Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 173 (2014)

Majority is incompressible by $\mathsf{AC}^0[p]$ circuits*

Igor C. Oliveira† Rahul Santhanam‡

April 14, 2015

Abstract

We consider $\mathcal{C}$-compression games, a hybrid model between computational and communication complexity. A $\mathcal{C}$-compression game for a function $f\colon \{0,1\}^n \to \{0,1\}$ is a two-party communication game, where the first party Alice knows the entire input x but is restricted to use strategies computed by $\mathcal{C}$-circuits, while the second party Bob initially has no information about the input, but is computationally unbounded. The parties implement an *interactive* communication protocol to decide the value of $f(x)$, and the *communication cost* of the protocol is the maximum number of bits sent by Alice as a function of $n = |x|$.

We show that any $\mathsf{AC}^0_d[p]$-compression protocol to compute $\mathsf{Majority}_n$ requires communication $n/(\log n)^{2d+O(1)}$, where p is prime, and $\mathsf{AC}^0_d[p]$ denotes polynomial size unbounded fan-in depth-d Boolean circuits extended with modulo p gates. This bound is essentially optimal, and settles a question of Chattopadhyay and Santhanam (2012). This result has a number of consequences, and yields a tight lower bound on the total fan-in of oracle gates in constant-depth oracle circuits computing $\mathsf{Majority}_n$.

We define *multiparty* compression games, where Alice interacts in parallel with a polynomial number of players that are not allowed to communicate with each other, and communication cost is defined as the sum of the lengths of the longest messages sent by Alice during each round. In this setting, we prove that the randomized r-round $\mathsf{AC}^0[p]$-compression cost of $\mathsf{Majority}_n$ is $n^{\Theta(1/r)}$. This result implies almost tight lower bounds on the maximum individual fan-in of oracle gates in certain restricted bounded-depth oracle circuits computing $\mathsf{Majority}_n$. Stronger lower bounds for functions in NP would separate NP from NC^1.

Finally, we consider the round separation question for two-party AC^0-compression games, and significantly improve known separations between r-round and $(r+1)$-round protocols, for any constant r.

*This work was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no. 615075. The first author was also supported by NSF grants CCF-1116702 and CCF-1115703.

†oliveira@cs.columbia.edu, Department of Computer Science, Columbia University.

‡rsanthan@inf.ed.ac.uk, Laboratory for Foundations of Computer Science, University of Edinburgh.

ISSN 1433-8092

Contents

1 Introduction

1.1 Motivation and Background

Computational complexity theory investigates the complexity of solving explicit problems in various computational models. While fairly strong lower bounds are known for restricted models such as constant-depth circuits (Ajtai [Ajt83], Furst, Saxe, and Sipser [FSS84], Yao [Yao85], and Håstad [Hås86]) and monotone circuits (Razborov [Raz85], Andreev [And85], and Alon and Boppana [AB87]), our understanding of general Boolean circuits is still very limited. For example, our current state of knowledge does not rule out that every function in $\mathsf{NTIME}(2^n)$ is computed by Boolean circuits of linear size.

Several barriers have been identified to proving lower bounds for general Boolean circuits, such as relativization (Baker, Gill, and Solovay [BGS75]), algebrization (Aaronson and Wigderson [AW09]), and the "natural proofs" barrier (Razborov and Rudich [RR97]). Most known lower bound techniques for restricted models are "naturalizable", and it is believed that substantially different methods will be required in order to prove strong lower bounds for unrestricted models.

In spite of this, the techniques used to prove lower bounds for weaker models are still interesting, and an improved understanding of these techniques can have substantial benefits. First, there is a developing theory of connections between unconditional lower bounds and algorithmic results, which involves satisfiability algorithms, learning algorithms, truth-table generation, among other models (cf. Williams [Wil14a], Oliveira [Oli13], and Santhanam [San12]). In particular, such connections provide new insights and results in both areas, and a better understanding of restricted classes of circuits can lead to improved algorithms (cf. Williams [Wil14b]). Second, strong enough lower bounds for weaker models imply lower bounds for more general models (Valiant [Val77, Val83], see Viola [Vio09] for a modern exposition). In a similar vein, we mention the surprising results from Allender and Koucký [AK10] showing that, in some cases, weak circuit size lower bounds of the form $n^{1+\varepsilon}$ yield much stronger results.

Furthermore, even if known proof techniques individually naturalize, it is possible they could be used as ingredients of a more sophisticated approach which is more powerful. A recent striking example of this is the use by Williams [Wil14c] of structural characterizations of ACC^0 circuits, together with various complexity tools such as completeness for problems on succinctly represented inputs, diagonalization, and the easy witness method, in order to separate NEXP from ACC^0. Given the paucity of techniques in the area of complexity lower bounds, it makes sense to try to properly understand the techniques we do have.

We focus in this work on $\mathcal{C}$-compression games (Chattopadhyay and Santhanam [CS12]), where $\mathcal{C}$ is some class of Boolean circuits. A $\mathcal{C}$-compression game is a 2-player (interactive) communication game where the first player Alice is computationally bounded (by being restricted to play strategies in $\mathcal{C}$) and has access to the entire input $x \in \{0,1\}^n$, while the second player Bob is computationally unbounded and initially has no information about the input. Alice and Bob communicate to compute $f(x)$ for a fixed Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, and the question is how many bits of communication sent by Alice are required. Note that if f is computable by $\mathcal{C}$, then 1 bit of communication suffices, as Alice can compute $f(x)$ by herself, and send the answer to Bob. Thus, if we are interested in unconditional lower bounds on the communication cost for an explicit function, we must study circuit classes $\mathcal{C}$ where lower bounds are already known for explicit functions, such as constant-depth circuits, and their extension with modulo p gates.

Compression games hybridize between communication complexity and computational complex-

ity as follows. In the traditional two-party communication complexity model, Alice and Bob are symmetric – they each know half of the input, and communicate to compute a given function on the whole input. Neither party is computationally bounded. Thus they are equally constrained (or unconstrained) informationally as well as computationally. In the compression game setting, an asymmetry appears. Alice now has an informational advantage over Bob – she begins with knowledge of the whole input, while Bob has no knowledge about the input at all. However, this informational advantage is offset by a computational constraint – Alice can only use strategies computable from $\mathcal{C}$. Thus studying compression games can be thought of as studying the tradeoff between information and computation. Typically, when studying the question of lower bounds against $\mathcal{C}$, we are merely interested in whether a function f is computable in $\mathcal{C}$ or not. Now, we are concerned instead by how much information can be obtained about $f(x)$ using merely circuits from $\mathcal{C}$, or conversely, how much assistance a $\mathcal{C}$-bounded party requires from an unbounded one in order to compute $f(x)$. In other terms, we would like to obtain a refined quantitative picture of solvability by $\mathcal{C}$-circuits, rather than a purely qualitative one.

