

UC San Diego

UC San Diego Previously Published Works

Title

Guest Column

Permalink

<https://escholarship.org/uc/item/80d93481>

Journal

ACM SIGACT News, 52(2)

ISSN

0163-5700

Authors

Knop, A
Lovett, S
McGuire, S
[et al.](#)

Publication Date

2021-06-14

DOI

10.1145/3471469.3471479

Peer reviewed

Guest Column: Models of computation between decision trees and communication ¹

A. Knop² S. Lovett³ S. McGuire⁴ W. Yuan⁵



Abstract

We survey recent work on the communication complexity of functions $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ of the form $f(x \circ y)$ where $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is a total boolean function and \circ represents either bit-wise XOR or bit-wise AND. This naturally leads to the study of models of computation that are, in a sense, ‘between’ communication complexity and decision tree complexity. These classes of functions capture a rich class of communication problems while simultaneously being amenable to analysis with minimal assumptions about the structure of f . This flexibility has shed new light on central topics in communication complexity, including restricted cases of the log-rank conjecture and the query-to-communication lifting methodology.

1 Introduction

Communication complexity studies the amount of communication necessary to compute a function whose value depends on information distributed among several entities. Yao [Yao79] initiated the study of communication complexity more than 40 years ago, and it has since become a central field in theoretical computer science with many applications in diverse areas such as data structures, streaming algorithms, property testing, approximation algorithms, coding theory, and machine learning. The textbooks [KN06,RY20] provide excellent overviews of the theory and its applications.

In the basic version of communication complexity, two players, call them Alice and Bob, wish to compute a function $F : X \times Y \rightarrow \{0, 1\}$, where X and Y are some finite sets. Alice holds an input $x \in X$, Bob holds an input $y \in Y$, and they wish to compute $F(x, y)$ by sending messages back and forth according to some *protocol*. Importantly, Alice and Bob have arbitrary computational power, as we are interested only in how much information must be exchanged to compute the function. The goal is to design *low cost* protocols, measured in terms of the number of bits Alice and Bob exchange (in the worst case) and, ideally, we would show tight upper *and* lower bounds on the communication complexity of the communication problem of interest. Let $D^{\text{cc}}(F)$ denote the lowest achievable cost for a deterministic protocol computing F correctly on all inputs.

¹©A. Knop, S. Lovett, S. McGuire, W. Yuan, 2021.

²Mathematics, University of California, San Diego. Email: aaknop@gmail.com

³Computer Science and Engineering, University of California, San Diego. Research supported by NSF award 2006443. Email: slovett@ucsd.edu

⁴Computer Science and Engineering, University of California, San Diego. Research supported by NSF award 1909634 and a Simons Investigator award. Email: shmcguir@eng.uscd.edu

⁵Institute for Interdisciplinary Information Sciences, Tsinghua University. Email: yuanwq17@mails.tsinghua.edu.cn

Many of the celebrated results in the field concern the communication complexity of important *concrete functions*, such as Set Disjointness [Raz92a] and Gap Hamming Distance [CR12]. Unfortunately, the understanding of communication complexity of *arbitrary functions* is still lacking. Probably the most famous problem of this type is the so-called *log-rank conjecture*, formulated by Lovász and Saks [LS88]. The conjecture asserts that $D^{\text{cc}}(F)$ can be (roughly) characterized by the rank of the $|X| \times |Y|$ boolean matrix $M(F) \in \{0, 1\}^{X \times Y}$ defined as $M(F)_{x,y} = F(x, y)$.

The difficulty in analyzing the communication complexity of arbitrary functions, even in the deterministic case, has led researchers to study a special class of functions called ‘lifted functions’. These are functions of the form $f \circ g^n$, where $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is an arbitrary boolean function and $g : X \times Y \rightarrow \{0, 1\}$ is a two-party function. Typically f is called an *outer function* and g is called a *gadget*. Formally, $F = f \circ g^n$ is a two-party function $F : X^n \times Y^n \rightarrow \{0, 1\}$ defined as

$$F((x_1, \dots, x_n), (y_1, \dots, y_n)) = f(g(x_1, y_1), \dots, g(x_n, y_n)).$$

This framework captures a number of the closely-studied concrete functions in communication complexity. For example, Set Disjointness is obtained by taking f as the OR of n bits and g as the AND of 2 bits, and Equality is obtained by taking f as the AND of n bits and g as the negated XOR of 2 bits. In general, the size of g ’s domain is allowed to be non-constant; namely, depend on n , the number of inputs for the function f .

Studying lifted functions has been quite successful, leading to, for example, a resolution of the clique vs. independent set problem [Gö15, GPW18a], separations between monotone and non-monotone circuits [GKRS18], NP-hardness of automating various proof systems [GKMP20], lower bounds on semi-algebraic proof systems [GP18], and sub-exponential size lower bounds for approximating CSPs with linear programming relaxations [KMR17].

This success is, in broad terms, due to the fact that the communication complexity of lifted functions $F = f \circ g^n$ can often be bounded in terms of different *query complexity* measures of the outer function f . Such relationships allow us to ‘lift’ query complexity lower bounds for f into communication complexity lower bounds for F , so we will refer to them as ‘lifting theorems’ (an imprecise definition, though it will be clear after some examples). This is useful because proving query complexity lower bounds is generally easier than proving communication complexity lower bounds. The query complexity of a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is the number of ‘queries’ an (in general, adaptive) query algorithm must make to its input $z \in \{0, 1\}^n$ in order to compute $f(z)$, where the objective is to minimize the number of queries made and the exact notion of a query depends on the particular model. For studying lifted functions $f \circ g^n$, the query model we use will depend on the choice of gadget g .

The simplest type of query model — the deterministic decision tree — can adaptively ask questions of the form ‘what is the value of some input bit?’. For example, suppose we want to find the single ‘1’ in a Hamming weight one, n -bit string. The obvious algorithm is to simply check every bit position, and it is not too hard to see that checking every bit is essentially necessary: an adversary can always delay their choice of where the ‘1’ will go until the decision tree has queried $n - 1$ positions without answering any two queries inconsistently. Decision tree complexity is polynomially related to a number of other complexity measures of boolean functions, including certificate complexity, block sensitivity, sensitivity, and related algebraic notions such as degree and approximate degree. This equivalence is due to Nisan-Szegedy [NS94] and the recent breakthrough of Huang [Hua19].⁶

⁶For background on decision trees and their related complexity measures, see the beautiful survey of [BdW02].

Despite the model’s simplicity, the decision tree complexity of f is roughly equivalent to the communication complexity of F when we choose an appropriate gadget g . A concrete example of such a gadget is the m -bit *indexing gadget*, defined by

$$\text{Ind}_{\log m, m} : [m] \times \{0, 1\}^m \rightarrow \{0, 1\}; \text{Ind}_{\log m, m}(x, y) = y_x$$

where $m \sim \text{poly}(n)$. In this case, Raz-McKenzie [RM99] and Göös-Pitassi-Watson [GPW17] showed that F ’s communication complexity is equivalent to f ’s decision tree complexity, up to a $\log m$ factor. However, this relationship is only known to work if g satisfies a pseudorandomness condition [CFK⁺19b] and its domain $X \times Y$ has size at least polynomial in n . We are instead interested in understanding structured choices of g with constant domain size $|X| = |Y| = O(1)$, which are not captured by this setting.

Sherstov addressed the constant-sized gadget setting in [She10], showing that $D^{\text{cc}}(f \circ g^n)$ is lower bounded by a polynomial in $D^{\text{dt}}(f)$ when g embeds both AND and OR as sub-functions. More specifically, Sherstov showed that the communication complexity of either $f \circ \wedge^n$ or $f \circ \vee^n$ is at least the *degree of f as a multilinear polynomial* (known to be polynomially related to the decision tree complexity of f), where \wedge is a one-bit AND gadget and \vee is a one-bit OR gadget.⁷

While sufficient for Sherstov’s applications, the lower bound does not explain *which* of $f \circ \wedge^n$ or $f \circ \vee^n$ has large communication complexity. We are interested in obtaining more explicit communication lower bounds on F .

This survey will primarily focus on cases when $g : \{0, 1\}^2 \rightarrow \{0, 1\}$ is either the AND or the XOR of the two input bits, which are the simplest non-trivial gadgets with constant domain size $|X| = |Y| = O(1)$. Unlike the previous two cases [RM99, GPW17, She10], we consider query models which are *strictly stronger than standard decision trees*, which can be thought of as ‘intermediate models’ between standard decision trees and communication complexity. Efficient query algorithms in these stronger models are connected to a number of interesting structural properties of f distinct from (but related to) the complexity measures classically studied in query complexity.

1.1 Lifted functions with one-bit gadget

We now introduce the main objects of study, which are lifted functions where the gadget is a simple one-bit function. There are only two non-degenerate and non-equivalent one-bit gadgets: AND and XOR, as all other ones can be obtained from these by either flipping the inputs or outputs of the gadget g .

Definition 1.1. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a boolean function. Its corresponding AND-function is $f_{\wedge} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ and XOR-function is $f_{\oplus} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, defined as*

$$\begin{aligned} f_{\wedge}(x, y) &= f(x \wedge y) \\ &\text{and} \\ f_{\oplus}(x, y) &= f(x \oplus y). \end{aligned}$$

Here $x, y \in \{0, 1\}^n$, $x \wedge y$ is entry-wise AND, and $x \oplus y$ is entry-wise XOR.

Many natural communication problems can be cast as either of these models. For example, the Equality function is an XOR-function and Set Disjointness is an AND-function.

⁷That is, $\wedge : \{0, 1\}^2 \rightarrow \{0, 1\}$ is defined as $\wedge(x, y) = x \wedge y$ for $x, y \in \{0, 1\}$, and \vee is defined analogously.

