\nu_i + \rho_n \le \mu_i \le \nu_i + \rho_1$ for $1 \le i \le n$.

For every $i \in [q]$, we have $\lambda'_i \in [\lambda_i - O(\frac{\|\mathbf{W}'\|_2^2}{Z^2(n)}), \lambda_i + O(\frac{\|\mathbf{W}'\|_2^2}{Z^2(n)})]$. Notice that $\mathbf{W}'$ is an $(n-p) \times q$ i.i.d. Gaussian matrix. Thus, we have $\|W'\|_2^2 \le \|W'\|_F^2$ which is the sum of $(n-p)q$ independent squared Gaussian random elements. Hence, it is bounded by $O((n-p)q)$ with high probability. More specifically, we have $Pr(\|W'\|_F^2 \ge (n-p)q(1+t)) \le \exp(-(n-p)qt^2/18)$ for all $t \in [0, 3]$ according to Example 2.12 in [38]. Recalling that $q \le p$, in what follows, we condition on the event $\lambda'_i \in [\lambda_i - O(\frac{np}{Z^2(n)}), \lambda_i + O(\frac{np}{Z^2(n)})]$.

We then show the gaps between eigenvalues λ_i are sufficiently large. Note that, since $\mathbf{G}^T$ is i.i.d. Gaussian and $\mathbf{V}$ is an orthonormal matrix, each row in $\mathbf{W} = \mathbf{G}^T\mathbf{V}$ is independently drawn from an

q-variate normal distribution; thus, the probability distribution of $\mathbf{W}^T\mathbf{W}$ is a Wishart distribution $W_q(p, I_q)$. It is enough to show the case where $q = p$. Since $\lambda_1, \ldots, \lambda_p$ is sorted in descending order, by Lemma 1, the density function of $\mathbf{\Lambda}$ is:

$$f(\mathbf{\Lambda}) = \frac{1}{Z_p} \exp\left(-\frac{1}{2}\sum_{i=1}^{p}\lambda_i\right)\prod_{i=1}^{p}\lambda_i^{-1/2}\prod_{1\le i<j\le p}(\lambda_i - \lambda_j). \tag{1}$$

Let $\mathcal{E}$ denote the event that $\lambda_p \ge \frac{0.01}{\sqrt{n}}$ and $\forall 1 \le i < j \le p, \lambda_i - \lambda_j \ge \gamma = 2^{-\Theta(p^2 \log p)}$.

Lemma 4. *For $\mathbf{W}^T\mathbf{W}$ defined as above and sufficiently small $\gamma = 2^{-\Theta(p^2 \log n)}$, $\Pr[\mathcal{E}] > 0.9$.*

Proof. By Equation (2) in [34] we know that for the smallest singular value λ_p, we have $\Pr[\sqrt{n}\lambda_p \ge y] = \exp(-(y^2/2 + y))$. Thus, for $y = 0.01$ and $\mathcal{E}_0 \stackrel{\text{def}}{=} \{\lambda_p \ge 0.01/\sqrt{n}\}$, we get:

$$\Pr[\mathcal{E}_0] = \Pr\left[\lambda_p \ge \frac{0.01}{\sqrt{n}}\right] = \exp(-0.01005) > 0.99.$$

Also, we note that, for every i, $\Pr[|\lambda_i| \le 100n] \ge 1 - 2\exp(-32n)$, by setting $t = 8\sqrt{n}$ in Corollary 5.35 of [37]. In what follows, we condition on the event $\mathcal{E}_0'$ that $|\lambda_i| \le 100n$ for every $i \in [p]$.

Then we consider the joint distribution μ of $\lambda_1, \ldots, \lambda_p$ in $\mathbf{\Lambda}$. Let $\mathcal{E}_i \stackrel{\text{def}}{=} \{\lambda_i - \lambda_{i+1} < \gamma\}$ be the event that λ_i and λ_{i+1} has a gap smaller than γ. Thus, $\mathcal{E} = \mathcal{E}_0 \wedge (\wedge_{i=1}^{p-1}\overline{\mathcal{E}_i})$. To lower bound $\Pr[\mathcal{E}]$, we need to upper bound the probability of $\mathcal{E}_i$ for $1 \le i \le p-1$.

Let f be the density function of μ as in (1), and let $\mathsf{Leb}(\cdot)$ be the Lebesgue measure in n dimensions. Then, for every i,

$$\Pr[\mathcal{E}_i \mid \mathcal{E}_0'] = \mu(\lambda_i - \lambda_{i+1} < \gamma) \le \mathsf{Leb}(\lambda_i - \lambda_{i+1} < \gamma) \cdot |f|_\infty = O(\gamma/n) \cdot |f|_\infty$$

Note that conditioning on $\mathcal{E}_0$ such that $\lambda_p \ge 0.01/\sqrt{n}$, the density function f is bounded as:

$$|f|_\infty \le O\left(\exp\left(-\frac{1}{2}\lambda_1\right)\left(100\sqrt{n}\right)^{p/2}\lambda_1^{p^2/2}\right) = 2^{O(p^2 \log n)}$$

As a result, we get $\Pr[\mathcal{E}_i \wedge \mathcal{E}_0 \mid \mathcal{E}_0'] \le \gamma \cdot 2^{O(p^2 \log n)}$.

Therefore, the probability of $\mathcal{E}$ is lower bounded for sufficiently small $\gamma = 2^{-\Theta(p^2 \log n)}$,

$$\begin{aligned}
\Pr[\mathcal{E}] &\ge \Pr[\mathcal{E}_0'] \cdot \Pr\left[\mathcal{E}_0 \wedge \left(\wedge_{i=1}^{p-1}\overline{\mathcal{E}_i}\right) \mid \mathcal{E}_0'\right] \\
&\ge \Pr[\mathcal{E}_0'] \cdot \left(\Pr\left[\mathcal{E}_0 \mid \mathcal{E}_0'\right] - \sum_{i=1}^{p-1}\Pr\left[\mathcal{E}_i \wedge \mathcal{E}_0 \mid \mathcal{E}_0'\right]\right) \\
&> (1 - 2p\exp(-32n)) \cdot \left(0.99 - (p-1)\gamma \cdot 2^{O(p^2 \log n)}\right) > 0.9.
\end{aligned}$$

□

Conditioned on event $\mathcal{E}$ and recalling that $\lambda_i' \in [\lambda_i - O(\frac{np}{Z^2(n)}), \lambda_i + O(\frac{np}{Z^2(n)})]$, the probability density of $\mathbf{\Lambda}'$ has only a negligible difference from that of $\mathbf{\Lambda}$, since the small disturbance of eigenvalues

is dominated by the corresponding terms in $f(\mathbf{\Lambda})$.