Communication complexity has long been an important tool in the complexity theorist's toolkit. In particular, several lower bound techniques such as the crossing sequence method, the Nečiporuk method [Neč66] and the Khrapchenko method [Khr71] can be interpreted as uses of communication complexity (cf. Kushilevitz and Nisan [KN97]). Often, when a computational model is relatively weak, lower bound techniques exploit some sort of information bottleneck in the model, which is how communication complexity enters the picture. By studying compression games, where the model explicitly incorporates both communication and computation, we hope to better understand the interplay between communication complexity techniques and computational complexity techniques.

We explore in this work the power of the *polynomial approximation method* (Razborov [Raz87], Smolensky [Smo87]) and the *random restriction method* (cf. Furst, Saxe, and Sipser [FSS84] and Håstad [Hås86]) in the context of interactive compression games. We use these techniques and the compression framework to prove significant generalizations of known lower bounds for constant-depth circuits.

Compression games have been considered before, both to prove unconditional and conditional lower bounds. The pioneering work of Dubrov and Ishai [DI06] showed that Parity_n requires AC^0-compression cost $n^{1-\varepsilon}$ (for any fixed $\varepsilon > 0$, and large enough n) when there is only one round of communication between Alice and Bob. Dubrov and Ishai were motivated by questions about the randomness complexity of sampling, and their work has later found applications in leakage-resilient cryptography (Faust et al. [FRR+10]). Chattopadyay and Santhanam [CS12] strengthened the Dubrov-Ishai lower bound to $n/\mathsf{poly}(\log n)$, and showed that the lower bound holds for multi-round games where Alice is allowed to use a randomized strategy. Their main technique was a generic connection between correlation and multi-round compression. As strong correlation lower bounds are not known for $\mathsf{AC}^0[p]$ circuits (see e.g. Srinivasan [Sri13]), their technique does not yield strong lower bounds for multi-round $\mathsf{AC}^0[p]$-compression games, which constitute the main topic of this work.

The investigation of single-round compression (also known as instance compression) has found connections to other topics in areas such as cryptography (Harnik and Naor [HN10]), parameterized complexity (cf. Bodlaender et al. [BDFH09]), probabilistic checkable proofs (Fortnow and Santhanam [FS11]), and structural complexity (Buhrman and Hitchcock [BH08]), and has received considerable attention recently (see e.g. Drucker [Dru12] and Dell [Del14]). There has also been a long line of work on proving lower bounds for $\mathsf{SIZE}(\mathsf{poly}(n))$-compression games under complexity-

theoretic assumptions (cf. Dell and van Melkebeek [DvM14]), but papers along this line use very different ideas, and hence are tangential to our work.

1.2 Main Results and Techniques

For a circuit class $\mathcal{C}$, we use $\mathcal{C}_d$ to denote the restriction of $\mathcal{C}$ to polynomial size circuits of depth d. For instance, AC^0_d refers to polynomial size depth-d circuits. Recall that $\mathsf{Majority}_n\colon \{0,1\}^n \to \{0,1\}$ is the function that is 1 on an input x if and only if $\sum_{i\in[n]} x_i \geq n/2$. Further, we let $\mathsf{MOD}^n_q\colon \{0,1\}^n \to \{0,1\}$ be the function that is 1 if and only if q divides $\sum_{i\in[n]} x_i$.

The proof that $\mathsf{Majority}_n \notin \mathsf{AC}^0_d[p]$ for $d(n) = o(\log n/\log\log n)$ (Razborov [Raz87], Smolensky [Smo87]) remains one of the strongest lower bounds for an explicit function. There are no known explicit lower bounds for polynomial size circuits of depth $d = \omega(\log n/\log\log n)$, nor for constant depth circuits with arbitrary (composite) modulo gates.

In the framework of compression games, the Razborov-Smolensky lower bound is equivalent to the claim that in any $\mathsf{AC}^0[p]$ game for $\mathsf{Majority}$, there must be non-trivial communication between Alice and Bob. More recently, Chattopadhyay and Santhanam [CS12] proved that in any *randomized single-round* $\mathsf{AC}^0_d[p]$-compression protocol for this function, Alice must communicate $\sqrt{n}/(\log n)^{O(d)}$ bits. However, their technique does not extend to multiple-round compression games. Before this work, the only known technique to prove unconditional lower bounds for games with an arbitrary number of rounds used a connection between compressibility and correlation. The lack of strong correlation bounds for low-degree $\mathbb{F}_p$ polynomials computing explicit Boolean functions prevents us from using this connection to get $\mathsf{AC}^0[p]$-compression lower bounds (see Srinivasan [Sri13] for more details).

In this work, we bypass this difficulty through a new application of the polynomial approximation method, obtaining the following result.

Theorem 1.1. *Let p be a prime number. There exists a constant $c \in \mathbb{N}$ such that, for any $d \in \mathbb{N}$, and every $n \in \mathbb{N}$ sufficiently large, the following holds.*

(*i*) *Any $\mathsf{AC}^0_d[p]$-compression game for $\mathsf{Majority}_n$ (with any number of rounds) has communication cost at least $n/(\log n)^{2d+c}$.*

(*ii*) *There exists a single-round AC^0_d-compression game for $\mathsf{Majority}_n$ with communication cost at most $n/(\log n)^{d-c}$.*

The argument for the lower bound part of this result proceeds roughly as follows. First, we show via a reduction in the interactive compression framework that a protocol for $\mathsf{Majority}_n$ can be used to compress other symmetric functions, such as MOD^n_q. In other words, it is enough to prove a strong communication lower bound for MOD^n_q in order to establish the lower bound in Theorem 1.1. We then employ a general technique that allows us to transform an interactive protocol for a Boolean function f into an exponentially large circuit computing f, following an approach introduced in Chattopadhyay and Santhanam [CS12]. We have thus reduced the original problem involving computation and communication to a certain circuit lower bound for MOD_q.

