On Identity Testing of Tensors, Low-rank Recovery and Compressed Sensing

[Extended Abstract] [*]

Michael A. Forbes[†]
MIT CSAIL
Cambridge, USA
miforbes@mit.edu

Amir Shpilka[‡]
Faculty of Computer Science
Technion — Israel Institute of Technology
Haifa, Israel
shpilka@cs.technion.ac.il

ABSTRACT

We study the problem of obtaining efficient, deterministic, *black-box polynomial identity testing algorithms* for depth-3 set-multilinear circuits (over arbitrary fields). This class of circuits has an efficient, deterministic, white-box polynomial identity testing algorithm (due to Raz and Shpilka [36]), but has no known such black-box algorithm. We recast this problem as a question of finding a low-dimensional subspace $\mathcal{H}$, spanned by rank 1 tensors, such that any non-zero tensor in the dual space $\ker(\mathcal{H})$ has high rank. We obtain explicit constructions of essentially optimal-size hitting sets for tensors of degree 2 (matrices), and obtain the first quasi-polynomial sized hitting sets for arbitrary tensors.

We also show connections to the task of performing *low-rank recovery* of matrices, which is studied in the field of compressed sensing. Low-rank recovery asks (say, over $\mathbb{R}$) to recover a matrix M from few measurements, under the promise that M is rank $\leq r$. In this work, we restrict our attention to recovering matrices that are exactly rank $\leq r$ using deterministic, non-adaptive, linear measurements, that are free from noise. Over $\mathbb{R}$, we provide a set (of size $4nr$) of such measurements, from which M can be recovered in $\mathcal{O}(rn^2 + r^3n)$ field operations, and the number of measurements is essentially optimal. Further, the measurements can be taken to be all rank-1 matrices, or all sparse matrices. To the best of our knowledge no explicit constructions with those properties were known prior to this work.

We also give a more formal connection between low-rank recovery and the task of *sparse (vector) recovery*: any sparse-recovery algorithm that exactly recovers vectors of length n and sparsity $2r$, using m non-adaptive measurements, yields a low-rank recovery scheme for exactly recovering $n \times n$ matrices of rank $\leq r$, making $2nm$ non-adaptive measurements. Furthermore, if the sparse-recovery algorithm runs in time τ, then the low-rank recovery algorithm runs in time $\mathcal{O}(rn^2 + n\tau)$. We obtain this reduction using linear-algebraic techniques, and not using convex optimization, which is more commonly seen in compressed sensing algorithms.

Finally, we also make a connection to *rank-metric codes*, as studied in coding theory. These are codes with codewords consisting of matrices (or tensors) where the distance of matrices M and N is $\text{rank}(M - N)$, as opposed to the usual hamming metric. We obtain essentially optimal-rate codes over matrices, and provide an efficient decoding algorithm. We obtain codes over tensors as well, with poorer rate, but still with efficient decoding.

Categories and Subject Descriptors

F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems; E.4 [**Data**]: Coding and Information Theory

General Terms

Theory

Keywords

Polynomial Identity Testing, Low-rank Recovery, Sparse Recovery, Tensor Rank, Derandomization

1. INTRODUCTION

We start with a motivating example. Let $\mathbf{x}$ and $\mathbf{y}$ be vectors of n variables each. Let M be an $n \times n$ matrix (over some field, say $\mathbb{R}$), and define the quadratic form

$$f_M(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \mathbf{x}^\dagger M \mathbf{y} \ .$$

Suppose now that we are given an oracle to f_M, that can evaluate f_M on inputs $(\mathbf{x}, \mathbf{y})$ that we supply. The type of question we consider is: how many (deterministically chosen) evaluations of f_M must we make in order to determine whether M is non-zero?

It is not hard to show that n^2 evaluations to f_M are necessary and sufficient to determine whether M is non-zero. The

9[*]A full version of this paper is available at http://arxiv.org/abs/1111.0663

[†]Supported by NSF grant 6919791, MIT CSAIL and a Siebel Scholarship.

[‡]The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575.

STOC'12, May 19–22, 2012, New York, New York, USA.
Copyright 2012 ACM 978-1-4503-1245-5/12/05 ...$10.00.

question becomes more interesting when we are promised that $\mathrm{rank}(M) \leq r$. That is, given that $\mathrm{rank}(M) \leq r$, can we (deterministically) determine whether $M = 0$ using $\ll n^2$ evaluations of f_M? It is not hard to show that there (non-explicitly) *exist* $\approx 2nr$ evaluations to determine whether $M = 0$, and one of the new results in this paper is to give an *explicit* construction of $2nr$ such evaluations (over $\mathbb{R}$).

We also consider various generalizations of this problem. The first generalization is to move from matrices (which are in a sense 2 dimensional) to the more general notion of *tensors* (which are in a sense d-dimensional). That is, a tensor is a map $T : [n]^d \to \mathbb{F}$ and like a matrix we can define a polynomial

$$f_T(x_{1,1}, \ldots, x_{1,n}, \ldots, x_{d,1}, \ldots, x_{d,n}) \overset{\mathrm{def}}{=} \sum_{i_1,\ldots,i_d \in [n]} T(i_1, \ldots, i_d) \prod_{j=1}^{d} x_{j,i_j} \ .$$

As with matrices, tensors have a notion of rank (defined later), and we can ask: given that $\mathrm{rank}(T) \leq r$ how many (deterministically chosen) evaluations of f_T are needed to determine whether $T = 0$. As $T = 0$ iff $f_T = 0$, we see that this problem is an instance of *polynomial identity testing*, which asks: given oracle access to a polynomial f that is somehow "simple", how many (deterministically chosen) queries to f are needed to determine whether $f = 0$?

The above questions ask whether a certain matrix or tensor is zero. However, we can also ask for more, and seek to reconstruct this matrix/tensor fully. That is, how many (deterministically chosen) evaluations to f_M are needed to determine M? This question can be seen to be related to compressed sensing and sparse recovery, where the goal is to reconstruct a "simple" object from "few" measurements. In this case, "simple" refers to the matrix being low-rank, as opposed to a vector being sparse. As above, it is not hard to show that there *exist* $\approx 4nr$ evaluations that determine M, and this paper gives an *explicit* construction of $4nr$ such evaluations, as well as an efficient algorithm to reconstruct M from these evaluations.