$$\begin{aligned}
\frac{f(\mathbf{\Lambda}')}{f(\mathbf{\Lambda})} &= \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{p}\lambda_i'\right)\prod_{i=1}^{p}\lambda_i'^{-1/2}\prod_{1\le i<j\le p}(\lambda_i'-\lambda_j')}{\exp\left(-\frac{1}{2}\sum_{i=1}^{p}\lambda_i\right)\prod_{i=1}^{p}\lambda_i^{-1/2}\prod_{1\le i<j\le p}(\lambda_i-\lambda_j)} \\
&\le \exp\left(\frac{p\cdot np}{Z^2(n)}\right)\left(\frac{\lambda_p-\frac{np}{Z^2(n)}}{\lambda_p}\right)^{-p/2}\prod_{1\le i<j\le p}\frac{\lambda_i-\lambda_j+\frac{2np}{Z^2(n)}}{\lambda_i-\lambda_j} \\
&\le \exp\left(\frac{np^2}{Z^2(n)}\right)\cdot\left(1+\frac{np}{\lambda_p\cdot Z^2(n)}\right)^{p}\left(1+\frac{2np}{Z^2(n)\cdot\min_{i\ne j}|\lambda_i-\lambda_j|}\right)^{p(p-1)/2} \\
&\le \exp\left(\frac{np^2}{Z^2(n)}\right)\cdot\left(1+\frac{100\sqrt{n}\cdot np}{Z^2(n)}\right)^{p}\left(1+\frac{2np}{Z^2(n)\cdot\gamma}\right)^{p(p-1)/2} = 1+O\left(\frac{np^3\gamma^{-1}}{Z^2(n)}\right)
\end{aligned}$$

Similarly, we can prove $f(\mathbf{\Lambda}')/f(\mathbf{\Lambda}) \ge 1 - O\left(np^3\gamma^{-1}/Z^2(n)\right)$. Thus, the total variation distance between $\mathbf{\Lambda}$ and $\mathbf{\Lambda}'$ conditioned on $\mathcal{E}$ is $\mathrm{D_{TV}}(\mathbf{\Lambda}, \mathbf{\Lambda}' \mid \mathcal{E}) \le (np^3\gamma^{-1}/Z^2(n)) = (1/n^2)$ for sufficiently large $Z(n) \ge (np)^{1.5}\gamma^{-0.5} = 2^{\Theta(p^2\log n)}$. Thus, for sufficiently large n, we have:

$$\mathrm{D_{TV}}\left(\mathbf{\Lambda}, \mathbf{\Lambda}'\right) \le \Pr[\overline{\mathcal{E}}] + \Pr[\mathcal{E}]\cdot\mathrm{D_{TV}}(\mathbf{\Lambda}, \mathbf{\Lambda}' \mid \mathcal{E}) \le 0.1 + O(1/n^2) < 0.11.$$

Therefore, with as many as $q = p$ non-adaptive queries to the oracle matrix $\mathbf{M}$, the two distributions $\mathbf{M}_1$ and $\mathbf{M}_2$ cannot be distinguished with advantage greater than 0.11. At least $p+1$ queries are necessary to distinguish those two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ of rank $\le p$ and rank n, respectively.

Indeed, the above argument holds for every constant advantage ε if $y = \varepsilon/3$, $t > \sqrt{12n/\varepsilon}$, and γ is sufficiently small in the proof of Lemma 4, and letting $Z(n)$ be sufficiently large. □

3.1.2 Equivalence between Adaptive and Non-Adaptive Protocols. Now, we consider the adaptive query matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_q]$ where $\mathbf{v}_i$ is the ith query vector. Without loss of generality, we can assume that $\forall i, \mathbf{v}_i$ is a unit vector and it is orthogonal to query vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$. This gives us the following formal definition of an adaptive query protocol:

Definition 1. For a target matrix $\mathbf{M}$, an adaptive query protocol P will output a sequence of query vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots$. It is called a *normalized adaptive protocol* if for any i, the query vector $\mathbf{v}_i$ output by P satisfies

(1) $\mathbf{v}_i$ is a unit vector;
(2) $\mathbf{v}_i$ is orthogonal to the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$;
(3) $\mathbf{v}_i$ is deterministically determined by $\mathbf{M} \times [\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}]$.

Let P^{std} be a standard protocol which outputs $\mathbf{e}_1, \mathbf{e}_2, \ldots$ where $\mathbf{e}_i$ is the ith standard basis vector. We then show that adaptivity is unnecessary by proving that P^{std} has the same power as any normalized adaptive protocol to distinguish the matrix $\mathbf{M}_1$ and $\mathbf{M}_2$ defined in the previous subsection.

More formally, we show the following lemma for matrix $\mathbf{M}_2$:

Lemma 5. *Fix any $n \times n$ orthogonal matrix $[\mathbf{U}, \mathbf{U}^{\perp}]$ and any normalized adaptive protocol P. Consider $\mathbf{M}_2 = \mathbf{U} \times \mathbf{G}^T + \frac{1}{\mathrm{poly}(n)} \cdot \mathbf{U}^{\perp} \times \mathbf{H}^T$ where $\mathbf{G}^T$ be a $p \times n$ i.i.d. Gaussian matrix, and $\mathbf{H}^T$ be a $(n-p) \times n$ i.i.d. Gaussian matrix. Let matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_q]$ and $\mathbf{V}^{std} = [\mathbf{e}_1, \ldots, \mathbf{e}_q]$ be the query matrix output by protocol P and P^{std}, correspondingly. We have the matrix $\mathbf{M}_2\mathbf{V}$ has the same distribution as $\mathbf{M}_2\mathbf{V}^{std}$.*

Proof. Since the matrix $\mathbf{G}^T \cdot \mathbf{V}^{std}$ and $\mathbf{H}^T \cdot \mathbf{V}^{std}$ is i.i.d. Gaussian, that is, every element in two matrices is from standard Gaussian distribution and independent of each other, it is enough to show both $\mathbf{G}^T \cdot \mathbf{V}$ and $\mathbf{H}^T \cdot \mathbf{V}$ are i.i.d. Gaussian. In the following, we will show $\mathbf{G}^T \cdot \mathbf{V}$ are i.i.d. Gaussian and independent of $\mathbf{H}^T \cdot \mathbf{V}$, and the similar argument also holds for $\mathbf{H}^T \cdot \mathbf{V}$.

Let $\mathbf{V}_i = [\mathbf{v}_1, \ldots, \mathbf{v}_i]$ and $\mathbf{V}_i^{std} = [\mathbf{e}_1, \ldots, \mathbf{e}_i]$. Note that $\mathbf{v}_1, \ldots, \mathbf{v}_q$ are unit vectors and orthogonal to each other. We first define orthogonal rotation matrices $\mathbf{R}_1, \mathbf{R}_2, \ldots$ recursively as follows. The matrix $\mathbf{R}_1$ will take $\mathbf{v}_1$ to $\mathbf{e}_1$. The matrix $\mathbf{R}_i$ will take $\mathbf{e}_j$ to $\mathbf{e}_j$ for any $j < i$ and takes $\mathbf{R}_{i-1} \cdots \mathbf{R}_1 \mathbf{v}_i$ to $\mathbf{e}_i$. Note, for any i, the orthogonal rotation matrix which satisfies the condition is not unique. We can arbitrary choose any orthogonal rotation matrix $\mathbf{R}_i$ deterministically and $\mathbf{R}_i$ only depends on the first i query vectors. We have $\mathbf{R}_i \cdots \mathbf{R}_1 \mathbf{V}_i = \mathbf{V}_i^{std}$ for any $i \le q$, and $\mathbf{G}^T\mathbf{V} = \mathbf{G}^T \cdot \mathbf{R}_1^{-1} \cdots \mathbf{R}_q^{-1} \cdot \mathbf{V}^{std}$. Define matrix $\mathbf{GR}_i \triangleq \mathbf{G}^T \cdot \mathbf{R}_1^{-1} \cdots \mathbf{R}_i^{-1}$. In the following, we use induction to show for any $i \le q$, $\mathbf{GR}_i$ is i.i.d. Gaussian and independent of $\mathbf{H}^T \cdot \mathbf{V}$. It is enough to show that, for any fixed H, for any $i \le q$, $\mathbf{GR}_i$ is i.i.d. Gaussian.