XOR-functions correspond to the query model of parity decision trees, which we abbreviate as PDTs. This model extends the standard decision tree model by allowing nodes to query the parity (namely, XOR) of an arbitrary subset of the input bits. To see the connection between PDTs and XOR-functions, notice that if f can be computed by a PDT of depth d , then f_{\oplus} has a deterministic communication protocol which sends $2d$ bits (that is, $D^{\text{cc}}(f_{\oplus}) \leq 2d$), as the players can simulate the computation by the PDT. Let $z = x \oplus y$. If at a given node the PDT needs to compute a parity $\bigoplus_{i \in S} z_i$, then the first player computes the parity on her input $\bigoplus_{i \in S} x_i$, the second player computes the parity on his input $\bigoplus_{i \in S} y_i$, and exchanging the answers and XORing them reveals the required parity on z , which allows both players to move to the correct child of the node. PDTs and XOR-functions have both been studied in complexity theory, either from a communication complexity perspective [MO09, ZS09, LLZ11, Zha14, HHL18] or a boolean function analysis perspective [ZS10, TWXZ13, OWZ⁺14, BCK14, Yao15, BTW15, San19].

Similarly, AND-functions correspond to the query model of AND decision trees, which we abbreviate as ADTs. This model extends the standard decision tree model by allowing nodes to query the AND of an arbitrary subset of the input bits. Similar to PDTs, if f can be computed by an ADT of depth d , then f_{\wedge} can be computed by a communication protocol which sends $2d$ bits, by simulating the computation of the ADT. ADTs are much less studied than PDTs; to the best of our knowledge, the only papers studying them are [LM19, KLMY21].

Since PDTs and ADTs are intermediate models between decision trees and communication complexity, it seems natural to study PDT and ADT complexity of arbitrary functions. This, in turn, implies necessity of study of communication complexity of AND- and XOR-functions due to aforementioned connection.

1.2 Lifting with complicated gadgets

Next, we compare our setting to lifting theorems appearing in the literature which use ‘complicated’ gadgets. Let $D^{\text{dt}}(f)$ denote the number of queries used in the optimal deterministic decision tree for f . Göös, Pitassi and Watson [GPW17], extending the work of Raz and McKenzie [RM99], showed the following:

Theorem 1.2 (D^{dt}-to-D^{cc} lifting, [RM99, GPW17]). *For a fixed n , let $m = \text{poly}(n)$ and $g = \text{Ind}_{\log m, m}$. Then for any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$,*

$$D^{\text{cc}}(f \circ g^n) = D^{\text{dt}}(f) \cdot \Theta(\log n).$$

One direction of the equality — designing a communication protocol from a decision tree — is easy. Trivially, $D^{\text{cc}}(\text{Ind}_{\log m, m}) = O(\log m)$, as Alice can simply send her input to Bob (and this is tight). Alice and Bob can simulate f ’s decision tree and compute the i -th copy of g using the trivial protocol whenever the decision tree for f queries its i -th input bit. The other direction uses a much more sophisticated argument, constructing a decision tree by simulating the communication protocol round-by-round on an ‘unknown input’ while paying careful attention to which copies of g Alice and Bob appear to know a lot about. See [RM99, GPW17] for more details.

A drawback of known simulation-type arguments is their reliance on ‘pseudorandom’ choices of g , of which $\text{Ind}_{\log m, m}$ is a special case. More generally, g needs to resemble a ‘two-source extractor’, meaning $g(A, B)$ is close to unbiased if $A \subseteq X$ and $B \subseteq Y$ are large enough.⁸ We want

⁸See Chattopadhyay et al. [CFK⁺19b], which gives a general simulation argument that works for any g with sufficiently low *discrepancy*. Indeed, this property is satisfied by random choices of g .

to understand choices of g which are very much *not* pseudorandom, and it's unclear how to adapt simulation-type arguments to handle this.

A crucial parameter in Theorem 1.2 is the size m of the gadget. If g 's domain has size polynomial in n , then we would incur a factor which depends on the size of the universe. For applications, this is undesirable: we would like to obtain a weaker dependence on n (or even no dependence). Lovett et al. [LMM⁺20] address this issue by proving an analogue of Theorem 1.2 where $m = n^{1+\epsilon}$ is nearly-linear, and conjecture that a poly-logarithmic dependence is obtainable. Additionally, [GKMP20] (reproved in [LMM⁺20] with an alternative argument) show that it suffices to pick $m \geq D^{\text{dt}}(f)^{O(1)}$ for the size of the gadget in Theorem 1.2. This improves the $m = n^{1+\epsilon}$ when $D^{\text{dt}}(f)$ is a sufficiently small polynomial in n . In contrast, our focus is on the case where $m = 2$.

Finally, it is important to consider the possible choices for f when considering ‘small and structured gadgets’ versus ‘large and pseudorandom gadgets’. Theorem 1.2 can be stated more generally for *search problems*, where we can replace f by any search relation. This flexibility is important for applications in, e.g., proof complexity [dRNV16, BPR95] and circuit complexity [RM99], and it is possible because simulation arguments tend to use very little (or no) information about the outer function. Our context is, in some sense, the opposite: we choose very small gadgets g with no pseudorandomness properties, but transfer the burden onto understanding the structure of f .

Gadget	Query Model	Communication Model	Total Functions	Reference
$\text{Ind}_{\log m, m}$	DT	deterministic	No	[RM99]
	probabilistic DT	bounded error probabilistic	No	[GPW17]
IP_n	DT	deterministic	No	[CFK ⁺ 19b]
	probabilistic DT	bounded error probabilistic	No	[CFK ⁺ 19a]
EQ_n	ADT	deterministic	No	[LM19]
\oplus	PDT	deterministic	Yes	[HHL18]
\wedge	ADT	deterministic	Yes	[KLMY21]
g	polynomial degree	rank	Yes	[She10]

Figure 1: Query-to-communication lifting theorems. m in the first line is polynomial in n . g in the last line is any function that has as sub-functions both an AND and an OR.

1.3 The log-rank conjecture

A noteworthy example of a problem for which the connection between AND-/XOR-functions and ADTs/PDTs has been fruitful is the *log-rank conjecture* of Lovász and Saks [LS88].

To explain the origins of this conjecture, let us prove a simple lower bound on communication complexity of the equality function $\text{EQ}_n : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$, the function such that $\text{EQ}_n(x, y) = 1$ iff $x = y$ for $x, y \in \{0, 1\}^n$. First, note that any function $F : X \times Y \rightarrow \{0, 1\}$ corresponds to the $|X| \times |Y|$ *communication matrix* M_F such that $(M_F)_{x,y} = F(x, y)$ for $x \in X$ and $y \in Y$ (in this survey we identify F with its communication matrix). Using this connection, Mehlhorn and Schmidt [MS82] proved a simple lower bound on communication complexity. We use the following notation: given a communication problem F , we define $D^{\text{cc}}(F)$ to be the deterministic communication complexity of F ; namely, the minimal number of bits needed by a deterministic

protocol computing F . We define $\text{rank}(F)$ to be the rank of the communication matrix M_F of F , where the rank is computed over reals.

Theorem 1.3 ([MS82]). *Let $F : X \times Y \rightarrow \{0, 1\}$. Then*

$$D^{\text{cc}}(F) \geq \log(\text{rank}(F)).$$

Note that the communication matrix of EQ_n is the $2^n \times 2^n$ identity matrix; hence, $D^{\text{cc}}(\text{EQ}_n) \geq n$. The aforementioned log-rank conjecture conjectures that this bound is almost tight.

Conjecture 1.4 (Log-rank Conjecture [LS88]). *Let $F : X \times Y \rightarrow \{0, 1\}$. Then*

$$D^{\text{cc}}(F) \leq \text{polylog}(\text{rank}(F)).$$

Here and throughout, $\text{polylog}(\cdot)$ is a shorthand for $\log(\cdot)^{O(1)}$.

Despite over 30 years of research, little is known about the validity of the log-rank conjecture. A well-known elementary upper bound is $D^{\text{cc}}(F) \leq \text{rank}(F) + 1$, which is exponentially worse than the conjectured bound. This was improved by Lovett [Lov16] to $D^{\text{cc}}(F) \leq O(\sqrt{\text{rank}(F)} \log \text{rank}(F))$. On the lower bound side, Göös, Pitassi and Watson [GPW18a] constructed a function with $D^{\text{cc}}(F) \geq \Omega(\log^2 \text{rank}(F))$.

1.4 Randomized variants of lifting and log-rank

There are natural randomized variants of lifting and the log-rank conjecture which have also received recent attention. On the lifting side, [GPW17], with follow up work of [CFK⁺19a], showed a query-to-communication lifting theorem from bounded-error randomized decision trees to bounded-error public coin protocols. In the language of standard complexity classes, this is referred to as a ‘BPP lifting theorem’. A lifting theorem for the larger class $\text{ZPP}^{\text{NP}[1]}$ — consisting of zero-error protocols with a single query to an NP-oracle — was given in [Wat19]. As discussed by Watson in [Wat19], communication lower bounds against zero-error protocols with access to *two* NP queries would require proving lower bounds against Arthur-Merlin communication, which is a notorious open problem. Notably, these randomized lifting results are all in the *large gadget* regime, holding, for example, when g is the indexing gadget. No analogous randomized lifting results are known in the small-gadget regime, which we consider to be an important direction for future work.