We will now place this work in a broader context by providing background on polynomial identity testing, compressed sensing and low-rank recovery and the theory of rank-metric codes.

1.1 Polynomial Identity Testing

Polynomial identity testing (PIT) is the problem of deciding whether a polynomial (specified by an arithmetic circuit) computes the identically zero polynomial. The obvious deterministic algorithm that completely expands the polynomial unfortunately takes exponential time. This is in contrast to the fact that there are several (quite simple) randomized algorithms that solve this problem quite efficiently. Further, some of these randomized algorithms treat the polynomial as a *black-box*, so that they only use the arithmetic circuit to evaluate the polynomial on chosen points, as opposed to a *white-box* algorithm which can examine the internal structure of the circuit. Even in the white-box model, no efficient deterministic algorithms are known for general circuits.

Understanding the deterministic complexity of PIT has come to be an important problem in theoretical computer science. Starting with the work of Kabanets and Impagliazzo [27], it has been shown that the existence of efficient deterministic (white-box) algorithms for PIT has a tight connection with the existence of explicit functions with large circuit complexity. As proving lower bounds on circuit complexity is one of the major goals of theoretical computer science, this has led to much research into PIT.

Stronger connections are known when the deterministic algorithms are black-box. For, any such algorithm corresponds to a *hitting set*, which is a set of evaluation points such that any small arithmetic circuit computing a non-zero polynomial must evaluate to non-zero on at least one point in the set. Heintz and Schnorr [25], as well as Agrawal [3], showed that any deterministic black-box PIT algorithm very easily yields explicit polynomials that have large arithmetic circuit complexity. Moreover, Agrawal and Vinay [4] showed that a deterministic construction of a polynomial size hitting set for arithmetic circuits of depth-4 gives rise to a quasi-polynomial sized hitting set for general arithmetic circuits. Thus, the black-box deterministic complexity of PIT becomes interesting even for constant-depth circuits. However, currently no polynomial size hitting sets are known for general depth-3 circuits. Much of recent work on black-box deterministic PIT has identified certain subclasses of circuits for which small hitting sets can be constructed, and this work fits into that paradigm. See [41] for a survey of recent results on PIT.

One subclass of depth-3 circuits is the model of *set multilinear* depth-3 circuits, first introduced by Nisan and Wigderson [35]. Raz and Shpilka [36] gave a polynomial-time white-box PIT algorithm for non-commutative arithmetic formulas, which contains set-multilinear depth-3 circuits as a subclass. However, no polynomial-time black-box deterministic PIT algorithm is known for set-multilinear depth-3 circuits. The best known black-box PIT results for the class of set-multilinear circuits, with top fan-in $\leq r$ and degree d, are hitting sets of size $\min(n^d, \mathsf{poly}((nd)^r))$, where the first part of bound comes from a simple brute-force argument, and the second part of the bound ignores that we have set-multilinear polynomials, and simply uses the best known hitting sets for so-called $\Sigma\Pi\Sigma(k)$ circuits as established by Saxena and Seshadhri [40]. For non-constant d and r, these bounds are super-polynomial. Improving the size of these hitting sets is the primary motivation for this work.

To connect PIT for set-multilinear depth-3 circuits with the above questions on matrices and tensors, we now note that any such circuit of top fan-in $\leq r$, degree d, on dn variables (and thus size $\leq dnr$), computes a polynomial f_T, where T is an $[n]^d$ tensor of rank $\leq r$. Conversely, any such f_T can be computed by such a circuit. Thus, constructing better hitting sets for this class of circuits is exactly the question of finding smaller sets of (deterministically chosen) evaluations to f_T to determine whether $T = 0$.

1.2 Low-Rank Recovery and Compressed Sensing

Low-rank Recovery (LRR) asks (for matrices) to recover an $n \times n$ matrix M from few *measurements* of M. Here, a measurement is some inner product $\langle M, H \rangle$, where H is an $n \times n$ matrix and the inner product $\langle \cdot, \cdot \rangle$ is the natural inner product on n^2 long vectors. This can be seen as the natural generalization of the *sparse recovery* problem, which asks to recover sparse vectors from few linear measurements. For, over matrices, our notion of sparsity is simply that of being low-rank.

Sparse recovery and compressed sensing are active areas of research, see for example [1]. Much of this area focuses on constructing distributions of measurements such that the unknown sparse vector can be recovered efficiently, with high probability. Also, it is often assumed that the sequence of measurements will not depend on any of the measurement results, and this is known as *non-adaptive sparse recovery*. We note that Indyk, Price and Woodruff [26] showed that *adaptive sparse recovery* can outperform non-adaptive measurements in certain regimes, but, in general, adaptivity cannot give much improvement [5]. Much of the existing work also focuses on efficiency concerns, and various algorithms coming from convex programming have been used. As such, these algorithms tend to be stable under noise, and can recover approximations to the sparse vector (and can even do so only if the original vector was approximately sparse). One of the initial achievements in this field is an efficient algorithm for recovery of a k-sparse[1] approximation of n-entry vector in $\mathcal{O}(k\log(n/k))$ measurements [9].

Analogous questions for low-rank recovery have also been explored (for example, see [2] and references there in). Initial work (such as [12, 10]) asked the question of low-rank *matrix completion*, where entries of a low-rank matrix M are revealed individually (as opposed measuring linear combinations of matrix entries). It was shown in these works that for an $n \times n$ rank $\leq r$ matrix, $\mathcal{O}(nr\mathsf{polylog}n)$ noisy samples suffice for *nuclear-norm minimization* to complete the matrix efficiently. Further works (such as [16] and [17] for higher dimensional tensors) prove that a randomly chosen set of measurements (with appropriate parameters) gives enough information for low-rank recovery, other works (such as [11, 37]) giving explicit conditions on the measurements that guarantee that the nuclear norm minimization algorithm works, and finally other works seek alternative algorithms for certain ensembles of measurements (such as[2] [29]). As in the sparse recovery case, most of these work seek stable algorithms that can deal with noisy measurements as well as matrices that are only approximately low-rank. Finally, we note that some applications (such as quantum state tomography) have additional requirements for their measurements (for example, they should be easy to prepare as quantum states) and some work has gone into this as well [23, 22].