A crucial ingredient in our proof is a new exponential lower bound for a certain class of bounded-depth circuits extended with modulo p gates computing the MOD_q function. Although obtaining circuit lower bounds for depth d circuits beyond size roughly $2^{n^{1/(d-1)}}$ is a major open problem in circuit complexity (see e.g. Viola [Vio09]), we show that, under a certain *semantic* constraint

on the $\mathsf{AC}^0_d[p]$ circuit, MOD^n_q requires circuits of size $2^{n/(\log n)^{O(d)}}$. More specifically, we consider circuits consisting of a disjunction of exponentially many polynomial size circuits, for which the following holds: whenever the top gate evaluates to true, precisely one subcircuit evaluates to true.

The proof of this circuit lower bound relies on the application of the polynomial approximation method in the *exponentially small error regime*, as opposed to the original proofs of Razborov and Smolensky, which are optimized with constant error. In particular, this approach allows us to prove a stronger lower bound that avoids the correlation barrier mentioned before. In order to implement this idea, we rely on a recent strengthening of their method introduced by Kopparty and Srinivasan [KS12], and on an extension of the degree lower bounds of Razborov and Smolensky to very small error. We believe that this new circuit lower bound may be of independent interest, and that semantic restrictions will find more applications in circuit complexity. Altogether, these results give the lower bound in Theorem 1.1.

Theorem 1.1 implies a new result for $\mathsf{AC}^0[p]$ circuits extended with arbitrary oracle gates, which we state next.

Corollary 1.2. *Let $p \geq 2$ be prime, and $d \in \mathbb{N}$. There exists a constant $c \in \mathbb{N}$ such that, for every sufficiently large n, the following holds. If $\mathsf{Majority}_n$ is computed by polynomial-size $\mathsf{AC}^0_d[p]$ circuits with arbitrary oracle gates, then the total fan-in of the oracle gates is at least $n/(\log n)^{2d+c}$.*

Another interesting consequence of Theorem 1.1 is that it provides information about the *structure* of polynomial size circuits with modulo p gates computing $\mathsf{Majority}_n$. More precisely, it implies that in any layered circuit, at least $\lfloor n/(\log n)^{2k+c} \rfloor$ gates must be present in the k-th layer, which is essentially optimal.

Observe that Theorem 1.1 holds for deterministic compression games. For randomized protocols, in which Alice can employ a probabilistic strategy, we use our techniques to prove the following strengthening over previous results.

Theorem 1.3. *Let p and q be distinct primes. There exists a constant $c \in \mathbb{N}$ such that, for any $d \in \mathbb{N}$, and $n \in \mathbb{N}$ sufficiently large, every randomized $\mathsf{AC}^0_d[p]$-compression game for MOD^n_q with any number of rounds and error at most $1/3$ has communication cost at least $\sqrt{n}/(\log n)^{d+c}$.*

We stress that Theorems 1.1 and 1.3 hold both for $\mathsf{Majority}$ and MOD_q, whenever $p \neq q$ are distinct primes. Determining the correct communication cost for probabilistic and average-case games for these functions remains an interesting open problem. (We discuss these models in more detail in Section 2.)

We also consider a model of *multiparty* compression games. In this framework, Alice is allowed to interact during each round with k additional parties, and the communication cost of the round is defined to be the length of the longest message sent by Alice to one of the parties. Further, the cost of the protocol on a given input is defined as the sum of the costs of the individual rounds. We stress that the extra parties are not allowed to interact with each other during the execution of the protocol.

This is a natural communication framework, motivated by the question of lower bounds for oracle circuits. Lower bounds in this model with a bounded number of rounds imply lower bounds on the maximum individual fan-in of oracle gates in oracle circuits with a bounded number of such layers.

We prove the following bounds on the randomized multiparty $\mathsf{AC}^0[p]$-compression cost of $\mathsf{Majority}$.

Theorem 1.4. *Let $p \in \mathbb{N}$ be a fixed prime. For every $k, r, d \in \mathbb{N}$, the following holds.*

(*i*) *There exists a deterministic* $n^{1/r}$*-party* r*-round* $\mathsf{AC}^0[p]$*-compression game for* $\mathsf{Majority}_n$ *with cost* $\widetilde{O}(n^{1/r})$.

(*ii*) *Every randomized* n^k*-party* r*-round* $\mathsf{AC}^0_d[p]$*-compression game for* $\mathsf{Majority}_n$ *has cost* $\widetilde{\Omega}(n^{1/2r})$.

The proof of Theorem 1.4 also employs the polynomial approximation method, although the argument is different in this case. Observe that this result says that the communication cost of $\mathsf{Majority}_n$ in the randomized multiparty framework is $n^{\Theta(1/r)}$ for r-round protocols. In other words, allowing Alice to interact with more parties for more time reduces communication considerably (under the definition of communication cost for multiparty games).

We obtain a consequence of Theorem 1.4 for oracle circuits where there are a bounded number r of such layers, i.e., there are no more than r oracle gates on any input-output path in the circuit.

Corollary 1.5. *Let* $p \geq 2$ *be prime, and* $r, d \in \mathbb{N}$. *If* $\mathsf{Majority}_n$ *is computed by an* $\mathsf{AC}^0_d[p]$ *circuit of polynomial size with arbitrary oracle gates that contains at most* r *layers of such gates, then there is some oracle gate with fan-in at least* $\widetilde{\Omega}(n^{1/2r})$.

In fact, lower bounds for multiparty games are connected to the NP versus NC^1 question. It is possible to show that every Boolean function in $\mathsf{NC}^1/\mathsf{poly}$ admits $\mathsf{poly}(n)$-party r-round AC^0-compression games with cost $n^{O(1/r)}$. Thus, proving a lower bound of $n^{\Omega(1)}$ on the cost of $\mathsf{poly}(n)$-party AC^0-compression games with $\omega(1)$ rounds for a function in NP would separate NP from $\mathsf{NC}^1/\mathsf{poly}$. We conjecture that such a lower bound holds for the Clique function. Note that it is already known that strong enough lower bounds on the size of constant-depth circuits for NP functions implies a separation between NP and NC^1 (cf. Viola [Vio09]). The novelty here is that sufficiently strong results about *polynomial-size* constant depth circuits imply similar separations. Essentially, the computation of logarithmic-depth circuits can be factored into constant-depth and low-communication components, and our multiparty communication game models precisely this mixture of notions.