In formulating a randomized variant of the log-rank conjecture, Lee and Shraibman [LS09] noticed that if we expect linear algebraic measures to characterize deterministic communication complexity, then it is logical to expect matrix-analytic measures to characterize randomized communication complexity. Indeed, it is clear that if F has a short randomized communication protocol, then M_F can be approximated as a sum of matrices with small rank since any randomized communication protocol is a convex combination of deterministic protocols. Let $\text{rank}_\epsilon(F)$ be the minimal number r such that

$$\left\| M_F - \sum_{i=1}^r M_i \right\|_\infty \leq \epsilon$$

for some rank-1 matrices M_1, \dots, M_r . Hence, the previous observation can be written as follows:

$$\log(\text{rank}_\epsilon(F)) \leq R_\epsilon^{\text{cc}}(F),$$

where $R_\epsilon^{\text{cc}}(F)$ denotes randomized communication complexity of F with error bounded by ϵ . Lee and Shraibman conjectured, in analogy with the log-rank conjecture, that the opposite inequality holds too (this conjecture is known as the approximate-log-rank conjecture). Recently, Chattopadhyay, Mande and Sherif [CMS19] found a counterexample to the approximate-log-rank conjecture:

Theorem 1.5 ([CMS19]). *There are functions $F_n : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ such that $R_{1/3}^{\text{cc}}(F_n) = \Omega(\sqrt{n})$ but $\text{rank}_{1/3}(F_n) = O(n^2)$.*

In fact, the functions F_n from Theorem 1.5 are XOR-functions and the idea of the proof is based on the aforementioned connection between XOR-functions and PDTs.

Another natural analogue of the log-rank conjecture is the so-called the log-approximate-nonnegative-rank conjecture, suggested initially by [KMSY19] and then refined by [CMS19]. To formulate the conjecture we need to define approximate-nonnegative-rank. Let $\text{rank}_\epsilon^+(F)$ be the minimal number r such that

$$\left\| M_F - \sum_{i=1}^r M_i \right\|_\infty \leq \epsilon$$

for some rank-1 matrices M_1, \dots, M_r with nonnegative entries.

Conjecture 1.6 (Log-approximate-nonnegative-rank Conjecture [KMSY19, CMS19]). *Let $F : Y \times Y \rightarrow \{0, 1\}$. Then*

$$D^{\text{cc}}(F) \leq \text{polylog}(\max\{\text{rank}_{1/3}^+(F), \text{rank}_{1/3}^+(\neg F)\}),$$

where $\neg F = 1 - F$ is the negation of F .

It is important to notice that the fact that we have a maximum on the right side of inequality is important since in the same paper Chattopadhyay et al. [CMS19] proved the following statement.

Theorem 1.7 ([CMS19]). *Let $F_n : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ be the functions from Theorem 1.5. Then $\text{rank}_{1/3}^+(F_n) = n^{O(\log n)}$.*

Moreover, randomized communication complexity is closed under complement and so any matrix analytic characterization should also be closed under complement.

Organization. The rest of the survey focuses on one-bit gadgets. Section 2 covers XOR-functions and parity decision trees, and Section 3 covers AND-functions and AND decision trees.

2 Parity decision trees

A *parity decision tree* (abbrv. PDT) is an extension of the standard decision tree model, where nodes can query arbitrary parities of the input bits.

Definition 2.1 (Parity decision tree). *A parity decision tree is a binary tree T where each non-leaf node v is labelled by a vector $\gamma_v \in \mathbb{F}_2^n$ and each leaf node ℓ is labelled by a bit $b_\ell \in \{0, 1\}$. Given an input $z \in \mathbb{F}_2^n$, computation on a parity decision tree proceeds by walking from the root to a leaf, where at each step the choice of which child to take in the path is determined by the value of the bit $z_{\gamma_v} = \langle z, \gamma_v \rangle$ (denoting the inner product over \mathbb{F}_2^n): if $z_{\gamma_v} = 0$, we take the left child and if $z_{\gamma_v} = 1$, we take the right child. Computation terminates when this iterative process reaches a leaf ℓ and the outcome of the computation is b_ℓ . We denote by $T(z)$ the output of T on an input z .*

A PDT T computes a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ if for every input $z \in \{0, 1\}^n$, $T(z)$ is equal to $f(z)$. We denote by $D^{\oplus\text{-dt}}(f)$ the minimal depth of a PDT which computes f . We may also be interested in the *size* of the PDT, which is the number of leaves.

Viewing the input space as \mathbb{F}_2^n , PDTs of depth d partition the inputs into affine subspaces of co-dimension at most d on which f is constant. To see this, note that a root-to-leaf path of length

d specifies an inhomogeneous system of d linear equations over \mathbb{F}_2^n , the solutions of which form an affine subspace; moreover, each input z will travel to exactly one leaf over the course of computation and hence these subspaces partition \mathbb{F}_2^n . Such a viewpoint stands in contrast to standard decision trees, which partition the input space into *subcubes* (a special case of affine subspaces). Additionally, this viewpoint provides us with a simple strategy of proving lower bounds on PDTs computing a function f : if we know that f is non-constant on any affine subspace of co-dimension d , then f requires PDTs of depth greater than d .

This section will examine the PDT model in detail. First, we will look at the relationship between PDT complexity (and other related parity query measures) and the Fourier transform of f . Second, we will look at how PDTs are related to the communication complexity of XOR functions and lead to a natural special case of the log-rank conjecture. Finally, we will see how PDTs have been used in other contexts, such as property testing and proof complexity.

2.1 Relationships to the Fourier transform and other query measures

2.1.1 Fourier sparsity and spectral norm

The Fourier transform of a function $f : \mathbb{F}_2^n \rightarrow \mathbb{R}$ is defined by

$$f(z) = \sum_{\gamma \in \mathbb{F}_2^n} \widehat{f}(\gamma) (-1)^{\langle z, \gamma \rangle}$$

where the coefficients $\widehat{f}(\gamma) = \mathbf{E}_x[f(x) \cdot (-1)^{\langle z, \gamma \rangle}]$ are referred to as the *Fourier coefficients*. There are many interesting quantities associated with f 's Fourier transform and we will focus on two for the time being: the sparsity and the spectral norm. The *Fourier sparsity* (which we sometimes simply refer to as *sparsity* in this section) of f is defined as $\|\widehat{f}\|_0 = |\{\gamma \in \mathbb{F}_2^n : \widehat{f}(\gamma) \neq 0\}|$, namely the number of nonzero Fourier coefficients of f . The *spectral norm* of f is defined as $\|\widehat{f}\|_1 = \sum_{\gamma \in \mathbb{F}_2^n} |\widehat{f}(\gamma)|$.

Intuitively, the Fourier coefficient of γ in f 's Fourier representation measures the correlation between f and the linear function $z \mapsto \langle z, \gamma \rangle$. It might therefore seem unsurprising that there are relationships between the Fourier transform of f and its PDT complexity. For example, a reasonable heuristic for constructing PDTs could be to iteratively query the parity with the highest Fourier coefficient. Perhaps the simplest formal relationship between PDT complexity and the complexity measures on the Fourier transform introduced above is the following:

Claim 2.2. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be computed by a PDT of depth d . Then $\|\widehat{f}\|_0 \leq 4^d$ and $\|\widehat{f}\|_1 \leq 2^d$.*

Proof. Let T be a PDT of depth d computing f . Let $L = \{\ell_1, \dots, \ell_m\}$ be the leaves of T . For each leaf ℓ let $I_\ell(z)$ be the indicator for an input z reaching the leaf ℓ in the computation of $T(z)$, and let b_ℓ be the output bit of ℓ . Then

$$T(z) = \sum_{\ell \in L} I_\ell(z) b_\ell.$$

The claim follows as $|L| \leq 2^d$, each I_ℓ has Fourier sparsity 2^d and spectral norm 1. To see why, fix a leaf ℓ of depth $e \leq d$, let $\gamma_1, \dots, \gamma_e \in \mathbb{F}_2^n$ be the parities queried in the path to ℓ , and let $a_1, \dots, a_e \in \{0, 1\}$ be the labels of the edges followed. Then

$$I_\ell(z) = \prod_{i \in [e]} \mathbf{1}[\langle z, \gamma_i \rangle = a_i] = \prod_{i \in [e]} \frac{1 + (-1)^{\langle z, \gamma_i \rangle + a_i}}{2},$$

where $\mathbf{1}[\langle z, \gamma_i \rangle = a_i]$ is 1 if $\langle z, \gamma_i \rangle = a_i$ and 0 otherwise. A direct computation gives $\|\widehat{I}_\ell\|_0 \leq 2^e$ and $\|\widehat{I}_\ell\|_1 \leq 1$. \square

Given this upper bound, one might ask: does small sparsity or small spectral norm imply the existence of a small PDT? We first formulate this for sparsity:

Question 2.3. *Is $D^{\oplus-dt}(f) \leq \text{polylog}(\|\widehat{f}\|_0)$?*

The best known lower bound is that there are functions f with $D^{\oplus-dt}(f) \geq \log(\|\widehat{f}\|_0)^c$ for $c = \log_3 6 \approx 1.63$ due to O'Donnell et al. [OWZ⁺14]. This question will be discussed in more detail in Section 2.2 as it turns out to be related to the communication complexity of XOR functions (in fact, it is equivalent to the log-rank conjecture for XOR functions). For the spectral norm, we have the analogous question:

Question 2.4. *Is $D^{\oplus-dt}(f) \leq \text{polylog}(\|\widehat{f}\|_1)$?*

In this case, a strong negative result was given by Chattopadhyay et al. [CMS19]:

Theorem 2.5 ([CMS19]). *There is a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ so that $D^{\oplus-dt}(f) = \Omega(\sqrt{n})$ and $\|\widehat{f}\|_1 = O(\sqrt{n})$.*

Despite this barrier, low spectral norm can still guarantee some type of simplicity in the PDT model but there will be some dependence on n . As alluded to previously, a reasonable-sounding heuristic for constructing a PDT is to repeatedly query the parity with the largest corresponding Fourier coefficient. The problem with this heuristic is that f might not have any large Fourier coefficients (e.g. the inner product function on $2n$ bits), in which case one can't do much better than randomly guessing a parity to query. However, when the spectral norm is small, it turns out that a variant of this heuristic actually works.