We now make a crucial observation which shows that black-box PIT for the quadratic form f_M is actually very closely related to low-rank recovery of M. Note that

$$f_M(\mathbf{x},\mathbf{y}) = \mathbf{x}^\dagger M\mathbf{y} = \langle M, \mathbf{x}^\dagger\mathbf{y}\rangle \ .$$

That is, an evaluation of f_M corresponds to a measurement of M, and in particular this measurement is realized as a rank-1 matrix. Thus, we see that any low-rank-recovery algorithm that only uses rank-1 measurement can also determine if M is non-zero, and thus also performs PIT for quadratic forms. Conversely, suppose we have a black-box PIT algorithm for rank $\leq 2r$ quadratic forms. Note then that for any M, N with rank $\leq r$, $M - N$ has rank $\leq 2r$. Thus, if $M \neq N$ then f_{M-N} will evaluate to non-zero on some point in the hitting set. As $f_{M-N} = f_M - f_N$, it follows that a hitting set for rank $\leq 2r$ matrices will distinguish M and N. In particular, this shows that information-theoretically any hitting set for rank $\leq 2r$ matrices is also an LRR set. Thus, in addition to constructing hitting sets for the quadratic forms f_M, this paper will also use those hitting sets as LRR sets and also give efficient LRR algorithms for these constructions.

1.3 Rank-Metric Codes

Most existing work on LRR has focused on random measurements, whereas the interesting aspect of PIT is to develop deterministic evaluations of polynomials. As the main motivation for this paper is to develop new PIT algorithms, we will seek deterministic LRR schemes. Further, we will want results that are field independent, and so this work will focus on noiseless measurements (and matrices that are exactly of rank $\leq r$). In such a setting, LRR constructions are very related to *rank-metric codes*. These codes (related to *array codes*), are error-correcting codes where the messages are matrices (or tensors) and the normal notion of distance (the Hamming metric) is replaced by the rank metric (that is, the distance of matrices M and N is $\mathrm{rank}(M - N)$). Over matrices, these codes were originally introduced independently by Gabidulin, Delsarte and Roth [20, 19, 18, 14, 38]. They showed, using ideas from BCH codes, how to get optimal (that is, meeting an analogue of the Singleton bound) rank-metric codes over matrices, as well as how to decode these codes efficiently. A later result by Meshulam [34] constructed rank-metric codes where every codeword is a Hankel matrix. Roth [38] also showed how to construct rank-metric codes from *any* hamming-metric code, but did not provide a decoding algorithm. Later, Roth [39] considered rank-metric codes over tensors and gave decoding algorithms for a constant number of errors. Roth also discussed analogues to the Gilbert-Varshamov and Singleton bounds in this regime. This alternate metric is motivated by *crisscross errors* in data storage scenarios, where corruption can occur in bursts along a row or column of a matrix (and are thus rank-1 errors).

We now explain how rank-metric codes are related to LRR. Suppose we have a set of matrices $\mathcal{H}$ which form a set of (non-adaptive, deterministically chosen) LRR measurements that can recover rank $\leq r$ matrices. Define the code $\mathcal{C}$ as the set of matrices orthogonal to each matrix in $\mathcal{H}$. Thus, $\mathcal{C}$ is a linear code. Further, given some $M \in \mathcal{C}$ and E such that $\mathrm{rank}(E) \leq r$, it follows that $\mathcal{H}(M + E) = \mathcal{H}E$ (where we abuse notation and treat M and E as n^2-long vectors, and $\mathcal{H}$ as an $|\mathcal{H}| \times n^2$ matrix). That $\mathcal{H}$ is an LRR set means that E can be recovered from the measurements $\mathcal{H}E$. Thus the code $\mathcal{C}$ can correct r errors (and has minimum distance $\geq 2r + 1$, by a standard coding theory argument). Similarly, given a rank-metric code $\mathcal{C}$ that can correct up to rank $\leq r$ errors, the parity checks of this code define an LRR scheme. Thus, a small LRR set is equivalent to a rank-metric code with good rate.

The previous subsection showed the tight connection between LRR and PIT. Via the above paragraph, we see that hitting sets for quadratic forms are equivalent to rank-metric codes, when the parity check constraints are restricted to be rank 1 matrices.

9[1] A vector is k-sparse if it has at most k non-zero entries.

9[2] Interestingly, [29] use what they call *subspace expanders* a notion that was studied before in a different context in theoretical computer science and mathematics under the name of *dimension expanders* [32, 15].

1.4 Reconstruction of Arithmetic Circuits

Even more general than the PIT and LRR problems, we can consider the problem of reconstruction of general arithmetic circuits only given oracle access to the evaluation of that circuit. This is the arithmetic analog of the problem of learning a function using membership queries. For more background on reconstruction of arithmetic circuits we refer the reader to [41]. Just as with the PIT and LRR connection, PIT for a specific circuit class gives information-theoretic reconstruction for that circuit class. As we consider the PIT question for tensors, we can also consider the reconstruction problem.

The general reconstruction problem for tensors of degree d and rank r was considered before in the literature [7, 6, 30] where learning algorithms were given for any value of r. However, those algorithms are inherently randomized. Also of note is that the algorithms of [6, 30] output a *multiplicity automata*, which in the context of arithmetic circuits can be thought of as an *arithmetic branching program*. In contrast, the most natural form of the reconstruction question would be to output a degree d tensor.

1.5 Our Results

In this subsection we informally summarize our results. We again stress that our results handle matrices of exactly rank $\le r$, and we consider non-adaptive, deterministic measurements. We have two main results. The first gives a quasi-polynomially sized hitting sets for set-multilinear depth-3 circuits. Our second main result is the connection showing that low-rank recovery reduces to performing sparse-recovery, and that we can use dual Reed-Solomon codes to instantiate the sparse-recovery oracle to achieve a low-rank recovery set that only requires rank-1 (or even sparse) measurements. We find the fact that we can transform an algorithm for a combinatorial property (recovering sparse signals) to an algorithm for an algebraic property (recovering low-rank matrices) quite interesting.