For $i = 1$, since $\mathbf{R}_1$ is determined by $\mathbf{v}_1$ which is independent of $\mathbf{G}^T$ and $\mathbf{R}_1$ is an orthogonal matrix, $\mathbf{GR}_1 = \mathbf{G}^T\mathbf{R}_1^{-1}$ is i.i.d. Gaussian.

Now, suppose $\mathbf{GR}_i$ is i.i.d. Gaussian, we will prove $\mathbf{GR}_{i+1} = \mathbf{GR}_i \cdot \mathbf{R}_{i+1}^{-1}$ is also i.i.d. Gaussian. On the one hand, $\mathbf{R}_{i+1}$ is determined by $\mathbf{v}_1, \ldots, \mathbf{v}_{i+1}$ which are determined by the response of the first i queries, that is, determined by $\mathbf{M_2V}_i$. We have

$$\begin{aligned}
&\mathbf{M_2V}_i \\
&= \mathbf{U}\mathbf{G}^T\mathbf{R}_1^{-1} \cdots \mathbf{R}_i^{-1}\mathbf{V}_i^{std} + \tfrac{1}{\mathrm{poly}(n)}\mathbf{U}^\perp\mathbf{H}^T\mathbf{R}_1^{-1} \cdots \mathbf{R}_i^{-1}\mathbf{V}_i^{std} \\
&= (\mathbf{U} \times (\mathbf{G}^T\mathbf{R}_1^{-1} \cdots \mathbf{R}_i^{-1}) + \tfrac{1}{\mathrm{poly}(n)}\mathbf{U}^\perp \times (\mathbf{H}^T\mathbf{R}_1^{-1} \cdots \mathbf{R}_i^{-1})) \cdot \mathbf{V}_i^{std}
\end{aligned}$$

It means $\mathbf{R}_{i+1}$ is determined by the first i columns of matrix $\mathbf{GR}_i = \mathbf{G}^T\mathbf{R}_1^{-1} \cdots \mathbf{R}_i^{-1}$ and $\mathbf{H}^T\mathbf{R}_1^{-1} \cdots \mathbf{R}_i^{-1}$. Note by the inductive assumption, $\mathbf{GR}_i$ is i.i.d. Gaussian. Therefore, $\mathbf{R}_{i+1}$ is independent of the last $n - i$ columns of $\mathbf{GR}_i$.

On the other hand, $\mathbf{R}_{i+1}\mathbf{e}_j = \mathbf{e}_j$ for any $j \le i$, and thus $\mathbf{R}_{i+1}^{-1} = \begin{bmatrix} I_i & 0 \\ 0 & \mathbf{R}' \end{bmatrix}$ where I_i is the $i \times i$ identity matrix. Note the matrix $\mathbf{R}'$ is actually determined by the protocol, $\mathbf{U}$, $\mathbf{U}^\perp$, $\mathbf{H}$ and also the first i columns of $\mathbf{GR}_i$, but it is independent of the last $n - i$ columns of $\mathbf{GR}_i$. Consequently, in the multiplication of $\mathbf{GR}_i \times \mathbf{R}_{i+1}^{-1}$, the first i columns are the same as those in $\mathbf{GR}_i$. For the other $n-i$ columns, the ath element of jth column is $\sum_{b \ge i+1} gr_{ab} r'_{b,j}$ where $gr_{ab}, r'_{b,j}$ are the elements in $\mathbf{GR}_i, \mathbf{R}'$ correspondingly. Since $r'_{b,j}$ is independent of the last $n-i$ columns of $\mathbf{GR}_i$, it is independent of gr_{ab} when $b \ge i + 1$. Since $\mathbf{GR}_i$ is i.i.d Gaussian and $\mathbf{R}'$ is an orthogonal matrix, the last $n - i$ columns of $\mathbf{GR}_{i+1}$ is also i.i.d. Gaussian and independent of the first i columns. Therefore, we show $\mathbf{GR}_{i+1} = \mathbf{G}^T \cdot \mathbf{R}_1^{-1} \cdots \mathbf{R}_{i+1}^{-1}$ is still i.i.d. Gaussian.

By induction $\mathbf{G}^T\mathbf{V}$ is i.i.d. Gaussian, and independent of $\mathbf{H}^T\mathbf{V}$. This finishes our proof. □

Obliviously, the same also holds for $\mathbf{M}_1$. Combining these results and Theorem 1, together with Yao's minimax principle [41],

Theorem 2. *Let constant $\varepsilon > 0$ be the error tolerance and let $\mathbf{M}$ be an $n \times n$ oracle matrix with adaptive queries. For integer $p < p' \le n$, at least $p + 1$ queries are necessary for any randomized algorithm to distinguish whether* $\mathsf{rank}(\mathbf{M}) \le p$ *or* $\mathsf{rank}(\mathbf{M}) \ge p'$ *with advantage* $\ge \varepsilon$.

3.2 Lower Bound for Trace Estimation

In this section, we lower bound the number of queries needed to approximate the trace $\mathrm{tr}(\mathbf{M})$ of a matrix $\mathbf{M}$. In particular we reduce this problem to triangle detection as will be proved in Theorem 16. For the trace estimation problem, Avron and Toledo [6] analyzed the convergence of

randomized trace estimators via a similar matrix vector products framework. In their model, for an unknown matrix $\mathbf{M}$, they can access it via $\mathbf{v}^T\mathbf{M}\mathbf{v}$; while in our model, we only consider the right multiplication of the form $\mathbf{M}\mathbf{v}$.