There is an interesting contrast in the statement of Theorem 1.1: while the lower bound holds for protocols with any number of rounds, the upper bound is given by a single-round protocol. It is natural to wonder whether in the compression setting interaction allows Alice to solve more computational problems. We provide a natural example of the power of interaction in our framework in Section 6, where we observe that, while the inner product function cannot be computed by polynomial size $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits (Hajnal et al. [HMP$^+$93]), there exists an efficient two-party $(\mathsf{MAJ} \circ \mathsf{MAJ})$-compression game for this function.

In a similar direction, a *quantitative* study of the power of interaction in two-party compression games was initiated by Chattopadhyay and Santhanam [CS12] (with respect to AC^0-compression games). They obtained a quadratic gap in communication when one considers r and $(r-1)$-round protocols for a specific Boolean function. We obtain the following strengthening of their round separation theorem.

Theorem 1.6. *Let* $r \geq 2$ *and* $\varepsilon > 0$ *be fixed parameters. There is an explicit family of functions* $f = \{f_n\}_{n \in \mathbb{N}}$ *with the following properties:*

(*i*) *There exists an* $\mathsf{AC}^0_2(n)$*-bounded protocol* Π_n *for* f_n *with* r *rounds and cost* $c(n) \leq n^{\varepsilon}$, *for every* $n \geq n_f$, *where* n_f *is a fixed constant that depends on* f.

(*ii*) *Any* $\mathsf{AC}^0(\mathsf{poly}(n))$*-bounded protocol* Π *for* f *with* $r-1$ *rounds has cost* $c(n) \geq n^{1-\varepsilon}$, *for every* $n \geq n_\Pi$, *where* n_Π *is a fixed constant that depends on* Π.

Our hard function is based on a pointer jumping problem with a grid structure, while Chattopadhyay and Santhanam uses a tree structure. Similar constructions have been used in other works in communication complexity in the information theoretic setting (Papadimitriou and Sipser [PS84], and subsequent works), but our analysis needs to take into account computational considerations as well.

The proof of Theorem 1.6 relies on a careful application of the *random restriction method*, coupled with a round elimination strategy. Observe that the upper bound is achieved by protocols where Alice's strategy can be implemented by linear-size DNFs, while the communication lower bound holds for polynomial size circuits.

1.3 Organization

We define interactive compression games and introduce notation in the next section. In Section 3, we give the proof of our main result, deferring the discussion of some auxiliary results to the Appendix. Multiparty compression games are discussed in Section 4, followed by applications of our communication lower bounds to circuits with oracle gates in Section 5. A natural example for which interactive compression can be easier than computation is presented in Section 6. The round separation theorem for AC^0 games is proved in Section 7. Finally, we mention a few open problems and research directions in Section 8.

2 Preliminaries and Notation

The results of this paper are essentially self-contained, but some familiarity with basic notions from complexity theory and communication complexity can be helpful. A good introduction to these areas can be found in [AB09] and [KN97], respectively.

Basic definitions. For any positive integer $m \in \mathbb{N}$, let $[m] \stackrel{\text{def}}{=} \{1, \ldots, m\}$. We use $\mathsf{Majority}_n$ to denote the Boolean function over n variables that is 1 if and only if $\sum_i x_i \geq n/2$. For a prime p, we let MOD_p^n be the Boolean function over n variables that is 1 if and only if p divides $\sum_i x_i$. We let $\mathsf{Parity}_n \stackrel{\text{def}}{=} \neg\mathsf{MOD}_2^n$. A function $h\colon \{0,1\}^n \to \{0,1\}$ is *symmetric* if there exists a function $\phi\colon [n] \to \{0,1\}$ such that $h(x) = \phi(\sum_i x_i)$, for every $x \in \{0,1\}^n$. Clearly, $\mathsf{Majority}_n$ and MOD_p^n are symmetric functions. We say that a Boolean function f ε-approximates a Boolean function g over a distribution $\mathcal{D}$ if $\Pr_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x}) \neq g(\mathbf{x})] \leq \varepsilon$. An ε-error *probabilistic* polynomial $\mathbf{Q}(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n]$ for a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is a distribution $\mathcal{E}$ over polynomials such that, for every $x \in \{0,1\}^n$, $Pr_{\mathbf{Q}\sim\mathcal{E}}[f(x) \neq \mathbf{Q}(x)] \leq \varepsilon$.[1] The degree of a probabilistic polynomial is the maximum degree over the polynomials on which $\mathcal{E}$ is supported. We say that functions $f\colon \{0,1\}^n \to \{0,1\}$ and $g\colon \{0,1\}^n \to \{0,1\}$ are *disjoint* if $f^{-1}(1) \cap g^{-1}(1) = \emptyset$. Given a string w, we use $|w|$ to denote the length of w, and $|w|_1$ to denote the number of 1s in w. We will use p and q to denote prime numbers throughout the text, unless noted otherwise.

Languages and circuit classes. Given a language $L \subseteq \{0,1\}^*$, we let $L_n \stackrel{\text{def}}{=} L \cap \{0,1\}^n$. We view L_n as a Boolean function in the natural way. We will use $\mathcal{C}$ to denote a circuit class, such as AC^0 and $\mathsf{AC}^0[p]$. Unless stated otherwise, assume that any circuit class discussed in this paper

[1] We will use boldface notation whenever we want to emphasize that we are referring to a random variable or a probability distribution over the corresponding structures.

contains AND, OR, and NOT gates of unbounded fan-in. Our results hold with more general circuit classes, but we stick with this definition for simplicity. The size of a circuit corresponds to the total number of gates in the circuit. We use $\mathcal{C}_d(s(n))$ to denote the same class restricted to circuits of depth d and size $O(s(n))$. For instance, we abuse notation and write $\mathsf{AC}^0_d[p](\mathsf{poly}(n))$ to denote the set of languages decided by polynomial size circuits of depth at most d consisting of unbounded fan-in AND, OR, NOT and MOD_p gates, for a fixed prime $p \in \mathbb{N}$. As a convention, if we write $\mathcal{C}$ without a depth and size specialization, assume that it consists of constant depth polynomial size circuits with gates from $\mathcal{C}$. As usual, we will identify $\mathcal{C}$ both as a set of languages, and as a class of circuits, depending on the context. Furthermore, if C is a fixed circuit, we may also use C to refer to the Boolean function computed by this circuit. The correct meaning will always be clear in both cases.