Hitting Sets for Matrices and Tensors.

We begin with constructions of hitting sets for matrices, so as to get black box PIT for quadratic forms. By improving a construction of rank-preserving matrices from Gabizon-Raz [21], we are able to show the following result, which we can then leverage to construct hitting sets.

THEOREM 1.1. *Let $n \ge r \ge 1$. Let $\mathbb{F}$ be a "large" field, and let $\omega \in \mathbb{F}$ have "large" multiplicative order. Let M be an $n \times n$ matrix of rank $\le r$ over $\mathbb{F}$. Let $\hat{f}_M(x,y) = \mathbf{x}^\dagger M\mathbf{y}$ be the bivariate polynomial defined by the vectors $\mathbf{x} \in \mathbb{F}^n$ and $\mathbf{y} \in \mathbb{F}^n$ such that[3] $(\mathbf{x})_i = x^i$ and $(\mathbf{y})_i = y^i$.*

Then M is non-zero iff one of the univariate polynomials $\hat{f}_M(x,x), \hat{f}_M(x,\omega x), \ldots, \hat{f}_M(x,\omega^{r-1}x)$ is non-zero.

Intuitively this says that we can test if the quadratic form f_M is zero by testing whether each of r univariate polynomials are zero. As these univariate polynomials are of degree $< 2n$, it follows that we can interpolate them fully using $2n$ evaluations. As such a univariate polynomial is zero iff all of these evaluations are zero, this yields a $2nr$ sized hitting set. While this only works for "large" fields, we can combine this with results on simulation of large fields to derive results over any field with some loss. This is encapsulated in the next results for black-box PIT, where the log factors are unnecessary over large fields.

THEOREM 1.2. *Let $n \ge r \ge 1$. Let $\mathbb{F}$ be any field, then there is a $\mathsf{poly}(n)$-explicit[4] hitting set for $n \times n$ matrices of rank $\le r$, of size $\mathcal{O}(nr\lg^2 n)$.*

For tensors of higher degree we prove the following theorem.

THEOREM 1.3. *Let $n, r \ge 1$ and $d \ge 2$. Let $n, r \ge 1$, $d \ge 2$. Over any field $\mathbb{F}$, there is an $\mathsf{poly}(n,d,r)^{\mathcal{O}(\lg d)} \cdot \mathcal{O}(\lg(dnr))^d$-explicit hitting set for $[\![n]\!]^d$-tensors of rank $\le r$, of size $dn \cdot \mathcal{O}(dr^2\ln(nd))^{\mathcal{O}(\lg d)} \cdot \mathcal{O}(\lg(dnr))^d$. If $|\mathbb{F}| \ge \mathsf{poly}(n,d,r)$, then there is an $\mathsf{poly}(n,d,r)^{\mathcal{O}(\lg d)}$-explicit hitting set for $[\![n]\!]^d$-tensors of rank $\le r$, of size $dn \cdot \mathcal{O}(dr^2\ln(nd))^{\mathcal{O}(\lg d)}$.*

Thus, if we are allowed access to a field extension to the base field (which is a common assumption in black-box polynomial identity testing), then this yields the first quasi-polynomial sized hitting set for tensors, improving on the $\min(n^d, \mathsf{poly}((nd)^r))$ sized hitting set achievable by invoking the best known results for $\Sigma\Pi\Sigma(k)$ circuits [40]. We note that this result improves on the two obvious hitting sets. The first gives n^d tensors in the hitting set and is $\mathsf{polylog}(n,d,r)$-explicit while the second gives a set of size $\approx dnr$ while not being explicit at all. The above result non-trivially interpolates between these two results.

We also remark here that the actual construction claimed in the above result on tensors has qualitative similarities to a conjectured hitting set of Agrawal [3]. One of the main differences is that our hitting set is on $\lg d$ variables (which we brute-force interpolate), but Agrawal conjectures that a single variable is sufficient.

Low-Rank Recovery.

As mentioned in the previous section, black-box PIT results imply LRR constructions in an information theoretic sense. Thus, the above hitting sets imply LRR constructions but the algorithm for recovery is not implied by the above result. To yield algorithmic results, we actually establish a stronger claim. That is, we first show that the above hitting sets embed a natural sparse-recovery set arising from the dual Reed-Solomon code. Then we develop an algorithm that shows that *any* sparse-recovery set gives rise to a low-rank-recovery set, and that recovery can be performed efficiently given an oracle for sparse recovery. This connection (in the context that any error-correcting code in the hamming metric yields an error-correcting code in the rank-metric) was independently made by Roth [38] (see Theorem 3), who did not give a recovery procedure for the resulting LRR scheme. The next theorem, which is the main result of the paper, shows this connection is also efficient with respect to recovery.

THEOREM 1.4. *Let $n \ge r \ge 1$. Let $\mathcal{V}$ be a t-explicit set of (non-adaptive) measurements for $2r$-sparse-recovery for n-long vectors. Then there is a $\mathsf{poly}(n,t)$-explicit set $\mathcal{H}$, which*

9[3] In this paper, vectors and matrices are indexed from zero, so $\mathbf{x} = (1, x, x^2, \ldots, x^{n-1})^\dagger$.

9[4] A $n \times n$ matrix is t-explicit if each entry can be (deterministically) computed in t steps, where field operations are considered unit cost.

is a (non-adaptive) rank $\leq r$ low-rank-recovery set for $n \times n$ matrices, with a recovery algorithm running in time $\mathcal{O}(rn^2 + n\tau)$, where τ is the amount of time needed to do sparse-recovery from $\mathcal{V}$. Further, $|\mathcal{H}| = 2n|\mathcal{V}|$, and each matrix in $\mathcal{H}$ is n-sparse.

This result shows that sparse-recovery and low-rank recovery (at least in the exact case) are very closely connected. Interestingly, this shows that sparse-recovery (which can be regarded as a combinatorial property) and low-rank recovery (which can be regarded as an algebraic property) are tightly connected. Many fruitful connections have taken this form, such as in spectral graph theory, and perhaps the connection presented here will yield yet further results.