Theorem 3. *For any integer $C > 0$ and symmetric $n \times n$ matrix $\mathbf{M}$ with entries in $\{0, 1, 2, \ldots, n^3\}$, the number of possibly adaptively chosen query vectors, with entries in $\{0, 1, 2, \ldots, n^C\}$, needed to approximate $\mathrm{tr}(\mathbf{M})$ up to any relative error, is $\Omega(n/\log n)$.*

Proof. Suppose we had a possibly adaptive query algorithm making $q(n)$ queries which for a symmetric matrix $\mathbf{M}$, could approximate $\mathrm{tr}(\mathbf{M})$ up to any relative error. If $\mathbf{M} = \mathbf{A}^3$ for a symmetric matrix $\mathbf{A}$, we can run the trace esimation algorithm on $\mathbf{M}$ as follows: if $\mathbf{x}_1$ is the first query, we compute $\mathbf{A}\mathbf{x}_1$, then $\mathbf{A}(\mathbf{A}\mathbf{x}_1)$, then $\mathbf{A}(\mathbf{A}(\mathbf{A}\mathbf{x}_1)) = \mathbf{A}^3\mathbf{x}_1$. This then determines the second query $\mathbf{x}_2$, and we similarly compute $\mathbf{A}\mathbf{x}_2$, then $\mathbf{A}(\mathbf{A}\mathbf{x}_2)$, then $\mathbf{A}(\mathbf{A}(\mathbf{A}\mathbf{x}_2)) = \mathbf{A}^3\mathbf{x}_2$, and so on. Thus, given only query access to $\mathbf{A}$, we can simulate the algorithm on $\mathbf{M} = \mathbf{A}^3$ with $3q(n)$ adaptive queries.

Now, it is well known that for an undirected graph G with adjacency matrix $\mathbf{A}$, the trace $\mathrm{tr}(\mathbf{A}^3)/6$ is the number of triangles in G. By the argument above, it follows that with $3q(n)$ queries to $\mathbf{A}$, we can determine if G has a triangle or has no triangles. On the other hand, by Theorem 16 below, at least $\Omega(n/\log n)$ queries to $\mathbf{A}$ are necessary for any adaptive algorithm to decide if there is a triangle in G. Therefore, $3q(n) = \Omega(n/\log n)$ and hence we complete the proof with $q(n) = \Omega(n/\log n)$. □

3.3 Deciding if M is a Symmetric Matrix

Theorem 4. *Given an $n \times n$ matrix $\mathbf{M}$ over any finite field or over fields $\mathbb{R}$ or $\mathbb{C}$, $O(\log(\frac{1}{\varepsilon}))$ queries are enough to test whether $\mathbf{M}$ is symmetric or not with probability $1 - \varepsilon$.*

Proof. We choose two random vectors $\mathbf{u}$ and $\mathbf{v}$, where, over a finite field, we choose from a uniform distribution and over fields $\mathbb{R}$ or $\mathbb{C}$ we choose the Gaussian distribution. We then compute $\mathbf{M}\mathbf{u}$ and $\mathbf{M}\mathbf{v}$. We declare $\mathbf{M}$ to be symmetric if and only if $\mathbf{u}^T \cdot \mathbf{M}\mathbf{v} = \mathbf{v}^T \cdot \mathbf{M}\mathbf{u}$. It is easy to check that if $\mathbf{M}$ is symmetric, the test will succeed. We then show if $\mathbf{M}$ is not symmetric, $\mathbf{u}^T\mathbf{M}\mathbf{v} \neq \mathbf{v}^T\mathbf{M}\mathbf{u}$ with constant probability, so we obtain success probability $1 - \varepsilon$ by repeating the test $O(\log(\frac{1}{\varepsilon}))$ times.

Let $\mathbf{A} = \mathbf{M} - \mathbf{M}^T$. When $\mathbf{M}$ is not symmetric, $\mathbf{A}$ is not 0. Thus, $\mathbf{u}^T\mathbf{M}\mathbf{v} = \mathbf{v}^T\mathbf{M}\mathbf{u}$ means $\mathbf{u}^T\mathbf{A}\mathbf{v} = 0$. We can treat this as a degree-2 polynomial in the entries of $\mathbf{v}^T$ and $\mathbf{u}$, i.e., this is $\sum_{i,j} u_i v_j \mathbf{A}_{i,j} = \sum_i u_i \sum_j v_j \mathbf{A}_{i,j}$. Thus, this is a non-zero polynomial and has at most constant probability of evaluating to 0 for any underlying field. To see this, for each i, let $t_i = \sum_j v_j \mathbf{A}_{i,j}$. Then there will be at least one t_i which is non-zero with probability at least $1/2$, for any underlying field. So now we get $\sum_i u_i t_i$. Fix all the u_i except u_i for a given t_i that is non-zero. Then, we obtain $S + u_i t_i$. Then, if u_i has at least two possible values, this is 0 in one case and non-zero in the other case. So we obtain a probability of at least $1/4$ of detection overall. □

3.4 Deciding if M is a Diagonal Matrix

Theorem 5. *Given an $n \times n$ matrix $\mathbf{M}$ over any finite field or over fields $\mathbb{R}$ or $\mathbb{C}$, $O(\log(\frac{1}{\varepsilon}))$ queries are sufficient to test whether $\mathbf{M}$ is a diagonal matrix with failure probability $\leq \varepsilon$.*

Proof. The first query is an all ones vector which retrieves the sum of each row. Then we take $\Omega(\log \frac{1}{\varepsilon})$ random queries where each entry is uniformly sampled from $\{0, 1\}$. We then show that every row containing non-zero entries off the diagonal can be detected with probability $1/2$ under such a random query. We illustrate it with the first row $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ of matrix $\mathbf{M}$. Without loss of generality, assume the off-diagonal element $a_2 \neq 0$. Let $\mathbf{v}$ be a random query. Thus, we have $Pr(\sum_i a_i v_i = (\sum_i a_i)v_1) = Pr(v_2 = a_2^{-1}(v_1 + \sum_{i=3}^n a_i(v_1 - v_i))) \leq 1/2$. This means every

random query can detect this row with probability at least 1/2. This implies bounded error $\leq \varepsilon$ after $\Omega(\log \frac{1}{\varepsilon})$ random queries. Furthermore, this algorithm works over any field. □

3.5 Deciding if M is a Unitary Matrix

Theorem 6. *Given an $n \times n$ complex matrix $\mathbf{M}$, 1 query is enough to test whether $\mathbf{M}$ is unitary or not, that is $\mathbf{M}^*\mathbf{M} = \mathbf{M}\mathbf{M}^* = I$.*

Proof. We choose a random Gaussian vector $\mathbf{v}$, and compute $\mathbf{M}\mathbf{v}$. We declare $\mathbf{M}$ to be unitary if and only if $|\mathbf{M}\mathbf{v}|_2 = |\mathbf{v}|_2$. It is easy to check that, if $\mathbf{M}$ is unitary, the test will succeed. We then show that if $\mathbf{M}$ is not unitary, $|\mathbf{M}\mathbf{v}|_2 \neq |\mathbf{v}|_2$ with probability 1. Let the singular value decomposition of $\mathbf{M}$ be $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$. We have $|\mathbf{M}\mathbf{v}|_2^2 = |\Sigma\mathbf{u}|_2^2$, where $\mathbf{u} = \mathbf{V}^T\mathbf{v}$ is a random Gaussian vector with $|\mathbf{u}|_2^2 = |\mathbf{v}|_2^2$. The diagonal values in Σ are not all 1 since $\mathbf{M}$ is not unitary. Consider $\sum_i \sigma_i^2 u_i^2$, where $\sigma_i = \Sigma_{i,i}$. We want this to equal $|\mathbf{v}|_2^2 = |\mathbf{u}|_2^2 = \sum_i u_i^2$, so this is $\sum_i u_i^2(\sigma_i^2 - 1) = 0$. This is a non-zero polynomial and has probability 0 of evaluating to 0 since the u_i^2 are drawn from a continuous distribution. □

3.6 Approximating the Maximum Eigenvalue

The upper bound is due to [31]. Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, we can ε-approximate the maximum eigenvalue of $\mathbf{M}$ by taking a random vector $\mathbf{v} \in \mathbb{R}^n$ and computing $\mathbf{M}\mathbf{v}, \mathbf{M}^2\mathbf{v}, \ldots, \mathbf{M}^r\mathbf{v}$ for $r = O(\varepsilon^{-0.5}\log n)$. This requires r adaptive oracle queries to $\mathbf{M}$. See [31] for details. See [35] for a matching lower bound for adaptive queries. A non-adaptive $\Omega(n)$ lower bound is given in [29].