The idea is to select *two* parities, which we'll think of as vectors $\alpha, \beta \in \mathbb{F}_2^n$, whose Fourier coefficients $\widehat{f}(\alpha)$ and $\widehat{f}(\beta)$ are rather large. These can be easily show to exist under the assumption that f has small spectral norm. Then our PDT will repeatedly query the value of $\langle \alpha + \beta, z \rangle$. It can be shown that, regardless of the value of $\langle \alpha + \beta, z \rangle$, making such a query will reduce the spectral norm of the restricted function significantly.

Theorem 2.6 ([TWXZ13, STIV17]). *Suppose $\|\widehat{f}\|_1 = S$. Then f can be computed by a PDT of size $2^{S^2} n^{2S}$.*

A question similar in spirit to the existence of short PDTs from low spectral norm is whether there exists an 'efficient' decomposition of f into subspaces. Specifically, how many affine subspaces $V_1, \dots, V_m \subset \mathbb{F}_2^n$ are needed to ensure that f can be decomposed as $f(z) = \sum_{i \in [m]} \pm \mathbf{1}[z \in V_i]$? Note that we are not restricting the co-dimension of these subspaces.

By writing a PDT as a sum of leaf indicators (similar to the proof of Claim 2.2), it is easy to see that if f has a PDT of depth d (respectively, size t), then it suffices to take $m \leq 2^d$ (respectively, $m \leq t$). In particular, the above theorem gives us an upper bound of $2^{S^2} n^{2S}$ when $S = \|\widehat{f}\|_1$. This has the undesirable characteristic of depending on n . Sanders [San19] (see also Green-Sanders [GS08]) gives an upper bound with no dependence on n while maintaining the exponential dependence on S :

Theorem 2.7 ([San19]). *Suppose $\|\widehat{f}\|_1 = S$. Then there are $m = 2^{S^{3+o(1)}}$ affine subspaces $V_1, \dots, V_m \subset \mathbb{F}_2^n$ and coefficients $c_1, \dots, c_m \in \{-1, 1\}$ so that*

$$f(z) = \sum_{i \in [m]} c_i \mathbf{1}[z \in V_i].$$

Additional relationships between the Fourier transform and PDT complexity have also been discovered. For example, a result of O’Donnell and Servedio [OS07] states that the sum of the *linear* Fourier coefficients are upper bounded by the square root of the (standard) decision tree depth.

Theorem 2.8 ([OS07]). *If f is computable by a depth- d decision tree then*

$$\sum_{i \in [n]} \widehat{f}(i) \leq O(\sqrt{d}).$$

Blais, Tan and Wan in [BTW15] generalized this result to PDTs as follows:

Theorem 2.9 ([BTW15]). *If f is computable by a depth- d PDT then*

$$\sum_{i \in [n]} \widehat{f}(i) \leq O(\sqrt{d}) \cdot \mathbf{Var}[f].$$

Having described some relationships between deterministic PDT complexity and various measures of complexity in the Fourier basis, we now introduce *randomized* PDTs about which less is known.

2.1.2 Randomized parity decision trees

Let $\mathcal{P}_{n,d}$ be the set of PDTs of depth d computing over \mathbb{F}_2^n . A randomized PDT \mathcal{T} of depth d is a distribution over trees $T \in \mathcal{P}_{n,d}$. We say \mathcal{T} *computes* f if for every z , $\Pr_{T \sim \mathcal{T}}[T(z) = f(z)] \geq 2/3$.

Unlike the result of Nisan and Szegedy [NS94] establishing the equivalence of randomized and deterministic decision trees (up to polynomial factors), randomized PDTs are exponentially stronger than deterministic PDTs. To see this, consider the conjunction of n bits; that is, the function $\text{AND}_n : \{0, 1\}^n \rightarrow \{0, 1\}$ such that $\text{AND}_n(x_1, \dots, x_n) = \bigwedge_{i=1}^n x_i$. It can be easily verified that a deterministic PDT requires depth n to compute the function. However, there is a randomized PDT of constant depth computing AND_n : sampling independent, uniformly random parities $\gamma_1, \dots, \gamma_k \in \mathbb{F}_2^n$ to query and checking that $\langle z, \gamma_i \rangle = \langle \mathbf{1}, \gamma_i \rangle$ (where $\mathbf{1}$ is the all 1’s vector) for each $i \in [k]$. The probability of a false negative is 0 and the probability of a false positive is 2^{-k} , so a constant number of queries k suffices to compute AND.

A useful perspective which provides a more conceptual justification to the randomized-deterministic separation is via the Hadamard code. Recall the Hadamard encoding of $z \in \mathbb{F}_2^n$ is the 2^n -bit string where the i -th bit is equal to $\langle z, \gamma_i \rangle$, where γ_i is the i -th vector in \mathbb{F}_2^n under some ordering. Then we can see the PDT as having regular query access to the Hadamard encoding of z . Since the Hadamard code has distance $n/2$, random queries to the codeword quickly demarcate inequivalent strings.

Unsurprisingly, randomized PDTs have their own set of limitations. For example, one can show that the inner product function on $2n$ bits requires $n/2$ queries. This can be seen as an example of a more general family of hard functions called *affine extractors*. An affine extractor for co-dimension d , informally, is a function $g : \mathbb{F}_2^n \rightarrow \{0, 1\}$ so that for every affine subspace V of co-dimension d , $\mathbf{E}_{x \sim V}[g(x)] \approx 1/2$. It is a simple observation that such functions require randomized PDT of depth $\Omega(d)$.

2.1.3 Parity kill number

A relaxation of affine extractors called *affine dispersers* are also relevant to PDT complexity. Affine dispersers for dimension d are functions $g : \mathbb{F}_2^n \rightarrow \{0, 1\}$ which are *non-constant* on every affine subspace V of dimension d . This notion of affine disperser suggests another complexity measure, which was called the *parity kill number* in [OWZ⁺14], and we will keep this terminology here. The parity kill number of a function f is the co-dimension of the *largest* affine subspace on which f is constant. Note that affine dispersers of dimension d have parity kill number $n - d$.

The parity kill number was studied in [OWZ⁺14], giving a lower bound on the parity kill number of the composed function $f^{\circ k}$.⁹ The parity kill number is also connected to the construction of short PDTs for f . In particular, [TWXZ13] observe that if f has small sparsity and small parity kill number then it has a small PDT:

Theorem 2.10 ([TWXZ13]). *Let $T(\cdot)$ be a function such that, for any boolean function f , its parity kill number is at most $T(\|\widehat{f}\|_0)$. Then for any boolean function f , its PDT depth is at most $T(\|\widehat{f}\|_0) \cdot \log \|\widehat{f}\|_0$.*

In particular, it suggests the following natural question, connecting the parity kill number to Fourier sparsity as means of connecting it with PDT depth.

Question 2.11. *Is the parity kill number of any boolean function f at most $\text{polylog}(\|\widehat{f}\|_0)$?*

The analogous question, where Fourier sparsity is replaced by spectral norm, was raised in [TWXZ13] and disproved in [CMS19]:

Theorem 2.12. *There is a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ so that f has parity kill number $\Omega(\sqrt{n})$ and $\|\widehat{f}\|_1 = O(\sqrt{n})$.*

We view it as an interesting question to understand the relationship between randomized PDT complexity and the parity kill number:

Question 2.13. *Suppose a boolean function f can be computed by a randomized PDT of depth d . Does this imply that the parity kill number of f is at most $\text{poly}(d)$? namely, that f is constant on an affine subspace of co-dimension $\text{poly}(d)$?*

Generally, the relationship between *approximate global structure* of f (such as randomized PDTs) and *exact local structure* of f (such as a large monochromatic subspace) is poorly understood and worthy of more investigation.

2.2 Relationship to communication complexity of XOR-functions

Decision trees have historically been a useful tool to understand more complicated models of computation. One recent success story is *query-to-communication lifting theorems* from communication complexity, which tightly characterize the communication complexity of particular structured functions in terms of the decision tree complexity of a related function. Might PDTs have a similar utility? A simple connection between PDT complexity and communication complexity of XOR functions is the following claim, whose proof we sketched in the introduction.

Claim 2.14. *Suppose that f can be computed by a PDT of depth d . Then the function f_{\oplus} has deterministic communication complexity at most $2d$.*

⁹We do not define composed functions as it is tangential to our discussion. See [OWZ⁺14] for the definition.

The converse statement is also true, which establishes a (qualitatively) tight characterization of the communication complexity of XOR functions in terms of PDT complexity. The following theorem was proved by Hatami, Hosseini and Lovett [HHL18].

Theorem 2.15 (Lifting for XOR functions [HHL18]). *Suppose that f_{\oplus} has deterministic communication complexity d . Then f can be computed by a PDT of depth $O(d^6)$.*

It is worth understanding the hypothesis of Theorem 2.15 in more detail. Recall that communication protocols for any two-party function F imply the existence of a partitioning of the associated communication matrix M_F into a small number of monochromatic rectangles. In our context, a rectangle $A \times B$ corresponds directly to the *sumset* $A + B = \{a + b : a \in A, b \in B\}$ over \mathbb{F}_2^n (since $F(x, y) = f(x + y)$) and so \mathbb{F}_2^n can be partitioned into monochromatic sumsets.