Also, the algorithm used in the above result is purely linear-algebraic, in contrast to the convex optimization approaches that many compressed sensing works use. However, we do not know if the above result is stable to noise, and regard this issue as an important question left open by this work.

When the above result is combined with our hitting set results, we achieve the following LRR scheme for matrices (as well as an LRR scheme for tensors, with parameters similar to the hitting set results mentioned above).

Theorem 1.5. *Let $n \geq r \geq 1$. Over any field $\mathbb{F}$, there is an $\mathsf{poly}(n)$-explicit set $\mathcal{H}$, of $\mathcal{O}(rn \lg^2 n)$ size, such that measurements against $\mathcal{H}$ allow recovery of $n \times n$ matrices of rank $\leq r$ in time $\mathsf{poly}(n)$. Further, the matrices in $\mathcal{H}$ can be chosen to be all rank 1, or all n-sparse.*

We note again that over large fields these logarithmic factors are seen to be unneeded.

Some prior work [20, 19, 18, 14, 38] on LRR focused on finite fields, and as such based their results on BCH codes. The above result is based on (dual) Reed-Solomon codes, and as such works over any field (when combined with results allowing simulation of large fields by small fields). Other prior work [37] on exact LRR permitted randomized measurements, while we achieve deterministic measurements.

Further, we are able to do LRR with measurements that are either all n-sparse, or all rank-1. As Roth [38] independently observed, the n-sparse LRR measurements can arise from any (hamming-metric) error-correcting code (but he did not provide decoding). Tan, Balzano and Draper [42] showed that random $(n \lg n)$-sparse measurements provide essentially the same low-rank recovery properties as random measurements. Thus, our results essentially achieve this deterministically.

We further observe that a specific code (the dual Reed-Solomon code) allows a change of basis for the measurements, and in this new basis the measurements are all rank 1. Recht et al. [37] asked whether low-rank recovery was possible when the measurements were rank 1 (or "factored"), as such measurements could be more practical as they are simpler to generate and store in memory. Thus, our construction answers this question in the positive direction, at least for exact LRR.

Rank-Metric Codes.

Appealing to the connection between LRR and rank-metric codes, we achieve the following constructions of rank-metric codes.

Theorem 1.6. *Let $\mathbb{F}$ be any field, $n \geq 1$ and $1 \leq r \leq n/2$. Then there are $\mathsf{poly}(n)$-explicit rank-metric codes with $\mathsf{poly}(n)$-time decoding for up to r errors, with parameters $[[n]^2, n^2 - 4(n-r)r \cdot \mathcal{O}(\lg^2 n), 2r+1]_{\mathbb{F}}$, and the parity checks on this code can be chosen to be all rank-1 matrices, or all n-sparse matrices.*

Earlier work on rank-metric codes over finite fields [20, 19, 18, 14, 38] achieved $[[n]^2, n(n-2r), 2r+1]_{\mathbb{F}_q}$ rank-metric codes, with efficient decoding algorithms. These are optimal (meeting the analogue of the Singleton bound for rank-metric codes). However, these constructions only work over finite fields. While our code achieves a worse rate, its construction works over any field, and over infinite fields the $\mathcal{O}(\lg^2 n)$ term is unneeded. Further, Roth [38] observed that the resulting $[[n]^2, (n-2r)^2, 2r+1]$ code is optimal (see discussion of his Theorem 3) over algebraically closed fields (which are infinite).

We are also able to give rank-metric codes over tensors, which can correct errors up to rank $\approx n^{d/\lg d}$ (out of a maximum n^{d-1}), while still achieving constant rate. The rank-metric code arising from the naive low-rank recovery never achieves constant rate, and prior work by Roth [39] only gave decoding against a constant number of errors.

Theorem 1.7. *Let $\mathbb{F}$ be any field, $n, r \geq 1$ and $d \geq 2$. Then there are $\mathsf{poly}((nd)^d, r^{\lg d})$-explicit rank-metric codes with $\mathsf{poly}((nd)^d, r^{\lg d})$-time decoding for up to r errors, with parameters $[[n]^d, n^d - \mathcal{O}(d^2 n r^{\lg d} \lg(dn)), 2r+1]_{\mathbb{F}}$.*

We note here that our decoding algorithm will return the *entire* tensor, which is of size n^d. Trivially, any algorithm returning the entire tensor must take at least n^d time. In this case, the level of explicitness of the code we achieve is reasonable. However, a more desirable result would be for the algorithm to return a rank $\leq r$ representation of the tensor, and thus the n^d lower bound would not apply so that one could hope for faster decoding algorithms. Unfortunately, even for $d = 3$ an efficient algorithm to do so would imply P = NP. That is, if an algorithm (even one which is not a rank-metric decoding or low-rank recovery algorithm) could produce a rank $\leq r$ decomposition for any rank $\leq r$ tensor, then one could compute tensor-rank as it is the minimum r such that the resulting rank $\leq r$ decomposition actually computes the desired tensor (this can be checked in $\mathsf{poly}(n^d)$ time). However, Håstad [24] showed that tensor-rank (over finite fields) is NP-hard for any fixed $d \geq 3$. It follows that for any (fixed) $d \geq 3$, if one could recover (even in $\mathsf{poly}(n^d)$-time) a rank $\leq r$ tensor into its rank $\leq r$ decomposition, then P = NP. While it is conceivable to instead output a rank $\leq r'$ decomposition, for some r' based on r, we do not explore this idea. We only discuss recovery of a tensor by reproducing its entire list of entries, as opposed to its more concise representation.

Finally, we remark that in [39] Roth discussed the question of decoding rank-metric codes of degree $d = 3$, gave decoding algorithms for errors of rank 1 and 2, and wrote that "Since computing tensor rank is an intractable problem, it is unlikely that we will have an efficient decoding algorithm … otherwise, we could use the decoder to compute the rank of any tensor. Hence, if there is any efficient decoding algorithm, then we expect such an algorithm to recover the error tensor without necessarily obtaining its rank. Such an algorithm, that can handle any prescribed number of errors, is not yet known." Thus, our work gives the first such algorithm for tensors of degree $d > 2$.