4 STREAMING AND STATISTICS PROBLEMS

In this part, we discuss the following streaming and statistics problems: testing an all ones column/row and identical columns/rows; approximating row norms or finding heavy hitters; and computing the majority or parity of columns/rows.

4.1 Testing Existence of an All Ones Column/Row

Given a matrix $\mathbf{M} \in \{0,1\}^{m \times n}$, we want to test if $\mathbf{M}$ has a column (or row) with all 1 entries. It is trivial to test whether $\mathbf{M}$ has an all 1 column (or row) using n queries, e.g., $\mathbf{e}_1, \ldots, \mathbf{e}_n$. We consider this problem both over $\mathbb{F}[2]$ and $\mathbb{R}$. Note in the case over $\mathbb{R}$, if we allow an arbitrary query vector, we can set one query $\mathbf{v} = \{1, 2, 4, 8, \ldots, 2^n\}$, and then reconstruct $\mathbf{M}$ exactly. Thus, in order to avoid such trivial cases, we also restrict the entries in the query to be in $\{0, 1, 2, \ldots, n^C\}$.

For testing the existence of an all ones column, we reduce the problem to the communication complexity of Disjointness. Disjointness requires $\Omega(n)$ bits of communication to decide whether two sets with characteristic vectors $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$ are disjoint with constant probability, where the randomness is taken only over the coin tosses of the protocol (not over the inputs). Suppose the first $m-1$ rows in $\mathbf{M}$ each equal $\mathbf{x}^T$ while the last row equals $\mathbf{y}^T$. If we can decide whether $\mathbf{M}$ has an all ones column with q non-adaptive queries $\mathbf{v}_1, \ldots, \mathbf{v}_q$, then we obtain a protocol for Disjointness with communication q by letting Alice send a message $(\mathbf{x}^T\mathbf{v}_1, \ldots, \mathbf{x}^T\mathbf{v}_q)$. Thus, from the communication complexity lower bound of Disjointness, $q = \Omega(n)$ queries over $\mathbb{F}[2]$ are necessary to test if there is an all ones column in $\mathbf{M}$, which shows that the naïve algorithm is already optimal. For queries over $\mathbb{R}$, note that each entry $\mathbf{x}^T\mathbf{v}_j$ in the message is represented with $\log n$ bits, and as a result $q \geq \Omega(n/\log n)$. To summarize, we have:

Theorem 7. *Given a matrix $\mathbf{M} \in \{0,1\}^{m \times n}$, we want to test if $\mathbf{M}$ has a column with all 1 entries. Over field $\mathbb{F}[2]$, $q = \Omega(n)$ queries are necessary; while over field $\mathbb{R}$, $\Omega(n/\log n)$ queries are necessary if we restrict the entries in the query to be in $\{0, 1, 2, \ldots, n^C\}$ for some constant C.*

Testing the existence of an all ones row with queries over $\mathbb{R}$ is trivial deterministically by querying $\mathbf{v} = (1, 1, \ldots, 1)$. Next, we study the query complexity of testing an all 1s row deterministically with queries over $\mathbb{F}[2]$. With any $q \leq n-1$ queries $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_q]$, there is a non-zero vector $\mathbf{z} \neq \mathbf{0}$ such that $\mathbf{z}^T\mathbf{V} = \mathbf{0}$. Therefore, the query matrix $\mathbf{V}$ cannot distinguish whether a row is from $\mathbf{x}^T$ or $\mathbf{x}^T + \mathbf{z}^T$. However, $\mathbf{x}^T$ and $\mathbf{x}^T + \mathbf{z}^T$ cannot be both all 1 rows, and hence n queries are necessary. This result also shows that the query complexity of the same problem over different fields might be quite different. We note for randomized algorithms, $O(\log(1/\varepsilon))$ queries suffice over $\mathbb{F}[2]$. Suppose a row $\mathbf{a}$ is not all ones where $a_1 = 0$ without loss of generality. Choose a random query $\mathbf{v}$ where each entry is uniformly sampled from $\{0, 1\}$. Thus, we have $Pr(\sum_i a_i v_i = \sum_i v_i) = Pr(v_1 = \sum_{i=2}^{n}(a_i - 1)v_i) = 1/2$. This means, a random query can detect a not-all-ones row with probability $1/2$. So $O(\log(1/\varepsilon))$ random queries suffice to check the existence of an all ones row with failure probability $\leq \varepsilon$ over $\mathbb{F}[2]$. To summarize, we have:

Theorem 8. *Given a matrix $\mathbf{M} \in \{0,1\}^{m\times n}$, we want to test if $\mathbf{M}$ has a row with all 1 entries. Over field $\mathbb{R}$, 1 query is enough. Over field $\mathbb{F}[2]$, if we use deterministic algorithm, n queries are necessary to test; while $O(\log(1/\varepsilon))$ random queries suffice to check the existence of an all ones row with failure probability $\leq \varepsilon$.*

Evaluating the OR/AND function of columns/rows of a Boolean matrix can be reduced to testing existence of an all 1 or all 0 column/row, and hence the same bounds follow.

4.2 Identical Columns/Rows

Given an $m \times n$ matrix $\mathbf{M}$, we want to test whether $\mathbf{M}$ has two identical columns or rows. The trivial solution naively retrieves all information with n queries (column vectors). In this section, we consider the query complexity over $\mathbb{F}[2]$.