Deterministic compression games. Given a circuit class $\mathcal{C}$ and a language L, we define a communication game between two players Alice and Bob. The goal is to decide whether a given string $x \in \{0,1\}^n$ belongs to L. We describe this game informally as follows. Alice knows x, but her computational power is limited to functions computed by circuits from $\mathcal{C}$. On the other hand, Bob can perform arbitrary computations, but has no information about x during the beginning of the game. The players exchange messages during the execution of the protocol, and at the end should be able to decide whether $x \in L$. The goal is to compute the initial function correctly while minimizing the total number of bits sent by Alice during the game.

Formally, a $\mathcal{C}$-bounded protocol $\Pi_n = \langle C^{(1)}, \ldots, C^{(r)}, f^{(1)}, \ldots, f^{(r-1)}, E_n \rangle$ with $r = r(n)$ rounds consists of a sequence of $\mathcal{C}$-circuits for Alice, a strategy for Bob, given by functions $f^{(1)}, \ldots, f^{(r-1)}$, and a set of accepting transcripts E_n. We associate to every protocol Π_n its signature $\mathsf{signature}(\Pi_n) = (n, s_1, t_1, \ldots, t_{r-1}, s_r)$, which is the sequence corresponding to the input size $n = |x|$ and the length of the messages exchanged by Alice and Bob during the execution of the protocol. For convenience, let $s = \sum_{i \in [r]} s_i$, and $t = \sum_{i \in [r-1]} t_i$. We always have $E_n \subseteq \{0,1\}^{t+s}$. In addition, we let $\mathsf{rounds}(\Pi_n) \stackrel{\text{def}}{=} r$. For every $i \in [r]$,

$$C^{(i)} \colon \{0,1\}^{n + \sum_{j<i}(s_j + t_j)} \to \{0,1\}^{s_i},$$

and for every $i \in [r-1]$,

$$f^{(i)} \colon \{0,1\}^{\sum_{j \le i} s_j} \to \{0,1\}^{t_i}.$$

In other words, before the beginning of the i-th round, Alice has sent messages $a^{(i)}, \ldots, a^{(i-1)}$ of size $s_1, \ldots, s_{i-1}$, respectively, and Bob has replied with messages $b^{(1)}, \ldots, b^{(i-1)}$ of size $t_1, \ldots, t_{i-1}$, respectively. The next message sent by Alice is given by $a^{(i)} \stackrel{\text{def}}{=} C^{(i)}(x, a^{(1)}, b^{(1)}, \ldots, a^{(i-1)}, b^{(i-1)})$. On the other hand, since Bob has unlimited computational power, its message during the i-th round is given simply by $b^{(i)} \stackrel{\text{def}}{=} f^{(i)}(a^{(1)}, \ldots, a^{(i)})$. The transcript of Π_n on $x \in \{0,1\}^n$ is the sequence of messages exchanged by Alice and Bob during the execution of the protocol on x, and will be denoted by $\mathsf{transcript}_{\Pi_n}(x) \stackrel{\text{def}}{=} \langle a^{(1)}, b^{(1)}, \ldots, a^{(r)} \rangle \in \{0,1\}^{s+t}$. We say that Π_n solves the compression game of a function $h_n \colon \{0,1\}^n \to \{0,1\}$ if

$$h(x) = 1 \quad \iff \quad \mathsf{transcript}_{\Pi_n}(x) \in E_n.$$

Finally, we let $\mathsf{cost}(\Pi_n) \stackrel{\text{def}}{=} s$. We stress that the length of the messages sent by Bob does not contribute to the cost of the protocol, and we assume for convenience that the length of these

messages are limited by the size of the circuits in $\mathcal{C}$. Observe that a *single-round* game consists of a protocol Π_n with $\mathsf{signature}(\Pi_n) = (n, s_1)$. Put another way, Alice sends a single message $a^{(1)} \in \{0,1\}^{s_1}$, and a decision is made.

Given a language L and a circuit class $\mathcal{C}$, we say that a sequence of $\mathcal{C}$-bounded protocols $\Pi = \{\Pi_n\}_{n\in\mathbb{N}}$ solves the compression game of L with cost $c(n)$ and $r(n)$ rounds if, for every n, Π_n solves the compression game of L_n, and in addition satisfies $\mathsf{cost}(\Pi_n) \leq c(n)$ and $\mathsf{rounds}(\Pi_n) \leq r(n)$.

Observe that if $L \in \mathcal{C}$ then Alice can compute $L(x)$ by herself, and there is a trivial protocol of cost $c(n) = 1$ for L. On the other hand, for every language L there exists a protocol solving its compression game with cost $c(n) \leq n$, since Alice can simply send her whole input to Bob.

Probabilistic and average-case compression games. The definition presented before captures deterministic games computing a function correctly on every input x. Our framework can be extended naturally to probabilistic and average-case games.

First, in a *probabilistic* $\mathcal{C}$-compression game, Alice is allowed to use randomness when computing her next message, while Bob's strategy remains deterministic. Formally, each circuit $C^{(i)}$ has an additional input of uniformly distributed bits, and different circuits have access to independent bits. Clearly, on any $x \in \{0,1\}^n$, $\mathsf{Transcript}_{\Pi_n}(x)$ is now a random variable distributed over $\{0,1\}^{s+t}$. The other definitions remain the same. We say that Π_n solves the compression game of a function $h_n \colon \{0,1\}^n \to \{0,1\}$ with error probability at most $\gamma(n) \in [0,1]$ if, for every $x \in \{0,1\}^n$,

$$\begin{aligned} h_n(x) = 1 \quad &\Longrightarrow \quad \Pr_{\Pi_n}[\mathsf{Transcript}_{\Pi_n}(x) \in E_n] \geq 1 - \gamma(n), \text{ and if} \\ h_n(x) = 0 \quad &\Longrightarrow \quad \Pr_{\Pi_n}[\mathsf{Transcript}_{\Pi_n}(x) \in E_n] \leq \gamma(n). \end{aligned}$$

On the other hand, in a *average-case* $\mathcal{C}$-compression game, we have deterministic games as defined before, but allow a small error during the computation of h_n with respect to the uniform distribution over $\{0,1\}^n$. More precisely, we say that a deterministic protocol Π_n solves the compression game of h_n with error at most $\gamma(n) \in [0,1]$ if

$$\Pr_{x\sim\{0,1\}^n}[h_n(x) = 1 \iff \mathsf{transcript}_{\Pi_n}(x) \in E_n] \geq 1 - \gamma(n).$$

These definitions are extended to languages in the natural way. Since in this paper all circuit classes are non-uniform, any probabilistic protocol for a language L with error at most $\gamma(n)$ can be converted into an average-case protocol with error at most $\gamma(n)$ (simply by fixing the randomness of Alice in order to minimize the error probability over $\{0,1\}^n$).