1.6 Proof Overview

In this section we give proof outlines of the results mentioned so far.

Hitting Sets for Matrices.

The main idea for our hitting set construction is to reduce the question of hitting (non-zero) $n \times n$ matrices to a question of hitting (non-zero) $r \times r$ matrices. Once this reduction is performed, we can then run the naive hitting set, which queries all r^2 entries. This can loosely be seen in analogy with the kernelization process in fixed-parameter tractability, where a problem depending on the input size, n, and some parameter, k, can be solved by first reducing to an instance of size $f(k)$, and then brute-forcing this instance.

To perform this kernelization, we first note that any $n \times n$ matrix M of rank exactly r can be written as $M = PQ^\dagger$, where P and Q are $n \times r$ matrices of rank exactly r. To reduce M to an $r \times r$ matrix, it thus suffices to reduce P and Q each to $r \times r$ matrices, denoted P' and Q'. As this reduction must preserve the fact that M is non-zero, we need that $P'Q' \neq 0$. We enforce this requirement by insisting that P' and Q' are also rank exactly r, so that $M' = P'Q'$ is also non-zero.

To achieve this rank-preservation, we turn to a lemma of Gabizon-Raz [21] (we note that this lemma has been used before for black-box PIT [28, 40]). They gave an explicit family of $\mathcal{O}(nr^2)$-many $r \times n$-matrices $\{A_\ell\}_\ell$, such that for any P and Q of rank exactly r, at least one matrix A_ℓ from the family is such that $\mathrm{rank}(A_\ell P) = \mathrm{rank}(A_\ell Q) = r$. Translating this result into our problem, it follows that one of the $r \times r$ matrices $A_\ell M A_\ell^\dagger$ is full-rank. The (i, j)-th entry of $A_\ell M A_\ell^\dagger$ is $\langle M, (A_\ell)_i (A_\ell)_j^\dagger \rangle$, where $(A_\ell)_i$ is the i-th row of A_ℓ. It follows that querying each entry in these $r \times r$ matrices corresponds to a rank 1 measurement of M, and thus make up a hitting set. As there were $\mathcal{O}(nr^2)$ choices of ℓ and r^2 choices of (i, j), this gives a $\mathcal{O}(nr^4)$-sized hitting set.

To achieve a smaller hitting set, we use the following sequence of ideas. First, we observe that in the above, we can always assume $i = 0$. Loosely, this is because when $A_\ell M A_\ell^\dagger$ is full-rank, none of its rows are all zero. Thus, only the first row of $A_\ell M A_\ell^\dagger$ needs to be queried to determine this. Second, we improve upon the Gabizon-Raz lemma, and provide an explicit family of rank-preserving matrices with size $\mathcal{O}(nr)$. This follows from modifying their construction so the degree of a certain determinant is smaller. To ensure that the determinant is a non-zero polynomial, we show that it has a unique monomial that achieves maximal degree, and that the term achieving maximal degree has a non-zero coefficient as a Vandermonde determinant (formed from powers of an element ω, which has large multiplicative order) is non-zero. Finally, we observe that the hitting set constraints can be viewed as constraints regarding polynomial interpolation. This view shows that some of the constraints are linearly-dependent, and thus can be removed. Each of the above observations saves a factor of r in the size of the hitting set, and thus produces an $\mathcal{O}(nr)$-sized hitting set.

Low-Rank Recovery.

Having constructed hitting sets, a standard argument implies that the same construction yields low-rank-recovery sets. As this lemma does not provide a recovery algorithm, we provide one. To do so, we must first change the basis of our hitting set. That is, the hitting set $\mathcal{B}$ yields a set of constraints on a matrix M, and we are free to choose another basis for these constraints, which we call $\mathcal{D}$. The virtue of this new basis is that each constraint is non-zero only on some k-diagonal (the entries (i, j) such that $i + j = k$). It turns out that these constraints are the parity checks of a dual Reed-Solomon code with distance $\Theta(r)$. This code can be decoded efficiently using standard BCH decoding techniques, such Berlekamp-Massey [8, 33] or Gorenstein-Peterson-Zierler (which is also known as Prony's method [13], which was developed in 1795). In the full version of this work, we give an exposition of the Gorenstein-Peterson-Zierler method, where we show how to syndrome-decode this code up to half its minimum distance, counting erasures as half-errors. Thus, given a $\Theta(r)$-sparse vector (which can be thought of as errors from the vector $\mathbf{0}$) these parity checks impose constraints from which the sparse vector can be recovered. Put another way, our low-rank-recovery set naturally embeds a sparse-recovery set along each k-diagonal.

Thus, in designing a recovery algorithm for our low-rank recovery set, we do more and show how to recover from any set of measurements which embed a sparse-recovery set along each k-diagonal. In terms of error-correcting codes, this shows that any hamming-metric code yields a rank-metric code over matrices, and that decoding the rank-metric code efficiently reduces to decoding the hamming-metric code.

To perform recovery, we introduce the notion of a matrix being in $(< k)$-upper-echelon form. Loosely, this says that $M^{(<k)}$, the entries (i, j) of the matrix with $i + j < k$, are in row-reduced echelon form. We then show that for any matrix M in $(< k)$-upper-echelon form, the k-diagonal is $2\,\mathrm{rank}(M)$-sparse. As an example, suppose $M^{(<k)}$ was entirely zero. It follows then that M is in $(< k)$-upper-echelon form. Further, the rows that have non-zero entries on the k-diagonal of M are then linearly-independent, as they form a triangular system. It follows that the k-diagonal can only have $\mathrm{rank}(M)$ non-zero entries. The more general case is slightly more complicated technically, but not conceptually. Thus, this echelon-form translates the notion of low-rank into the notion of sparsity.