Another property of f that follows from the existence of short communication protocols for F is that f has small *Fourier sparsity*.

Claim 2.16. *Suppose f_{\oplus} has deterministic communication complexity d . Then $\|\widehat{f}\|_0 \leq 2^d$.*

This claim can be seen in two steps. First, we use the generic observation that short protocols for F implies that the communication matrix M_F has correspondingly small *rank*. Next, we connect rank to sparsity as follows:

Claim 2.17. *For any $f : \{0, 1\}^n \rightarrow \mathbb{R}$, $\text{rank}(M_{f_{\oplus}}) = \|\widehat{f}\|_0$.*

Proof. Let $M = M_{f_{\oplus}}$ where we identify its rows and columns with \mathbb{F}_2^n . It satisfies $M_{x,y} = f(x \oplus y)$. We first show how to express M in terms of the Fourier decomposition of f .

The Fourier decomposition of f is given by $f(z) = \sum_{\gamma \in \mathbb{F}_2^n} \widehat{f}(\gamma) (-1)^{\langle z, \gamma \rangle}$. For each $\gamma \in \mathbb{F}_2^n$ define the vector $v_{\gamma} \in \mathbb{R}^{2^n}$ given by $(v_{\gamma})_z = (-1)^{\langle z, \gamma \rangle}$ (here we identify the coordinates of v_{γ} with \mathbb{F}_2^n). We have the identity

$$M = \sum_{\gamma \in \mathbb{F}_2^n} \widehat{f}(\gamma) v_{\gamma}^T v_{\gamma},$$

which holds since

$$M_{x,y} = f(x \oplus y) = \sum_{\gamma \in \mathbb{F}_2^n} \widehat{f}(\gamma) (-1)^{\langle x \oplus y, \gamma \rangle} = \sum_{\gamma \in \mathbb{F}_2^n} \widehat{f}(\gamma) (-1)^{\langle x, \gamma \rangle} (-1)^{\langle y, \gamma \rangle} = \sum_{\gamma \in \mathbb{F}_2^n} \widehat{f}(\gamma) (v_{\gamma})_x (v_{\gamma})_y.$$

As each matrix $v_{\gamma}^T v_{\gamma}$ is a rank-one matrix, we immediately get that the rank of M is at most the number of nonzero Fourier coefficients of f :

$$\text{rank}(M) \leq \|\widehat{f}\|_0.$$

Next, observe that the vectors $\{v_{\gamma} : \gamma \in \mathbb{F}_2^n\}$ are linearly independent. Simple linear algebra then gives that in fact

$$\text{rank}(M) = \|\widehat{f}\|_0.$$

□

Going back to the question of lifting XOR-functions to PDTs, we already established two useful facts: (i) an efficient communication protocol for F implies sparsity in the Fourier basis; and (ii) sparsity in the Fourier basis combined with a small parity kill number implies the existence of a small PDT. This leads into a natural question — can we use the existence of a large monochromatic

sumset to show that the parity kill number of f is small, thereby proving a lifting result for PDTs? It turns out that in general, large sumsets need not contain large subspaces, which is an obstacle to using this approach. The authors of [HHL18] get around this by using more directly the assumption that F has an efficient protocol in order to prove the lifting theorem for XOR functions. For further details see the paper [HHL18].

In addition to lifting, one might ask for additional applications of the structure of XOR functions. For example, is the log-rank conjecture true for XOR functions? By the previous claim which characterizes the rank as the Fourier sparsity, this question is equivalent to the following:

Question 2.18. *Is it true that for any boolean function f , the deterministic communication complexity of f_{\oplus} is at most poly-logarithmic in the Fourier sparsity of f ? Namely, does $D^{\text{cc}}(f_{\oplus}) \leq \text{polylog}(\|f\|_0)$?*

Since we can upper bound the communication complexity in terms of the PDT complexity, it would also be sufficient to answer Question 2.3 in the affirmative (and this is indeed equivalent, by the lifting theorem for XOR functions). Various special cases of this question are known to be the case: for example, when f has constant \mathbb{F}_2 -degree [TWXZ13] or is monotone [MO09].

Analogous questions may be formulated in the randomized setting. To begin with:

Question 2.19. *Suppose f_{\oplus} has randomized communication complexity d . Does f have randomized PDTs of depth $\text{poly}(d)$?*

This question remains open. As an intermediate step, it is worthwhile to investigate whether lifting theorems hold for other notions of global structure. For example, we can ask this question about the *smooth rectangle bound* [Kla10] which is conjectured, but not known, to be equivalent to randomized communication complexity:

Question 2.20. *Suppose, for f_{\oplus} , there exists a function $G(x, y) = \sum \alpha_R R(x, y)$ with $\alpha_R \geq 0$ so that*

1. *Each R is the indicator function of a rectangle.*
2. *$|f_{\oplus}(x, y) - G(x, y)| \leq \epsilon$ for every (x, y)*
3. *$M = \sum_R \alpha_R$.*

Then is there a function $g(z) = \sum \beta_V V(z)$ where V are indicators for affine subspaces, $\beta_V \geq 0$ such that $|f(z) - g(z)| \leq \epsilon$ and $\sum_V \beta_V = M^{O(1)}$?

Define for any $\epsilon > 0$, the ϵ -approximate sparsity of a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ as the smallest s so that there exists a function g with $\|\widehat{g}\| = s$ and $|g(x) - f(x)| \leq \epsilon$ for every $x \in \{0, 1\}^n$. In analogy with Question 2.3, we can ask the following randomized variant:

Question 2.21. *Suppose f has $1/3$ -approximate sparsity r . Does f have a constant-error randomized PDT of depth $\text{polylog}(r)$?*

The answer to this question turns out to be negative, as shown by Chattopadhyay et al. [CMS19]:

Theorem 2.22 ([CMS19]). *There is a boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ so that any constant-error randomized PDT has depth $\Omega(\sqrt{n})$ and yet f has $1/3$ -approximate sparsity $O(n^2)$.*

Such a separation also yields a separation for a randomized variant of the log-rank conjecture. In particular, for any $\epsilon > 0$, the ϵ -approximate rank of a matrix $M \in \{0, 1\}^{n \times n}$ is the smallest s so that there exist rank-1 matrices $M_1, \dots, M_s \in \mathbb{R}^{n \times n}$ for which $|M_{x,y} - (\sum_{i \in [s]} M_i)_{x,y}| \leq \epsilon$ for any entry (x, y) .

Theorem 2.23 ([CMS19]). *There is a boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ so that any constant-error randomized communication protocol for f_{\oplus} has depth $\Omega(\sqrt{n})$ and yet $M_{f_{\oplus}}$ has $1/3$ -approximate rank $O(n^2)$.*

This achieved simply by observing that if f has small approximate sparsity then f_{\oplus} has small approximate rank.

2.3 Relationships to other areas of theory

The study of PDTs has also appeared in application areas other than communication complexity. Here, we briefly discuss relationships to property testing, circuit complexity and proof complexity.

2.3.1 Testing linear and quadratic functions

Property testing is the study of determining, in sub-linear time, whether an input object satisfies some property \mathcal{P} or is ‘far’ from any object which satisfies the property \mathcal{P} . See, for example, [Gol17] for an introduction to property testing. A randomized algorithm which performs this task is called a *tester* T . When T is testing a property of the function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ we give it black-box access to f and generally ask for two-sided error guarantees.

Linear functions $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ are determined by their evaluation over some basis of \mathbb{F}_2^n . In particular, $f(x) = \sum_{i : x_i=1} f(e_i)$. A randomized tester T for linear functions having some property \mathcal{P} using query access to f can then be seen to be equivalent to a randomized PDT for a function $g_{\mathcal{P}}$ applied to strings $(f(e_1), f(e_2), \dots, f(e_n))$.

This perspective can be applied to obtain property testing for k -linearity — testing whether your function is of the form $z \mapsto \langle z, \gamma \rangle$ where γ has k non-zero entries — and can be generalized to quadratic forms in order to prove lower bounds for testing affine isomorphism to the inner product function. See [BCK14] for details.

2.3.2 DNFs of parities

A frontier open problem in circuit complexity is the construction of strong correlation bounds and pseudorandom generators for the circuit class $AC^0[\oplus]$ consisting of bounded-depth, unbounded fan-in circuits of polynomial size with \wedge, \vee, \oplus and \neg gates. An intermediate model is $AC^0 \circ \oplus$, consisting of constant-depth \wedge, \vee, \neg circuits with one layer of parity gates at the bottom. A concrete problem which is so far unsolved is the following:

Question 2.24. *Does $IP_n(x_1, \dots, x_{2n}) = \sum_{i \in [n]} x_i x_{i+n} \pmod 2$ require depth- d $AC^0 \circ \oplus$ circuits of size $\exp(\Omega(n^{1/d}))$?*

A low-depth variant of this model is directly related to PDTs. Specifically, a DNF of parities is a function of the form $f(x) = \vee_i C_i(x)$ where $C_i(x)$ is the indicator function of an affine subspace V_i . We say the *width* of C_i is simply the co-dimension of V_i . We say that the DNF of parities has width k if largest co-dimension of a subspace V_i is k . By writing a PDT as a union of the accepting affine subspaces, we can see that a PDT of depth d can be written as a DNF of parities of width d .

Given the difficulty in proving strong correlation bounds for $AC^0[\oplus]$, it seems natural to investigate to what extent our techniques for AC^0 can extend to $AC^0 \circ \oplus$. In particular, do common techniques used in the study of DNFs, such as the switching lemma or sparsification, extend to DNF of parities?