Interacting with several Bobs. We discuss here a more general family of multi-party compression games that allow Alice to interact with multiple Bobs during a single round of the game. The different Bobs are not allowed to communicate with each other, only with Alice. The definition of round complexity for such games is slightly different than for standard 2-party compression games. The reason is as follows. For 2-party games, we can assume that the game concludes with a message to Bob, as Bob is all-powerful and can determine the result of the protocol from the final message. In the case of multi-party games, this assumption isn't well motivated, as no individual Bob might have access to all the information about the protocol. It makes more sense to say the game for a Boolean function h concludes with Alice computing whether $h(x) = 1$, where x is her input. Thus, a 1-round game will consist of Alice sending messages to the various Bobs, the Bobs responding, and finally Alice computing the answer. This naturally extends to a definition of r-round games.

We will also measure the cost of a protocol somewhat differently. We will again count only the communication from Alice to Bob, but the cost of a protocol will not be the sum of the lengths of all messages sent by Alice. Instead, we will define the cost of a round to be the maximum length of a message sent by Alice to some Bob, and then the cost of the protocol to be the sum of the costs over all rounds. This definition of protocol cost is motivated by the connection of our model with lower bounds on oracle circuits, which we elaborate later. A formal definition is presented below.

Let $\mathcal{C}$ be a circuit class, and $k = k(n), r = r(n)$ be arbitrary functions. A $\mathcal{C}$-bounded $(k+1)$-party protocol $\Pi_n^{[k]} = \langle D^{(1,1)}, \dots, D^{(1,k)}; D^{(2,1)}, \dots, D^{(2,k)}; \dots; D^{(r+1,1)}, g^{(1,1)}, \dots, g^{(r,1)}; g^{(1,2)}, \dots, g^{(r,2)}; \dots; g^{(1,k)}, \dots, g^{(r,k)} \rangle$ with r rounds consists of a sequence of $\mathcal{C}$-circuits for Alice, and strategies for each Bob_i, given by $g^{(1,i)} \dots g^{(r,i)}$. We associate to every k-party protocol $\Pi_n^{[k]}$ its signature $\mathsf{signature}(\Pi_n^{[k]}) = (n, s_1, t_1, \dots s_r, t_r)$, where for each $j \in [r], i \in [k]$, s_j is the maximum length of a message sent by Alice to any Bob_i during the j-th round, and t_j is the maximum length of a message sent by any Bob_i to Alice during the j-th round. For every $i \in [r], j \in [k]$, $D^{(i,j)}$ maps the sequence of the input x, all messages sent to Alice before the i-th round and all of Alice's messages before the i-th round to Alice's message in the j-th round to Bob_j. $D^{(r+1,1)}$ maps the sequence of x and all messages sent during the protocol to a single bit. For every $i \in [r], j \in [k]$, $g^{(i,j)}$ maps the sequence of all Alice's messages to Bob_j from the first to the i-th round to Bob_j's message to Alice in the i-th round. We say that $\Pi_n^{[k]}$ solves the compression game for a function h_n on n bits if $D^{(r+1,1)}$ outputs 1 on x if and only if $h_n(x) = 1$.

Finally, we let $\mathsf{cost}(\Pi_n^{[k]}) \stackrel{\text{def}}{=} s$, where $s = \sum_{i \in [r]} s_i$. We assume for convenience that the number of parties is always limited by the size of the circuits used by Alice. These definitions extend to languages, probabilistic games, and average-case games in the natural way.

3 The communication cost of $\mathsf{AC}^0[p]$-compression games

We start with a construction of single-round compression games for an arbitrary symmetric function.

Lemma 3.1. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be an arbitrary symmetric function. Then, for every $1 \le d(n) \le \log n / \log\log n$, the function f admits a single-round $\mathsf{AC}^0_d(\mathsf{poly}(n))$-compression game with communication*

$$c_d(n) = O\left((d-1)! \cdot n \cdot \left(\frac{\log\log n}{\log n}\right)^{d-1}\right).$$

In particular, for every fixed integer $d \ge 1$, we have $c_d(n) = O\big(n/(\log n)^{(d-1)-o(1)}\big)$.

Proof. Let f be a symmetric function that receives as input an n-bit string $x \in \{0,1\}^{[n]}$. We sketch the construction of depth-d circuits for the corresponding compression games. Observe that any integer $n \in \mathbb{N}$ can be represented with at most $\lceil \log(n+1) \rceil$ bits. For simplicity, we will approximate these values by $\log n$. This will be compensated by the use of asymptotic notation in the final bounds.

Observe that for $d = 1$ the result is obvious, since Alice can simply send x to Bob. For every $d \ge 2$, we design an $\mathsf{AC}^0_d(\mathsf{poly}(n))$ circuit that, on a given input x, outputs $m_d \stackrel{\text{def}}{=} (d-2)! \cdot n \cdot (\log\log n)^{d-2}/(\log n)^{d-1}$ binary strings $a_d^1, \dots, a_d^{m_d}$ of size $s_d \stackrel{\text{def}}{=} (d-1) \cdot \log\log n$, which together encode the number of 1's in x. More precisely, $|x|_1 = \sum_{i=1}^{m_d} \mathsf{dec}(a_d^i)$, where $\mathsf{dec}(a)$ denotes the integer encoded by the binary string a. Therefore, it is enough that Alice communicates in a single-round

at most $m_d \cdot s_d$ bits to Bob, which is then able to compute the original value $f(x)$. This last step relies on the assumption that f is a symmetric function.

First, we give a depth-2 circuit with these properties. Partition the n input bits into $m_2 = n/\log n$ blocks of size $t = \log n$. In other words, let $[n] = B_1 \dot\cup \ldots \dot\cup B_{m_2}$, where $|B_i| = t$. For each block B_i, there exists CNFs $\phi_1^i, \ldots, \phi_{\log\log n}^i$ of size $O(n)$ that compute the string $a_2^i \in \{0,1\}^{\log\log n} = \{0,1\}^{s_2}$ corresponding to the number of 1's in $x_{B_i} \in \{0,1\}^{B_i}$ (the projection of x to B_i). A small formula of this form exists because the number of input bits is $\log n$. Together with the previous discussion, this completes the proof for $d = 2$.