The algorithm then follows naturally. We induct on k, first putting $M^{(<k)}$ into $(< k)$-upper-echelon form (using row-reduction), and then invoking a sparse-recovery oracle on the k-diagonal of M to recover it. This then yields $M^{(\leq k)}$, and we increment k. However, as described so far, the use of the sparse-recovery oracle is adaptive. We show that the row-reduction procedure can be understood such that the adaptive use of the sparse-recovery oracle can be simulated using non-adaptive calls to the oracle. More specifically, we will apply the measurements of the sparse-recovery oracle on each k-diagonal of M (which may not be sparse), and show how to compute the measurements of the adaptive algorithm (where the k-diagonals are sparse) from the measurements made. Putting these steps together, this shows that exact non-adaptive low-rank-recovery reduces to exact non-adaptive sparse-recovery. Instantiating this claim with our hitting sets from above gives a concrete low-rank-recovery set, with accompanied recovery algorithm.

Hitting Sets and Low-Rank Recovery for Tensors.

The results for matrices naturally generalize to tensors in the sense that an $[\![n]\!]^{2d}$ tensor can be viewed as an $[\![n^d]\!]^2$ matrix. However, we can do better. Specifically, the hitting set results were done via *variable reduction*, which shows that a rank $\le r$ bivariate polynomial

$$f_M(x, y) = (1, x, x^2, \dots, x^{n-1})M(1, y, y^2, \dots, y^{n-1})^\dagger$$

is zero iff a set of r univariate polynomials are all zero. Further, the degrees of these polynomials is only twice the original degree. As each univariate polynomial can be interpolated using $\mathcal{O}(n)$ measurements, this yields $\mathcal{O}(nr)$ measurements total. This motivates the more general idea of treating a degree d tensor as a d-variate polynomial, and showing that we can test whether this polynomial is zero by testing if a collection of d'-variate polynomials are zero, for $d' < d$. Recursing on this procedure then reduces the d-variate case to the univariate case, and the univariate case is brute-force interpolated.

The recursion scheme we develop for this is to show that a d-variate polynomial is zero iff r $d/2$-variate polynomials are zero, and this naturally leads to an $\mathcal{O}(dnr^{\lg d})$-sized hitting set. To prove its correctness, we show that the bivariate case (corresponding to matrices) applied to two groups of variables allows us to reduce to a single group of variables (with an increase in the number of polynomials to test). Finally, since we saw how to do low-rank recovery for matrices, and the tensor-case essentially only uses the matrix case, we can also turn this hitting set procedure into a low-rank recovery algorithm.

However, the above scheme will require elements of order $\approx n^d$, which makes the scheme not very explicit. Improving upon an earlier version of this paper, we show how techniques from Klivans and Spielman [31] can be used to reduce the need for an element of large order. This allows the hitting set to be both quasi-polynomial sized and constructible in quasi-polynomial time.

Simulation of Large Fields by Small Fields.

Most all of the results mentioned require a field of size $\approx \mathsf{poly}(n, d, r)$. When getting results over small fields, we show that, with some loss, we can simulate such large fields inside the hitting sets. We break-up each tensor H in the original hitting set into new tensors $\tilde{H}_i$ such that for any $\mathbb{F}$-tensor T, $\langle T, H\rangle$ can be reconstructed from the set of values $\{\langle T, \tilde{H}_i\rangle\}_i$. To do so, we use the well-known representation of a extension field $\mathbb{K}$ of $\mathbb{F}$ as a field of matrices over $\mathbb{F}$. As the entries of a rank-1 tensor are multiplications of d elements of $\mathbb{K}$, we can expand these multiplications out as iterated matrix multiplications, which yields $(\dim_{\mathbb{F}} \mathbb{K})^{d+1}$ terms to consider, each of which corresponds to some $\tilde{H}_i$.

Note that if we did not insist on generating rank-1 matrices over the small field, then we could achieve this with a loss of only one log factor, by simply expanding each tensor to $\log(\mathsf{poly}(n, d, r))) = \mathcal{O}(\log(ndr))$ tensors, where the ith tensor consists of the ith bit of each of the entries of the original tensor.

Rank-Metric Codes.

The above techniques give the existence of low-rank recovery sets (and corresponding algorithms) for tensors, over any field. Via the connections presented in Section 1.3, this readily yields rank-metric codes with corresponding parameters.

2. DISCUSSION

We briefly discuss some directions for further research.

Reducing Noisy Low-Rank Recovery to Noisy Sparse Recovery.

We show that low-rank-recovery of matrices can be done using any sparse-recovery oracle. This reduction was for non-adaptive measurements, and was done in the presence of no noise. As much of the compressed sensing community is interested in the noisy case (so M is only close to rank $\le r$) the main open question of this work is whether the reduction extends to the noisy case.

Smaller Hitting Sets.

While the observations of Roth [38] show that our hitting set for matrices is optimal over algebraically closed fields, our results over tensors with $d > 2$ are much larger than the existential bounds. Can these hitting sets be improved to size $\mathcal{O}(\mathsf{poly}(d)n^{o(d)}r^k)$ for $k = \mathcal{O}(1)$? As mentioned in the preliminaries section in the full version, any such hitting set with $k < 2$ would yield improved tensor-rank lower bounds. However, as the best tensor-rank lower bounds for $d = 3$ are $\Theta(n)$ and our hitting set (over infinite fields) yields this bound (with a smaller constant), even improving our hitting set for $d = 3$ by constant factors could yield interesting new results. Specifically, for $d = 3$ can one construct (say over infinite fields) a hitting set of size $\le nr^2/10$ for $[\![n]\!]^3$ tensors of rank $\le r$?

Large Field Simulation.

Our results show that hitting sets (and low-rank recovery sets) that involve tensors over an extension field imply hitting sets (and LRR sets) over the base field. While we show that we can preserve the rank-1 property of these tensors while doing so, it introduces an $\exp(d)$ factor in the size of the hitting set. Can this be improved?

Connection to rigidity.

In [43] Valiant defined the notion of an (r, k)-rigid matrix and proved that if a matrix M is $(n/\log\log(n), n^\epsilon)$-rigid then the function $f(x) = M \cdot x$ cannot be computed by a linear sized logarithmic depth linear circuit. A matrix M is (r, k)-rigid if one cannot have $M = A+B$ where $\mathrm{rank}(A) \le r$ and each row in B has at most k nonzero elements (we call such B a k-row sparse matrix). Note that any k-row sparse B defines a sparse bilinear form $B(x, y) = x^t By$. Namely, a bilinear form that has only nk nonzero monomials. Imagine that we could find a hitting set $\mathcal{H}$ for the class of all matrices that can be written as a sum of rank $\le r$ matrix and a k-row sparse matrix. Then, as before, any nonzero matrix in the dual of $\mathcal{H}$ will be (r, k)-rigid. In this work we constructed optimal hitting sets for rank $\le r$ matrices and in earlier works PIT algorithms for sparse polynomials were given (see e.g. the survey [41]). If we could find a way to combine the two hitting sets then we will be able to find an explicit rigid matrix, which is a major open problem in circuit complexity.