2.3.3 Average-case (unrestricted) circuit lower bounds

Another important problem in circuit complexity is proving lower bounds on (unrestricted) circuit complexity of explicit functions. The best known lower bound bound was proven by [LY21]. Below, we denote by a B_2 -circuit a circuit over the full binary basis.

Theorem 2.25. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be an affine extractor¹⁰ for dimension d . Then any B_2 -circuit for f has size at least $(3.1 - o(1))n$.*

However, if we only require the circuit to be correlated with the function, the best bound was proven by Chen and Kabanets [CK16] by using connections between circuit complexity and PDTs.

Theorem 2.26. *Any boolean function computed by a B_2 -circuit of size $s < 2.5n$ is computable by a PDT of size $2^{n - \Omega((2.5n - s)^2/n)}$.*

This theorem combined with the observation that affine extractors for dimension d require large PDTs implies the following lower bound.

Theorem 2.27. *Let $\delta > 0$. There are explicit functions $f_n : \{0, 1\}^n \rightarrow \{0, 1\}$ such that any B_2 -circuit of size $2.5n - \delta n$, has correlation with f_n at most $2^{-\Omega(n)}$.*

Question 2.28. *Is it possible to construct explicit functions f_n such that any B_2 -circuit of size $3n - o(n)$ has correlation with f_n at most $2^{-\Omega(n)}$?*

2.3.4 Resolution with linear constraints

A major open question in proof complexity is proving lower bounds on Frege proof systems (this is the name used in proof complexity for the standard text-book Hilbert-style proof system).

Question 2.29. *Is there an unsatisfiable CNF formula ϕ such that any Frege refutation of ϕ has size exponential in the size of ϕ ?*

Currently the best known result is for so-called AC^0 -Frege.

Theorem 2.30 ([BIK⁺92, PBI93, Raz95]). *For any d , there is a family of unsatisfiable formulas ϕ_n and $\delta < (1/5)^d$ such that*

- *the size of ϕ_n is polynomial in n and*
- *any Frege refutation of ϕ_n has size 2^{n^δ} , provided that each line of the refutation is a AC^0 circuit of depth d .*

However, like in circuit complexity the situation with $AC^0[\oplus]$ is not clear.

Question 2.31. *Let d be a constant. Is there a family of unsatisfiable CNF formulas ϕ_n such that any Frege refutation of ϕ_n has size exponential in the size of ϕ_n , provided that each line of the refutation is a $AC^0[\oplus]$ circuit of depth d ?*

¹⁰In fact this theorem holds even for affine dispersers.

Hence, one may try to consider subsystems of $AC^0[\oplus]$ -Frege to find the specific obstacles for currently known methods. One of the most popular such proof systems is $\text{Res}(\text{lin}_{\mathbb{F}_2})$. This is the resolution based proof system that operates with disjunctions of linear equations over \mathbb{F}_2 .

Definition 2.32. *We say that a formula C semantically implies a formula D if any assignment that satisfies C also satisfies D . The resolution principle says that any assignment that satisfies both $C \vee X$ and $D \vee \neg X$ also satisfies $C \vee D$. The clause $C \vee D$ is said to be a resolvent of $C \vee X$ and $D \vee \neg X$ derived by resolving on X .*

We say that C is a linear clause if C is a disjunction of linear equations over \mathbb{F}_2 . Note that any clause can be represented as a linear clause.

These definitions lead us to a definition of a resolution over linear combinations proof system ($\text{Res}(\text{lin}_{\mathbb{F}_2})$). A $\text{Res}(\text{lin}_{\mathbb{F}_2})$ refutation of a CNF ϕ is a sequence of linear clauses in which each linear clause is either a clause of ϕ , or is a resolvent of two previous linear clauses, or semantically implied by a previous linear clause; and the last disjunction in the sequence is empty; We say that this refutation is tree-like if each clause is used exactly once.

It is easy to see that this proof system is a fragment of depth-2 $AC^0[\oplus]$ -Frege since each line is a conjunction of XORs. However, even for this ultimately weak system the lower bounds are only on the tree-like version of the proof system [IS20].

Question 2.33. *Is there a family of unsatisfiable CNF formulas ϕ_n such that any $\text{Res}(\text{lin}_{\mathbb{F}_2})$ refutation of ϕ_n has size exponential in the size of ϕ_n ?*

The reason behind proving a lower bound on the tree-like $\text{Res}(\text{lin}_{\mathbb{F}_2})$ refutations is the following connection between these refutations and PDTs.

Theorem 2.34. *Let $\phi = \bigwedge_{i=1}^m C_i$ be an unsatisfiable CNF on n variables, and let $\text{Search}_\phi \subseteq \{0, 1\}^n \times [m]$ be a search problem such that*

$$(x, i) \in \text{Search}_\phi \iff C_i(x) = 0.$$

In other words, Search_ϕ is the problem of finding a falsified clause. If there is a $\text{Res}(\text{lin}_{\mathbb{F}_2})$ refutation of ϕ of size S , then there is a PDT for Search_ϕ of depth $\log S$.

Therefore, to prove a lower bound on $\text{Res}(\text{lin}_{\mathbb{F}_2})$ it is enough to prove a lower bound on PDT complexity. Similar connections between decision tree and communication complexity of Search_ϕ and proof complexity of ϕ is known for many other pairs of models of computations and proof systems [GP18, BPS07, Kno17, BPR97, IPU94]; probably the most explicit example of such connection is the stabbing planes proof system [BFI⁺18]: proofs in this proof system are decision trees that query linear inequalities (over the reals).

3 AND decision trees

In this section, we study the communication complexity of AND-functions and their corresponding query model: *AND decision trees* (abbrv. ADTs).

Similar to PDTs, an ADT extends the standard decision tree model, where every node could query the AND of arbitrary subset of the input bits.

Definition 3.1 (AND decision tree). *An AND decision tree is a binary tree T where each non-leaf node v is labelled by a subset $S_v \subseteq [n]$, its two branches are labelled by 0, 1 respectively, and each leaf node ℓ is labelled by a bit $b_\ell \in \{0, 1\}$. Given an input $z \in \{0, 1\}^n$, computation on a AND decision tree proceeds by walking from the root to a leaf, where at each step we take the child labelled by $z_{S_v} = \prod_{i \in S_v} z_i$. Computation terminates when this iterative process reaches a leaf ℓ and the outcome of the computation is b_ℓ . We denote by $T(z)$ the output of T on an input z .*

We say an ADT T computes $f : \{0, 1\}^n \rightarrow \{0, 1\}$ if for every input $z \in \{0, 1\}^n$ the output $T(z)$ is equal to $f(z)$. We denote by $D^{\wedge-dt}(f)$ the minimal depth of an ADT which computes f .

In the previous case of XOR functions, we were working with the Fourier basis for functions over $\{0, 1\}^n$. In this case, we instead work with the basis of all multilinear monomials $\{x^S : S \subseteq [n]\}$, where $x^S = \prod_{i \in S} x_i$. One can write an arbitrary function $f : \{0, 1\}^n \rightarrow \mathbb{R}$ uniquely as a multilinear polynomial

$$f(x) = \sum_{S \subseteq [n]} a_S x^S,$$

where $a_S \in \mathbb{R}$. The number of non-zero coefficients is called the *Möbius sparsity* (which we will usually refer to as just ‘sparsity’ in this section) of f , denoted by $\text{spar}(f) = |\{S : a_S \neq 0\}|$. Similar to PDT and Fourier sparsity, $D^{\wedge-DT}(f)$ is lower bounded in terms of $\text{spar}(f)$.

Claim 3.2. *Let $f : \{0, 1\}^n \rightarrow \mathbb{R}$ be computed by an ADT of depth d . Then $\text{spar}(f) \leq 3^d$.*

Proof. We prove the claim by induction on d . It clearly holds for $d = 0$. For $d \geq 1$, let $S \subseteq [n]$ denote the subset of variables whose AND is queried at the root, and let f_0, f_1 denote the functions computed by its two subtrees. Then we have

$$f(x) = x^S f_1 + (1 - x^S) f_0.$$

By induction, $\text{spar}(f_0), \text{spar}(f_1) \leq 3^{d-1}$ as they are computed by ADTs of depth at most $d - 1$. Hence $\text{spar}(f) \leq 3^d$. \square

Another important fact points out the relationship between the sparsity of a boolean function and the rank of its corresponding AND-function, analogous to Claim 2.17 which connected the rank of XOR functions to the Fourier sparsity.

Claim 3.3. *For any $f : \{0, 1\}^n \rightarrow \mathbb{R}$, $\text{rank}(M_{f_\wedge}) = \text{spar}(f)$.*

The proof of Claim 3.3 is analogous to the proof of Claim 2.17 and we omit it.

3.1 Relationship to communication complexity of AND-functions

Similarly to PDTs and XOR-functions, by simulating the computation of an ADT, one can give an upper bound on the deterministic communication complexity of its corresponding AND-function.

Claim 3.4. *Suppose that f has an ADT of depth d . Then the function f_\wedge has deterministic communication complexity at most $2d$.*

A natural question to be asked is whether the opposite direction is true. Knop, Lovett, McGuire, and Yuan [KLMY21] recently gave a positive answer up to a $\log n$ factor. Before reviewing this result, let us introduce a measure called *monotone block sensitivity*, which builds a bridge between ADT complexity and communication complexity.

Recall that the block sensitivity of a boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ at input x , denoted $bs(f, x)$, is the maximum number k of pairwise disjoint blocks $B_1, \dots, B_k \subseteq [n]$ so that for all $i \in [k]$, $f(x) \neq f(x^{\oplus B_i})$, where $x^{\oplus B_i} = x \oplus \mathbf{1}_{B_i}$ is the string obtained from x by flipping the value of every coordinate in B_i . The block sensitivity of f is defined as $bs(f) = \max_x bs(f, x)$.