Theorem 9. *Given an $m\times n$ matrix $\mathbf{M}$, over field $\mathbb{F}[2]$, to test whether $\mathbf{M}$ has two identical columns, when $m \geq 4\log(n/\varepsilon)$, any randomized algorithm with successful probability at least $1-\varepsilon$ needs $\Omega(n/m)$ queries. On the other hand, to test whether $\mathbf{M}$ has two identical rows, $O(\log(m/\varepsilon))$ queries are enough with probability $1-\varepsilon$.*

Proof. Testing identical columns can be reduced to Disjointness. Suppose Alice and Bob have $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$. Let Alice expand her vector $\mathbf{x}$ to an $\frac{m}{2} \times n$ matrix $\mathbf{M}_1$ as follows: the first row is $(1, \mathbf{x}^T) = (1, x_1, \ldots, x_n)$; for $2 \leq i \leq \frac{m}{2}$ the ith row is $(1, z_1^{(i)}, \ldots, z_n^{(i)})$ where $z_j^{(i)} = 1$ if $x_j = 1$, and $z_j^{(i)}$ is uniformly random over $\{0,1\}$ if $x_j = 0$, for $1 \leq j \leq n$. Bob expands his vector $\mathbf{y}$ to $\mathbf{M}_2$ similarly. Putting $\mathbf{M}_1, \mathbf{M}_2$ together, we let $\mathbf{M} \stackrel{\text{def}}{=} \left[\begin{smallmatrix}\mathbf{M}_1\\ \mathbf{M}_2\end{smallmatrix}\right]$. Then, $\mathbf{M}$ is an $m \times (n+1)$ matrix with the first column being all 1s. For $j \geq 2$, the jth column is all 1s if and only if $x_j = y_j = 1$, in which case $\mathbf{M}$ has two identical rows of all 1 entries. For columns where x_j, y_j are not both equal to 1, without loss of generality, we may assume the jth and j'th columns satisfy $x_j = x_{j'} = 0$ and $y_j = y_{j'}$. Then, two columns are identical only if $(z_j^{(2)}, \ldots, z_j^{(\frac{m}{2})}) = (z_{j'}^{(2)}, \ldots, z_{j'}^{(\frac{m}{2})})$, which happens with probability $\leq 1/2^{\frac{m}{2}-1}$. Therefore, the overall probability of two not-all-ones columns in $\mathbf{M}$ being identical is bounded by $n^2/2^{m/2}$. Thus, the error probability is less than ε if $m \geq 4\log(n/\varepsilon)$.

That is, except for a small error ε, two identical columns in $\mathbf{M}$ are both all ones columns, which turns out to be equivalent to the case that two vectors $\mathbf{x}, \mathbf{y}$ held by Alice and Bob are not disjoint. Then, because Disjointness requires $\Omega(n)$ bits of communication, and after one oracle queries, Alice or Bob can communicate at most m bits, at least $\Omega(n/m)$ oracle queries to $\mathbf{M}$ are necessary.

On the other hand, to test identical rows with error ε, if suffices to make $q = O(\log(m/\varepsilon))$ random queries with each entry uniform random over $\{0,1\}$. Since for every pair of distinct rows, a random query distinguishes them with probability $\frac{1}{2}$, with $\lceil \log(m^2/\varepsilon) \rceil$ queries each pair of distinct rows is

miscounted as identical with probability $\leq \varepsilon/m^2$. By a union bound, the overall false-positive error is bounded by $\frac{\varepsilon}{m^2} \cdot \binom{m}{2} < \varepsilon$, while there is no false-negative error since for all queries, identical rows always lead to identical outputs. □

4.3 Approximating Row Norms and Finding Heavy Hitters

Theorem 10. *Given a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, $O(\varepsilon^{-2}\log m)$ queries are enough to approximate the row norms up to a $(1\pm\varepsilon)$-factor, and $O(\varepsilon^{-2}\log m)$ queries are enough to find the heavy hitters with probability $1-\varepsilon$.*

Proof. To approximate the norms of each row in a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, we recall the Johnson-Lindenstrauss lemma which guarantees that norms are roughly preserved when embedded to a lower dimensional space. Thus, with $q = O(\varepsilon^{-2}\log m)$ and an $n \times q$ random query matrix $\mathbf{V}$, the output $\mathbf{M}\cdot\mathbf{V}$ preserves the row norms of $\mathbf{M}$ up to a $(1\pm\varepsilon)$-factor.

The above algorithm also gives a natural upper bound for finding heavy hitters in the matrix $\mathbf{M}$, which requires finding all rows $\mathbf{M}_i$ with norm $|\mathbf{M}_i|_2^2 \geq \frac{1}{10}|\mathbf{M}|_F^2$ and not outputting any row $\mathbf{M}_i$ with $|\mathbf{M}_i|_2^2 \leq \frac{1}{20}|\mathbf{M}|_F^2$ (rows with norm in between the two quantities can be classified arbitrarily). Again we use the Johnson-Lindenstrauss lemma to approximate all row norms and decide which row is a heavy hitter. □

4.4 Majority

Theorem 11. *Given a matrix $\mathbf{M} \in \{0,1\}^{m\times n}$ with queries over $\mathbb{R}$, 1 query is enough to compute the majority of rows in $\mathbf{M}$; while $\Omega(n/\log n)$ queries are necessary to compute the majority of columns in $\mathbf{M}$.*

Proof. The majority of each row in $\mathbf{M}$ is trivial with an all 1 query and addition over $\mathbb{R}$.

For the majority of columns in $\mathbf{M}$, we use a similar matrix $\mathbf{M}$ as that reduced from Disjointness in Section 4.2 to obtain a lower bound. More specifically, we consider $\mathbf{x}, \mathbf{y}$ whose intersection is at most 1. Let $\mathbf{M}$ be obtained from $\mathbf{x}, \mathbf{y}$ such that the first $m/2$ rows are identical to $\mathbf{x}$ and the remaining rows are identical to y. Thus, if $\mathbf{M}$ has a column with majority 1, then the column must be all 1s and we can conclude that $\mathbf{x}$ and $\mathbf{y}$ are not disjoint. As a result, $\Omega(n/\log n)$ queries are necessary to compute the majority of columns in $\mathbf{M}$. □

4.5 Parity

For parity, we consider a matrix $\mathbf{M} \in \{0,1\}^{m\times n}$ with only queries over $\mathbb{F}[2]$. Computing the parity of rows in $\mathbf{M}$ is trivial by using a vector $(1,1,\ldots,1)$. However, to compute the parity of all columns of $\mathbf{M}$, we claim at least n queries are necessary.

Theorem 12. *Given a matrix $\mathbf{M} \in \{0,1\}^{m\times n}$ with only queries over $\mathbb{F}[2]$, 1 query is enough to compute the parity of rows in $\mathbf{M}$; while $\Omega(n)$ queries are necessary to compute the parity of columns in $\mathbf{M}$.*

Proof. Let $\mathbf{V}$ be any $n\times q$ query matrix. Note that the parity of columns of $\mathbf{M}$ remains the same if we sum up all the rows, i.e., $\mathbf{M}' \stackrel{\text{def}}{=} \mathbf{P}\cdot\mathbf{M}$ has the same parity on each column as $\mathbf{M}$, where $\mathbf{P}$ is defined to be

$$\mathbf{P} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 1 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}_{m\times m}$$

Thus, $\mathbf{M'V} = \mathbf{PMV}$ is a $1 \times q$ row vector followed by $m-1$ zero rows, since $\mathbf{M'}$, as well as $\mathbf{P}$, is non-zero only in the first row. Then, we must have $q = \Omega(n)$ to obtain the output of n parity instances from $\mathbf{M'V}$. Indeed, if we were to place the uniform distribution on $\mathbf{M}$, then its columns define n uniform parity bits, and for any fixed $\mathbf{V}$, we only obtain q bits of information, which is a contradiction to Yao's minimax principle (since there must be a fixed V which succeeds with at least 2/3 probability on this distribution). This is a typical example illustrating the difference between left- and right-queries. □

5 GRAPH PROBLEMS

In this part, we provide our results related to graph problems: testing graph connectivity in Section 5.1 and triangle detection in Section 5.2.