Acknowledgements

We would like to thank Olgica Milenkovic for pointing us to the low-rank recovery problem, and Madhu Sudan for some helpful comments regarding decoding dual Reed-Solomon codes. We thank Zeev Dvir for pointing out the connection of our work to the problem of constructing rigid matrices.

Part of this work was done while the first author was visiting Stanford University, as well as when the second author was visiting the Bernoulli center at EPFL.

3. REFERENCES

[1] http://dsp.rice.edu/cs.

[2] http://perception.csl.uiuc.edu/matrix-rank/.

[3] M. Agrawal. Proving lower bounds via pseudo-random generators. In *Proceedings of the 25th FSTTCS*, volume 3821 of *LNCS*, pages 92–105, 2005.

[4] M. Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual FOCS*, pages 67–75, 2008.

[5] E. Arias-Castro, E. J. Candes, and M. Davenport. On the fundamental limits of adaptive sensing. *arXiv:1111.4646*, 2011.

[6] A. Beimel, F. Bergadano, N. H. Bshouty, E. Kushilevitz, and S. Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000.

[7] F. Bergadano, N. H. Bshouty, and S. Varricchio. Learning multivariate polynomials from substitution and equivalence queries. *ECCC*, 3(8), 1996.

[8] E. R. Berlekamp. *Algebraic coding theory.* McGraw-Hill Book Co., New York, 1968.

[9] E. Candes, J. Romberg, and T. Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *arXiv:math/0503066*, Mar. 2005.

[10] E. J. Candes and Y. Plan. Matrix Completion With Noise. *ArXiv e-prints*, Mar. 2009.

[11] E. J. Candés and Y. Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Trans. Inform. Theory*, 57(4):2342–2359, 2011.

[12] E. J. Candes and T. Tao. The Power of Convex Relaxation: Near-Optimal Matrix Completion. *ArXiv e-prints*, Mar. 2009.

[13] G. C. F. M. R. de Prony. Essai éxperimental et analytique: sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l'alkool, à différentes températures. *Journal de l'école Polytechnique*, 1:24–76, 1795.

[14] P. Delsarte. Bilinear forms over a finite field, with applications to coding theory. *J. Combin. Theory Ser. A*, 25(3):226–241, 1978.

[15] Z. Dvir and A. Shpilka. Towards dimension expanders over finite fields. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC)*, pages 304–310, 2008.

[16] Y. C. Eldar, D. Needell, and Y. Plan. Unicity conditions for low-rank matrix recovery. *arXiv:1103.5479*, 2011.

[17] A. Emad and O. Milenkovic. Information theoretic bounds for tensor rank minimization over finite fields. *arXiv:1103.4435*, 2011.

[18] E. M. Gabidulin. Optimal array error-correcting codes. *Probl. Peredach. Inform.*, 21(2):102–106, 1985.

[19] E. M. Gabidulin. Theory of codes with maximum rank distance. *Probl. Inform. Transm.*, 21(1):1–12, 1985.

[20] E. M. Gabidulin and V. I. Korzhik. Codes correcting lattice-pattern errors. *Zzvestiya VUZ. Radioelektronika*, 1972.

[21] A. Gabizon and R. Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008.

[22] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *ArXiv e-prints*, Oct. 2009.

[23] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.*, 105:150401, Oct 2010.

[24] J. Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990.

[25] J. Heintz and C. P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the 12th annual STOC*, pages 262–272, 1980.

[26] P. Indyk, E. Price, and D. P. Woodruff. On the Power of Adaptivity in Sparse Recovery. *Foundations of Computer Science*, Oct. 2011.

[27] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[28] Z. S. Karnin and A. Shpilka. Deterministic black box polynomial identity testing of depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 23rd Annual CCC*, pages 280–291, 2008.

[29] A. Khajehnejad, S. Oymak, and B. Hassibi. Subspace expanders and matrix rank minimization. *arXiv:1102.3947v1*, 2011.

[30] A. Klivans and A. Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(10):185–206, 2006.

[31] A. Klivans and D. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual STOC*, pages 216–223, 2001.

[32] A. Lubotzky and Y. Zelmanov. Dimension expanders. *J. Algebra*, 319(2):730–738, 2008.

[33] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. Information Theory*, IT-15:122–127, 1969.

[34] R. Meshulam. Spaces of Hankel matrices over finite fields. *Linear Algebra Appl.*, 218:73–76, 1995.

[35] N. Nisan and A. Wigderson. Lower bound on arithmetic circuits via partial derivatives. *Computational Complexity*, 6:217–234, 1996.

[36] R. Raz and A. Shpilka. Deterministic polynomial identity testing in non commutative models. *Computational Complexity*, 14(1):1–19, 2005.

[37] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[38] R. M. Roth. Maximum-rank array codes and their application to crisscross error correction. *IEEE*

Transactions on Information Theory, 37(2):328–336, 1991.

[39] R. M. Roth. Tensor codes for the rank metric. *IEEE Transactions on Information Theory*, 42(6):2146–2157, 1996.

[40] N. Saxena and C. Seshadhri. Blackbox identity testing for bounded top fanin depth-3 circuits: the field doesn't matter. In *Proceedings of the 43rd Annual STOC*, pages 431–440, 2011.

[41] A. Shpilka and A. Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[42] V. Y. F. Tan, L. Balzano, and S. C. Draper. Rank minimization over finite fields: Fundamental limits and coding-theoretic interpretations. *arXiv:1104.4302v2*, 2011.

[43] L. G. Valiant. Graph-theoretic arguments in low-level complexity. In *Lecture notes in Computer Science*, volume 53, pages 162–176. Springer, 1977.