Monotone block sensitivity is a variant of block sensitivity, obtained by restricting our modification of the input x towards higher-weight strings. Specifically, for a subset $B \subseteq [n]$ of coordinates, let $x^{\uparrow B} = x \vee \mathbf{1}_B$ be the string obtained from x by flipping every 0 occurring in B into a 1 and leaving every 1 unmodified. The monotone block sensitivity of f at x , denoted $mbs(f, x)$, is the maximum number of k of pairwise disjoint blocks $B_1, \dots, B_k \subseteq [n]$ so that for each $i \in [k]$, $f(x) \neq f(x^{\uparrow B_i})$. The monotone block sensitivity of f is $mbs(f) = \max_x mbs(f, x)$. To give two examples illustrating this definition, the monotone block sensitivity of n -bit AND is 1, where the monotone block sensitivity of n -bit OR is n (obtained at $x = 0^n$).

We next connect the monotone block sensitivity of f to the communication complexity of its AND-function f_\wedge . Suppose that $mbs(f, z) = k$ for some $z \in \{0, 1\}^n$. We will show that f_\wedge embeds as a sub-function Unique Disjointness on k bits (denoted $UDISJ_k$), a function which is known to require $\Omega(k)$ communication complexity in a variety of communication models (including deterministic and randomized communication).

First, let us define the Unique Disjointness function $UDISJ_k$: it is a partial two-party function whose inputs are $a, b \subseteq [k]$. Its value is 0 if they are disjoint, 1 if the inputs share a single element in common, and is not defined if they share two or more elements in common.

In order to see the connection, assume that $mbs(f, z) = k$ is witnessed by pairwise disjoint blocks $B_1, \dots, B_k \subseteq [n]$ such that $f(z) \neq f(z^{\uparrow B_i})$. Denote $w_i = \mathbf{1}_{B_i}$. Given $a, b \subseteq [k]$ define the following inputs to f :

$$x(a) = z \vee \bigvee_{i \in a} w_i, \quad y(b) = z \vee \bigvee_{i \in b} w_i.$$

Note that we have

$$x(a) \wedge y(b) = \begin{cases} z & \text{if } a \cap b = \emptyset \\ z \vee w_i & \text{if } a \cap b = \{i\}. \end{cases}$$

Thus, any protocol computing f can also compute $UDISJ_k$. It is well known that $UDISJ_k$ requires $\Omega(k)$ bits of communication under deterministic, non-deterministic, as well as randomized settings [GPW18b, Raz92b]. As a result, we get the following corollary.

Corollary 3.5. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Assume that the deterministic or randomized communication complexity of f_\wedge is c . Then $mbs(f) = O(c)$.*

Monotone block sensitivity is also at most poly-logarithmic in the sparsity of f .

Lemma 3.6. *For any $f : \{0, 1\}^n \rightarrow \{0, 1\}$, $mbs(f) = O(\log^2(\text{spar}(f)))$*

Some of the ideas underlying these definitions can be traced back to an early paper of Nisan and Wigderson on the log-rank conjecture [NW95]. There, they consider a function f with sparsity $2^{n^{\log_3 2}} \approx 2^{n^{0.63}}$ and monotone block sensitivity n . Applying Corollary 3.5 to $F = f \circ \wedge$, we see that $\log \text{rank}(M_F) = \log \text{spar}(f) = O(n^{0.63})$ and yet $mbs(f) = \Omega(n)$. As Nisan and Wigderson observed, this implies a lower bound for the log-rank conjecture. This example also demonstrates that the quadratic factor in Lemma 3.6 cannot be improved beyond a factor $1/(\log_3 2) \approx 1.59$.

One celebrated result in [NS94] is that block sensitivity and decision tree complexity are polynomially related. It is natural to ask whether it is true for monotone block sensitivity and ADT complexity. The following example refutes this.

Example 3.7. The function $\text{AND}_n \circ \text{OR}_2$ is defined as $f(x_1, \dots, x_{2n}) = \bigwedge_{1 \leq i \leq n} (x_{2i-1} \vee x_{2i})$. It is easy to verify that $\text{mbs}(f) = 2$, but $D^{\wedge\text{-dt}}(f) = \Omega(\log \text{spar}(f)) = \Omega(n)$.

Nevertheless, ADT complexity can be bounded in terms of monotone block sensitivity and sparsity. The proof uses *zero decision tree complexity* (abbrv. 0DT complexity) as a medium.

Definition 3.8 (Zero Decision Tree Complexity). The zero decision tree complexity of f , denote by $D^{0\text{-dt}}(f)$, is defined as the minimal zero-depth over all standard decision trees that compute f , where the zero-depth of a decision tree T is defined as the maximal number of 0-branches over all paths from the root to leaves in T .

It turns out that zero decision tree complexity is bounded by a polynomial in the monotone block sensitivity and sparsity of f .

Lemma 3.9 ([KLMY21]). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Then $D^{0\text{-dt}}(f) = O(\text{mbs}^2(f) \cdot \log \text{spar}(f))$.

Combining this lemma with Lemma 3.6 bounds the 0DT complexity of any boolean function by a poly-logarithmic function of its sparsity:

$$D^{0\text{-dt}}(f) = O(\log^5 \text{spar}(f)).$$

The final step is to connect zero decision tree complexity back to the AND decision tree complexity. Mukhopadhyay and Loff [ML19] observed that 0DT complexity is equivalent to ADT complexity, up to a $\log n$ factor.

Claim 3.10 ([ML19]). Let $f : \{0, 1\}^n \rightarrow \mathbb{R}$. Then $\frac{D^{\wedge\text{-dt}}(f)}{\log n} \leq D^{0\text{-dt}}(f) \leq D^{\wedge\text{-dt}}(f)$.

This establishes an “almost” log-rank conjecture for AND-functions, up to a $\log n$ factor.

Theorem 3.11 (Log-rank Conjecture for AND-functions, [KLMY21]). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and assume that $\text{rank}(f_{\wedge}) = r$. Then $D^{\wedge\text{-dt}}(f) = O(\log^5 r \cdot \log n)$ and in particular $D^{\text{cc}}(f_{\wedge}) = O(\log^5 r \cdot \log n)$.

Using the connection to Unique Disjointness gives the following related theorem.

Theorem 3.12 (Deterministic Lifting for AND-functions, [KLMY21]). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and assume that $D^{\text{cc}}(f_{\wedge}) = d$. Then $D^{\wedge\text{-dt}}(f) = O(d^3 \cdot \log n)$.

We conjecture that in both cases, the $\log n$ factor can be removed.

Conjecture 3.13 (Log-rank conjecture for AND-functions). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and assume that $\text{rank}(f_{\wedge}) = r$. Then $D^{\wedge\text{-dt}}(f) \leq \text{polylog}(r)$.

Conjecture 3.14 (Lifting for AND-functions). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and assume that $D^{\text{cc}}(f_{\wedge}) = d$. Then $D^{\wedge\text{-dt}}(f) \leq \text{poly}(d)$.

We end this subsection with some speculative remarks on the above conjectures. A conspicuous feature of the strategy used above in proving log-rank and lifting (modulo the $\log n$ factor) for AND-functions is that we didn’t directly design ADTs for f . Instead, we went through the intermediate 0-DT model which can simulate ADTs with a $\log n$ overhead (and this overhead is necessary). It therefore seems desirable to find a more direct way of building ADTs. One difficulty in doing so the following: when an ADT queries a set $S \subseteq [n]$ and gets a 0, there can be many subcubes which are consistent with the answer but inconsistent with one another. In this sense, it is not entirely clear what type of progress has been made. We can resolve this inconsistency by binary searching for the first bit in S set to 0 and keeping everything else unset, but the number of queries needed to do so will depend on $|S|$. It seems plausible that a more sophisticated understanding of the monomial structure of sparse boolean functions could help us design complexity measures which more readily simplify under AND queries.

3.2 Randomized ADT complexity

In this subsection, we introduce randomized ADT complexity and related problems.

Definition 3.15 (Randomized ADT Complexity). *A randomized ADT T is a distribution over (deterministic) ADTs. The depth of T is defined as the maximal depth of ADTs in the support of T . We say T computes f if $\Pr[f(x) = T(x)] \geq \frac{2}{3}$ for any input x . The randomized ADT complexity of f , denoted by $R^{\wedge-dT}(f)$, is the minimal depth of a randomized ADT which computes f .*

Recall that randomized PDTs are exponentially stronger than deterministic PDTs. However, no such separation is known for ADTs. The following conjecture was posed in [KLMY21], speculating that randomness does not help ADTs.

Conjecture 3.16. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and assume that $R^{\wedge-dt}(f) = d$. Then $D^{\wedge-dt}(f) \leq \text{poly}(d, \log n)$.*

A $\log n$ factor is necessary here, as the following example shows.

Example 3.17. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a threshold function, which outputs 1 if and only if $|x| \geq n - 1$. Then $D^{\wedge-dt}(f) = \Omega(\log \text{spar}(f)) = \Omega(\log n)$. However, $R^{\wedge-dt}(f) = O(1)$, since we can sample a subset $S \subseteq [n]$ uniformly at random, then output 0 if both $\bigwedge_{i \in S} x_i$ and $\bigwedge_{i \notin S} x_i$ equal to 0, and output 1 otherwise.*

Next, we introduce *approximate sparsity*, which might be a useful tool for this problem. Informally, it is the smallest sparsity of a polynomial which approximates f on all inputs.