Now fix an arbitrary $d > 2$. We will construct the corresponding AC^0_d circuit by induction. It will be clear from the description that its final size is a polynomial whose leading exponent does not depend on d. Assume that there is a depth $d-1$ circuit C that outputs m_{d-1} strings $a_{d-1}^1, \ldots, a_{d-1}^{m_{d-1}}$, as described before, on any given input $x \in \{0,1\}^n$. Assume also that its top gates are AND gates. This is without loss of generality, given the argument we use below.

Recall that $a_{d-1}^i \in \{0,1\}^{s_{d-1}}$. We partition these strings into m_d sets, each containing $t \stackrel{\text{def}}{=} m_{d-1}/m_d = \log n/((d-2)\cdot\log\log n) \geq 1$ strings, given our upper bound on d. More precisely, we have $[m_{d-1}] = T_1 \dot\cup \ldots \dot\cup T_{m_d}$, where $|T_i| = t$. For convenience, let $A_i = \{a_{d-1}^j \mid j \in T_i\}$. For any a_{d-1}^j, we have $\mathsf{dec}(a_{d-1}^j) \leq 2^{s_{d-1}} = (\log n)^{d-2}$. Consequently,

$$\sum_{j \in A_i} \mathsf{dec}(a_{d-1}^j) \leq |A_i| \cdot (\log n)^{d-2} = t \cdot (\log n)^{d-2} \leq (\log n)^{d-1}.$$

In particular, this sum can be represented with $s_d = (d-1)\cdot\log\log n$ bits. Observe that the strings in A_i have, together, $t \cdot s_{d-1} = \log n$ bits. Therefore, there exists DNFs $\psi_1^i, \ldots, \psi_{s_d}^i$ of size $O(n)$ that compute the sum of the strings in A_i, which we represent as a string $a_d^i \in \{0,1\}^{s_d}$. Since this is the case for every $i \in [m_d]$, we obtain circuits $\psi^i \circ C$ computing each string a_d^i. Finally, notice that the top three layers of $\psi_j^i \circ C$ can be collapsed into a depth-2 circuit, which gives us an AC^0_d circuit for the same function. This completes the proof of Lemma 3.1. □

Notice that this upper bound comes from a very restricted class of compression games, as there is no continuing interaction with Bob. A simpler and more efficient construction can be obtained for the MOD_q functions, as for them there is no need to keep track of the exact number of 1s in the original input.

As observed by [CS12], any compression game for $\mathsf{Majority}_{2n}$ can be used to solve the compression game for Parity_n, with some overhead. In general, the same argument provides the following connection, which implies that in order to prove lower bounds for $\mathsf{Majority}$, it is sufficient to get lower bounds for MOD_q.

Proposition 3.2. *Let $h\colon \{0,1\}^n \to \{0,1\}$ be an arbitrary symmetric function, $\mathcal{C}$ be a circuit class, and $d \geq 1$. Assume that the $\mathcal{C}_d(\mathsf{poly}(n))$-compression game for $\mathsf{Majority}_n$ can be solved with cost $c(n)$ in $r(n)$ rounds. Then the $\mathcal{C}_{d+O(1)}(\mathsf{poly}(n))$-compression game for h can be solved with cost $c_h(n) = O(c(2n)\cdot\log n)$ in $r_h(n) = O(r(2n)\cdot\log n)$ rounds.*

Proof. Let Π_{2n}^{Maj} be a protocol for $\mathsf{Majority}_{2n}$. We sketch the construction of a protocol Π_n^h for h. The idea is to run Π_{2n}^{Maj} about $\log n$ times in order to obtain the hamming weight $|x|_1$ of $x \in \{0,1\}^n$, the input given to Alice in the compression game for h.

In order to achieve this, Alice runs Π_{2n}^{Maj} on appropriate inputs of the form $y = x1^k0^{n-k} \in \{0,1\}^{2n}$, where a different k is used during each stage of Π_n^h. Here a stage is simply a complete

execution of Π_{2n}^{Maj}, and Alice performs a binary search with at most $O(\log n)$ stages to obtain $|x|_1$. Although we have defined protocols with an implicit set E of accepting transcripts, observe that with an extra round we can ensure that Bob sends the correct output $\mathsf{Majority}_{2n}(y)$ to Alice.

Finally, it is enough to verify that each string y can be computed by constant-depth polynomial size circuits. However, since there are no more than $O(\log n)$ stages, and since Bob sends one bit at each stage, each string y is a function of at most $O(\log n)$ bits, and can certainly be computed by depth-two polynomial size circuits. □

For our main theorem, we will need the following result, whose proof is discussed in more detail in Section B.

Proposition 3.3 ([Raz87, Smo87], folklore)**.** *Let $p, q \geq 2$ be distinct primes. There exist fixed constants $\zeta > 0$ and $n_0 \in \mathbb{N}$ for which the following holds. For every $n \geq n_0$ and $\varepsilon(n) \in [2^{-n}, 1/10q]$, any polynomial $P \in \mathbb{F}_p[x_1, \ldots, x_n]$ that ε-approximates the MOD_q^n function with respect to the uniform distribution has degree at least $\zeta \cdot \sqrt{n \cdot \log(1/\varepsilon)}$.*

Interestingly, our argument relies on a crucial way on the approximation of Boolean circuits by polynomials with exponentially small error. For convenience of the reader, we include the proof of the next result in Section C.

Proposition 3.4 ([Raz87, Smo87, KS12])**.** *Let p be a fixed prime. There exists a constant $\alpha = \alpha(p) \in \mathbb{N}$ such that, for every $\delta \in (0, 1/2)$ and $d(n) \geq 1$, any $\mathsf{AC}_d^0[p](s(n))$ circuit C admits a δ-error probabilistic polynomial $\mathbf{Q}(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n]$ of degree at most $(\alpha \cdot \log s)^{d-1} \cdot \log(1/\delta)$. In particular, it follows that for any distribution $\mathcal{D}$ over $\{0,1\}^n$, C is δ-approximated with respect to $\mathcal{D}$ by a polynomial of degree at most $(\alpha \cdot \log s)^{d-1} \cdot \log(1/\delta)$.*

The next proposition is a minor extension of a result implicit in [CS12]. It allows us to transform an interactive compression protocol for a function into a certain Boolean circuit that computes the same function.