5.1 Connectivity

Theorem 13. *Given the bipartite adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$ of a graph, we need $\Omega(n/\log n)$ queries to decide whether the graph is connected with constant probability.*

Proof. Consider two row vectors $\mathbf{u}, \mathbf{v} \in \{0,1\}^{n-1}$ and construct matrix $\mathbf{A}$ as follows: The first $n/2$ rows of $\mathbf{A}$ equal $\mathbf{u}$ and the rest are equal to $\mathbf{v}$. Also, add an all 1s column to $\mathbf{A}$. Now, matrix $\mathbf{A}$ can be treated as a bipartite adjacency matrix of a graph G with n vertices in each part, where $\mathbf{A}_{i,j} = 1$ iff there is an edge from the ith left vertex to the jth right vertex. Since all left vertices connect with the nth right vertex, the graph G is disconnected if and only if there exists some right vertex which does not connect with any left vertices, that is, the corresponding column of matrix $\mathbf{A}$ is an all 0s column. In other words, G is disconnected if and only if the two vectors $\mathbf{u}$ and $\mathbf{v}$ are 0 on the same position.

Thus, any algorithm that uses $q(n)$ non-adaptive queries on the right of $\mathbf{A}$ to decide the connectivity of G immediately implies a protocol for set disjointness, provided we replace 1s with 0s in the input characteristic vectors to the set disjointness problem. So the communication is at most $q(n)\log n$, thus $q(n) = \Omega(n/\log n)$. □

Theorem 14. *Given the signed edge-vertex incidence matrix $\mathbf{M} \in \{0,\pm 1\}^{n\times\binom{n}{2}}$ of a graph G with n vertices, the connectivity of G can be decided with* polylog (n) *non-adaptive queries.*

This follows from the main theorem of [23] (also proved in the work [1]). By the following theorem, every cut of G is multiplicatively approximated and hence G is connected iff H is connected, since a graph is disconnected iff it has a zero cut.

Theorem 15 ([23]). *There is a distribution on $\binom{n}{2} \times$ polylog (n) matrices $\mathbf{S}$ such that from $\mathbf{MS}$, one can construct a (1 ± 0.1)-sparsifier H of the graph G with constant probability. Here, $\mathbf{x}^T\mathbf{L}_G x = (1 \pm 0.1)\mathbf{x}^T\mathbf{L}_H x$ for all x, with constant probability, where $\mathbf{L}_G$ and $\mathbf{L}_H$ are the corresponding graph Laplacians.*

By the above, every cut of G is multiplicatively approximated and hence G is connected iff H is connected, since a graph is disconnected iff it has a zero cut.

5.2 Triangle Detection

Theorem 16. *If an $n\times n$ matrix $\mathbf{A}$ is the adjacency matrix of a graph G, then determining whether G contains a triangle or not requires $\Omega(n/\log n)$ queries, even for randomized algorithms succeeding with constant probability.*

Proof. To obtain a lower bound on $q(n)$, we use a 2-player communication lower bound of counting the number of triangles in a graph G, where the edges are distributed across the two

players, Alice and Bob. Namely, it is known [9, 17, 18] that if Alice has a subset of the edges of G, and Bob has the remaining (disjoint) subset of edges of G, then the multiround randomized communication complexity of deciding if there is a triangle in G is $\Omega(n^2)$. Alice can view her subset of edges as an adjacency matrix $\mathbf{A}'$, and Bob can view his subset of edges as an adjacency matrix $\mathbf{A}''$, so that $\mathbf{A} = \mathbf{A}' + \mathbf{A}''$. To execute the query algorithm on $\mathbf{A}$, Alice sends $\mathbf{A}'\mathbf{x}_1$ to Bob, who computes $\mathbf{A}''\mathbf{x}_1$ followed by $\mathbf{A}'\mathbf{x}_1 + \mathbf{A}''\mathbf{x}_1 = \mathbf{A}\mathbf{x}_1$, and sends the result back to Alice. Alice then possibly adaptively chooses $\mathbf{x}_2$, which is also known to Bob who knows $\mathbf{x}_1$ and $\mathbf{A}\mathbf{x}_1$, and sends Bob $\mathbf{A}'\mathbf{x}_2$, from which Bob can compute $\mathbf{A}''\mathbf{x}_2$ and $\mathbf{A}\mathbf{x}_2 = \mathbf{A}'\mathbf{x}_2 + \mathbf{A}''\mathbf{x}_2$. This process repeats until the entire $q(n)$ queries have been executed, at which point Bob, by the success guarantee of the algorithm, can decide if G contains a triangle with say, probability at least 2/3. Because of the bounds on the bit complexity of the queries while the total communication is $O(q(n)n \log n)$, which must be $\Omega(n^2)$, and consequently $q(n) = \Omega(n/\log n)$, as desired. □

6 CONCLUSIONS

We initiated the study of querying a matrix through matrix-vector products. We illustrated that, for some quantities, if one can only query matrix-vector products on one side, the problem becomes harder. We also illustrated the importance of the underlying field defining the matrix-vector products, as well as the representation of the graph for graph problems. Given connections to sketching algorithms, streaming, and compressed sensing, we believe this area deserves its own study. Some interesting open questions are for computing matrix norms, such as Schatten-p norms, for which tight bounds in streaming and communication complexity models remain elusive; for recent work on this, see [13], [28], and [30]. Given the success of our model in proving lower bounds for approximate rank, which we also do not have streaming or communication lower bounds for, perhaps tight bounds in our query model are possible for matrix norms. Such bounds may give insight for other models.

ACKNOWLEDGMENTS

We would like to thank Yi Li, Yan Shuo Tan, Max Simchowitz, and Roman Vershynin for helpful comments. We also thank the reviewers of ICALP 2019 for their detailed feedback, which has greatly helped the readability of the article.

REFERENCES

[1] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. 2012. Analyzing graph structure via linear measurements. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA'12)* (Kyoto, Japan, January 17–19, 2012). 459–467.

[2] Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. 2016. New characterizations in turnstile streams with applications. In *Proceedings of the 31st Conference on Computational Complexity (CCC'16)* (May 29 to June 1, 2016, Tokyo, Japan). 20:1–20:22.

[3] Noga Alon, Yossi Matias, and Mario Szegedy. 1996. The space complexity of approximating the frequency moments. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC'96)* (Philadelphia, PA, May 22–24, 1996). 20–29.

[4] Sepehr Assadi, Sanjeev Khanna, and Yang Li. 2017. On estimating maximum matching size in graph streams. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)* (Barcelona, Spain, Hotel Porta Fira, January 16–19). 1723–1742.