Definition 3.18 (Approximate Sparsity). *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$. The approximate sparsity of f is defined as the minimal sparsity of a polynomial $p : \{0, 1\}^n \rightarrow \mathbb{R}$ that satisfies $|p(x) - f(x)| \leq 1/3$ for all $x \in \{0, 1\}^n$.*

It is suspected that approximate sparsity is polynomially related to sparsity, which is sufficient for proving Conjecture 3.16.

Conjecture 3.19. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Assume that $\widehat{\text{spar}}(f) = r$. Then $\text{spar}(f) \leq \text{poly}(r)$.*

Claim 3.20. *Conjecture 3.19 implies Conjecture 3.16.*

Proof. Assume that $R^{\wedge-dt}(f) = d$. As each ADT in the support of the distribution of the randomized ADT has depth at most d , its sparsity is at most 3^d . A standard Chernoff bound shows that it suffices to consider distributions over $O(n)$ ADTs. This shows that $\widehat{\text{spar}}(f) \leq O(3^d n)$. Conjecture 3.19 then implies that $\text{spar}(f) \leq 3^{O(d)} n^{O(1)}$. Applying Theorem 3.11 gives $D^{\wedge-dt}(f) \leq O((d + \log n)^5 \cdot \log n)$ which proves Conjecture 3.16. \square

References

- [BCK14] Abhishek Bhattacharya, Sourav Chakraborty, and Raghav Kulkarni. Property testing bounds for linear and quadratic functions via parity decision trees. In *International Computer Science Symposium in Russia*, pages 97–110. Springer, 2014. 5, 15
- [BdW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. 3

- [BFI⁺18] Paul Beame, Noah Fleming, Russell Impagliazzo, Antonina Kolokolova, Denis Pankratov, Toniann Pitassi, and Robert Robere. Stabbing planes. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPICs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 17
- [BIK⁺92] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan R. Woods. Exponential lower bounds for the pigeonhole principle. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 200–220. ACM, 1992. 16
- [BPR95] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. In Frank Thomson Leighton and Allan Borodin, editors, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 575–584. ACM, 1995. 6
- [BPR97] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *J. Symb. Log.*, 62(3):708–728, 1997. 17
- [BPS07] Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity. *SIAM J. Comput.*, 37(3):845–869, 2007. 17
- [BTW15] Eric Blais, Li-Yang Tan, and Andrew Wan. An inequality for the Fourier spectrum of parity decision trees. *arXiv preprint arXiv:1506.01055*, 2015. 5, 11
- [CFK⁺19a] Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting for BPP using inner product. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 35:1–35:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 6, 7
- [CFK⁺19b] Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting using low-discrepancy gadgets. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:103, 2019. 4, 5, 6
- [CK16] Ruiwen Chen and Valentine Kabanets. Correlation bounds and #SAT algorithms for small linear-size circuits. *Theor. Comput. Sci.*, 654:2–10, 2016. 16
- [CMS19] Arkadev Chattopadhyay, Nikhil S. Mande, and Suhail Sherif. The log-approximate-rank conjecture is false. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 42–53. ACM, 2019. 7, 8, 10, 12, 14, 15
- [CR12] Amit Chakrabarti and Oded Regev. An optimal lower bound on the communication complexity of Gap-Hamming-Distance. *SIAM J. Comput.*, 41(5):1299–1317, 2012. 3
- [dRNV16] Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In

Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 295–304. IEEE Computer Society, 2016. 6

- [GKMP20] Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is NP-hard. *arXiv preprint arXiv:2004.08037*, 2020. 3, 6
- [GKRS18] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 3
- [Gol17] Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. 15
- [Göö15] Mika Göös. Lower bounds for clique vs. independent set. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1066–1076. IEEE, 2015. 3
- [GP18] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM J. Comput.*, 47(5):1778–1806, 2018. 3, 17
- [GPW17] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 132–143. IEEE Computer Society, 2017. 4, 5, 6, 7
- [GPW18a] Mika Goos, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM Journal on Computing*, 47(6):2435–2450, 2018. 3, 7
- [GPW18b] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *computational complexity*, 27(2):245–304, 2018. 19
- [GS08] Ben Green and Tom Sanders. Boolean functions with small spectral norm. *Geometric and Functional Analysis*, 18(1):144–162, 2008. 10
- [HHL18] Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for XOR functions. *SIAM Journal on Computing*, 47(1):208–217, 2018. 5, 6, 13, 14
- [Hua19] Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *CoRR*, abs/1907.00847, 2019. 3
- [IPU94] Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*, pages 220–228. IEEE Computer Society, 1994. 17
- [IS20] Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Ann. Pure Appl. Log.*, 171(1), 2020. 17
- [Kla10] Hartmut Klauck. A strong direct product theorem for disjointness. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 77–86. ACM, 2010. 14

- [KLMY21] Alexander Knop, Shachar Lovett, Sam McGuire, and Weiqiang Yuan. Log-rank and lifting for AND-functions. *STOC 2021, to appear*, 2021. 5, 6, 18, 20, 21
- [KMR17] Pravesh K Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of CSPs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 590–603, 2017. 3
- [KMSY19] Gillat Kol, Shay Moran, Amir Shpilka, and Amir Yehudayoff. Approximate nonnegative rank is equivalent to the smooth rectangle bound. *Comput. Complex.*, 28(1):1–25, 2019. 8
- [KN06] Eyal Kushilevitz and Noam Nisan. Communication complexity. 2006. 2
- [Kno17] Alexander Knop. IPS-like proof systems based on binary decision diagrams. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:179, 2017. 17
- [LLZ11] Ming Lam Leung, Yang Li, and Shengyu Zhang. Tight bounds on communication complexity of symmetric XOR functions in one-way and SMP models. In *International Conference on Theory and Applications of Models of Computation*, pages 403–408. Springer, 2011. 5
- [LM19] Bruno Loff and Sagnik Mukhopadhyay. Lifting theorems for equality. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPICs*, pages 50:1–50:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 5, 6
- [LMM⁺20] Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. In *Electron. Colloquium Comput. Complex.*, page 111, 2020. 6
- [Lov16] Shachar Lovett. Communication is bounded by root of rank. *Journal of the ACM (JACM)*, 63(1):1–9, 2016. 7
- [LS88] L. Lovasz and M. Saks. Lattices, Möbius functions and communication complexity. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science, SFCS '88*, page 81–90, USA, 1988. IEEE Computer Society. 3, 6, 7
- [LS09] Troy Lee and Adi Shraibman. Lower bounds in communication complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–398, 2009. 7
- [LY21] Jiayu Li and Tianqi Yang. $3.1n - o(n)$ circuit lower bounds for explicit functions. *Electron. Colloquium Comput. Complex.*, 28:23, 2021. 16
- [ML19] Sagnik Mukhopadhyay and Bruno Loff. Lifting theorems for equality. In *STACS 2019*, 2019. 20
- [MO09] Ashley Montanaro and Tobias Osborne. On the communication complexity of XOR functions. *arXiv preprint arXiv:0909.3392*, 2009. 5, 14

- [MS82] Kurt Mehlhorn and Erik Meineche Schmidt. Las Vegas is better than determinism in VLSI and distributed computing (extended abstract). In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 330–337. ACM, 1982. 6, 7
- [NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994. 3, 11, 19
- [NW95] Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995. 19
- [OS07] Ryan O’Donnell and Rocco A Servedio. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2007. 11
- [OWZ⁺14] Ryan ODonnell, John Wright, Yu Zhao, Xiaorui Sun, and Li-Yang Tan. A composition theorem for parity kill number. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 144–154. IEEE, 2014. 5, 10, 12
- [PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Comput. Complex.*, 3:97–140, 1993. 16
- [Raz92a] Alexander A. Razborov. On the distributional complexity of disjointness. *Theor. Comput. Sci.*, 106(2):385–390, 1992. 3
- [Raz92b] Alexander A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106:385–390, 1992. 19
- [Raz95] Alexander A. Razborov. Bounded arithmetic and lower bounds in boolean complexity. In Peter Clote and Jeffrey B. Remmel, editors, *Feasible Mathematics II*, pages 344–386, Boston, MA, 1995. Birkhäuser Boston. 16
- [RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. 4, 5, 6
- [RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. 2
- [San19] Tom Sanders. Boolean functions with small spectral norm, revisited. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 167, pages 335–344. Cambridge University Press, 2019. 5, 10
- [She10] Alexander A. Sherstov. On quantum-classical equivalence for composed communication problems. *Quantum Information & Computation*, 10(5&6):435–455, 2010. 4, 6
- [STIV17] Amir Shpilka, Avishay Tal, and Ben lee Volk. On the structure of boolean functions with small spectral norm. *computational complexity*, 26(1):229–273, 2017. 10
- [TWXZ13] Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 658–667. IEEE Computer Society, 2013. 5, 10, 12, 14

- [Wat19] Thomas Watson. A $ZPP^{NP[1]}$ lifting theorem. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPICs*, pages 59:1–59:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 7
- [Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 209–213. ACM, 1979. 2
- [Yao15] Penghui Yao. Parity decision tree complexity and 4-party communication complexity of XOR-functions are polynomially equivalent. *arXiv preprint arXiv:1506.02936*, 2015. 5
- [Zha14] Shengyu Zhang. Efficient quantum protocols for XOR functions. In *Proceedings of the 25th annual ACM-SIAM Symposium on Discrete algorithms*, pages 1878–1885. SIAM, 2014. 5
- [ZS09] Zhiqiang Zhang and Yaoyun Shi. Communication complexities of symmetric XOR functions. *Quantum Information & Computation*, 9(3):255–263, 2009. 5
- [ZS10] Zhiqiang Zhang and Yaoyun Shi. On the parity complexity measures of boolean functions. *Theoretical Computer Science*, 411(26-28):2612–2618, 2010. 5