[5] Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. 2016. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'16)* (Arlington, VA, January 10–12, 2016). 1345–1364.

[6] Haim Avron and Sivan Toledo. 2011. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM* 58, 2 (2011), 8:1–8:34.

[7] Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. 2010. Lower bounds for sparse recovery. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)* (Austin, TX, January 17–19, 2010). 1190–1197.

[8] Maria-Florina Balcan, Yi Li, David P. Woodruff, and Hongyang Zhang. 2019. Testing matrix rank, optimally. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)* (San Diego, CA, January 6–9, 2019). 727–746.

[9] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. 2002. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'02)* (January 6–8, 2002, San Francisco, CA), 623–632.

[10] Anders Björner, László Lovász, and Andrew C. C. Yao. 1992. Linear decision trees: Volume estimates and topological bounds. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC'92)*. ACM, 170–177.

[11] Eric Blais, Joshua Brody, and Kevin Matulef. 2012. Property testing lower bounds via communication complexity. *Computational Complexity* 21, 2 (2012), 311–358.

[12] Mark Braverman, Elad Hazan, Max Simchowitz, and Blake E. Woodworth. The gradient complexity of linear regression. In *Proceedings of the Conference on Learning Theory (COLT'20),* (9–12 July 2020, Virtual Event [Graz, Austria]).

[13] Vladimir Braverman, Stephen R. Chestnut, Robert Krauthgamer, Yi Li, David P. Woodruff, and Lin Yang. 2018. Matrix norms in data streams: Faster, multi-pass and row-order. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018*. 648–657.

[14] Emmanuel J Candes, Justin K Romberg, and Terence Tao. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59, 8 (2006), 1207–1223.

[15] Gautam Dasarathy, Parikshit Shah, Badri Narayan Bhaskar, and Robert D. Nowak. 2015. Sketching sparse matrices, covariances, and graphs via tensor products. *IEEE Trans. Inf. Theory* 61, 3 (2015), 1373–1388.

[16] Talya Eden, Amit Levi, Dana Ron, and C. Seshadhri. 2017. Approximately counting triangles in sublinear time. *SIAM J. Comput.* 46, 5 (2017), 1603–1646.

[17] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. 2008. Graph distances in the data-stream model. *SIAM J. Comput.* 38, 5 (2008), 1709–1727.

[18] Magnús M. Halldórsson, Xiaoming Sun, Mario Szegedy, and Chengu Wang. 2012. Streaming and communication complexity of clique approximation. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming, Part I (ICALP'12)* (Warwick, UK, July 9–13, 2012), 449–460.

[19] Piotr Indyk, Eric Price, and David P. Woodruff. 2011. On the power of adaptivity in sparse recovery. In *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS'11)* (Palm Springs, CA, October 22–25, 2011). 285–294.

[20] Akshay Kamath and Eric Price. 2019. Adaptive sparse recovery with limited adaptivity. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)* (San Diego, CA, January 6–9, 2019). 2729–2744.

[21] Daniel M. Kane, Shachar Lovett, and Shay Moran. 2018. Near-optimal linear decision trees for k-SUM and related problems. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)* (Los Angeles, CA, June 25–29, 2018). 554–563.

[22] Sampath Kannan, Elchanan Mossel, Swagato Sanyal, and Grigory Yaroslavtsev. 2018. Linear Sketching over F_2. In *Proceedings of the 33rd Computational Complexity Conference (CCC'18),* (San Diego, CA, June 22–24, 2018), 8:1–8:37.

[23] Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. 2017. Single pass spectral sparsification in dynamic streams. *SIAM J. Comput.* 46, 1 (2017), 456–477.

[24] John T. Kent and R. J. Muirhead. 1984. Aspects of multivariate statistical theory. *The Statistician* 33, 2 (1984), 251.

[25] Christian Konrad. 2015. Maximum matching in turnstile streams. In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA'15),* (Patras, Greece, September 14–16, 2015). 840–852.

[26] Yi Li, Huy L. Nguyen, and David P. Woodruff. 2014. On sketching matrix norms and the top singular vector. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14)* (Portland, OR, January 5–7, 2014). 1562–1581.

[27] Yi Li, Huy L. Nguyen, and David P. Woodruff. 2014. Turnstile streaming algorithms might as well be linear sketches. In *Proceedings of the Symposium on Theory of Computing (STOC'14),* (New York, NY, May 31 - June 3, 2014). 174–183.

[28] Yi Li and David P. Woodruff. 2016. On approximating functions of the singular values in a stream. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC'16)* (Cambridge, MA, June 18–21, 2016). 726–739.

[29] Yi Li and David P Woodruff. 2016. Tight bounds for sketching the operator norm, Schatten norms, and subspace embeddings. *RANDOM/APPROX 2016* 60, 39 (2016), 1–11.

[30] Yi Li and David P. Woodruff. 2017. Embeddings of Schatten norms with applications to data streams. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP'17)* (July 10–14, 2017, Warsaw, Poland). 60:1–60:14.

[31] Cameron Musco and Christopher Musco. 2015. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15) (December 7-12, 2015, Montreal, Canada)*. 1396–1404.

[32] Vasileios Nakos, Xiaofei Shi, David P. Woodruff, and Hongyang Zhang. 2018. Improved algorithms for adaptive compressed sensing. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP'18)* (July 9–13, 2018, Prague, Czech Republic). 90:1–90:14.

[33] Eric Price and David P. Woodruff. 2013. Lower bounds for adaptive sparse recovery. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'13)* (New Orleans, LA, January 6–8, 2013). 652–663.

[34] Jianhong Shen. 2001. On the singular values of Gaussian random matrices. *Linear Algebra Appl.* 326 (2001), 1–14.

[35] Max Simchowitz, Ahmed El Alaoui, and Benjamin Recht. 2018. Tight query complexity lower bounds for PCA via finite sample deformed Wigner law. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, (STOC'18)* (Los Angeles, CA, June 25–29, 2018). 1249–1259.

[36] Xiaoming Sun, David P. Woodruff, Guang Yang, and Jialin Zhang. 2019. Querying a matrix through matrix-vector products. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP'19)* (July 9–12, 2019, Patras, Greece), Vol. 132. 94:1–94:16.

[37] Roman Vershynin. 2010. Introduction to the non-asymptotic analysis of random matrices. *Compressed Sensing: Theory and Applications* (11 2010).

[38] Martin Wainwright. High-dimensional statistics: A non-asymptotic viewpoint. https://www.stat.berkeley.edu/~mjwain/stat210b/Chap2_TailBounds_Jan22_2015.pdf.

[39] Hermann Weyl. 1912. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Math. Ann.* 71, 4 (1912), 441–479.

[40] David P. Woodruff. 2014. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science* 10, 1–2 (2014), 1–157.

[41] Andrew Chi-Chin Yao. 1977. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*. IEEE, 222–227.

[42] Qiaochu Yuan. 2017. Singular value decomposition. (2017). https://qchu.wordpress.com/2017/03/13/singular-value-decomposition/.

Received February 2020; revised January 2021; accepted June 2021