Proposition 3.5. *Let $c\colon \mathbb{N} \to \mathbb{N}$ be a function such that $c(n) \leq n$, $s\colon \mathbb{N} \to \mathbb{N}$ be a function with $s(n) = \Omega(n)$, $\gamma\colon \mathbb{N} \to [0, 1/2)$, L be a language, and $\mathcal{C}$ be a circuit class. If there exists an average-case $\mathcal{C}_d(\mathsf{poly}(n))$-compression game for L with cost $c(n)$ and error probability $\gamma(n)$ with respect to the uniform distribution over $\{0,1\}^n$, then there exist circuits $C_1, \ldots, C_T$ from $\mathcal{C}_{d+O(1)}(\mathsf{poly}(n))$, where $T \leq 2^{c(n)}$, such that*

$$\Pr_{x \sim \{0,1\}^n}[L(x) \neq \bigvee_{i \in [T]} C_i(x)] \;\leq\; \gamma(n).$$

Furthermore, these circuits are disjoint: $C_i^{-1}(1) \cap C_j^{-1}(1) = \emptyset$ for every pair $i, j \in [T]$ with $i \neq j$.

Proof. Let $\Pi_n = \langle C^{(1)}, \ldots, C^{(r)}, f^{(1)}, \ldots, f^{(r-1)}, E_n \rangle$ be an average-case protocol for L_n with $r(n)$ rounds and error probability $\gamma(n)$. Observe that Π_n solves the $\mathcal{C}$-compression game of some function $h_n\colon \{0,1\}^n \to \{0,1\}$, and that h_n is $\gamma(n)$-close to L_n. Recall that Π_n has a signature $\mathsf{signature}(\Pi_n) = (n, s_1, t_1, \ldots, t_{r-1}, s_r)$. For convenience, let $t \stackrel{\text{def}}{=} \sum_{i \in [r-1]} t_i$, and $s \stackrel{\text{def}}{=} c(n) = \sum_{i \in [r]} s_i$.

Given a string $w \in \{0,1\}^{s+t}$, we write $w = (w^{(A,1)}, w^{(B,1)}, \ldots, w^{(B,r-1)}, w^{(A,r)})$ as a concatenation of strings whose sizes respect the signature of Π_n. In other words, $|w^{(A,i)}| = s_i$ and $|w^{(B,j)}| = t_i$, for all $i \in [r]$ and $j \in [r-1]$. We say that w is Alice-consistent on an input x if, for every $i \in [r]$,

$w^{(A,i)} = C^{(i)}(x, w^{(A,1)}, w^{(B,1)}, \ldots, w^{(B,i-1)})$. On the other hand, we say that w is Bob-consistent if, for every $j \in [r-1]$, $w^{(B,j)} = f^{(j)}(w^{(A,1)}, \ldots, w^{(A,j-1)})$. Observe that whether a string w is Bob-consistent or not does not depend on x. Let $B_n \subseteq \{0,1\}^{t+s}$ denote the set of Bob-consistent strings. For convenience, set $W_n \stackrel{\text{def}}{=} E_n \cap B_n$.

We claim that $h(x) = 1$ if and only if there exists a string $w \in W_n$ that is Alice-consistent on x. One direction is clear, since if $h(x) = 1$ then $\mathsf{transcript}_{\Pi_n}(x) \in E_n$, and this string is both Bob-consistent and Alice-consistent on x. On the other hand, assume there exists $w \in \{0,1\}^{s+t}$ that is Bob-consistent and Alice-consistent on x. An easy induction on the number of rounds of the protocol shows that $w = \mathsf{transcript}_{\Pi_n}(x)$. Furthermore, if $w \in W_n$ then $w \in E_n$, and it must be the case that $h(x) = 1$, since Π_n is a protocol for h_n. Observe that this argument also shows that if $h(x) = 1$ then there is a unique $w \in W_n$ that serves as a certificate for x.

Notice that there are at most $2^{c(n)}$ Bob-consistent strings. This is because for every string $w^A = (w^{(A,1)}, w^{(A,2)}, \ldots, w^{(A,r)}) \in \{0,1\}^s$, there exists a unique completion of w^A by a string $w \in \{0,1\}^{s+t}$ that is Bob-consistent. In particular, $|W_n| \leq 2^{c(n)}$.

For every fixed $w \in W_n$, we claim that there exists a circuit $C_w(x)$ from $\mathcal{C}_{d+O(1)}(\mathsf{poly}(n))$ that checks if w is Alice-consistent on x. Recall that for every $i \in [r]$, $C^{(i)}$ is a circuit from $\mathcal{C}_d(\mathsf{poly}(n))$. Therefore, we can check in parallel whether $w^{(A,i)} = C^{(i)}(w^{(A,1)}, w^{(B,1)}, \ldots, w^{(B,i-1)})$, for all $i \in [r]$, using just a constant number of additional layers, since we assume throughout that $\mathcal{C}$ has unbounded fan-in AND and OR gates. which proves the claim. It follows that

$$h(x) = \bigvee_{w \in W_n} C_w(x),$$

for every $x \in \{0,1\}^n$. In addition, C_{w_1} and C_{w_2} are disjoint whenever $w_1 \neq w_2$, since exactly one $w \in W_n$ is Alice-consistent on x. Finally, recall that h_n is $\gamma(n)$-close to L_n, which completes the proof of Proposition 3.5. □

Proposition 3.5 implies that in order to prove communication lower bounds for interactive compression games, it is enough to prove circuit lower bounds of a particular form. We obtain the following result.

Lemma 3.6. *Let p and q be distinct primes, $\gamma\colon \mathbb{N} \to (0,1)$ be an arbitrary function, $k \in \mathbb{N}$, and $d = d(n) \in \mathbb{N}$. Assume that*

$$\Pr_{x \sim \{0,1\}^n}[\mathsf{MOD}_q^n(x) \neq \bigvee_{i \in [T(n)]} C_i(x)] \leq \gamma(n),$$

where each C_i is computed by an $\mathsf{AC}_d^0[p](n^k)$ circuit, and C_i and C_j are disjoint whenever $i \neq j$. Then, the following holds.

(i) $\log T(n) \geq \sqrt{n}/(\log n)^{d+O(1)}$ *if* $\gamma(n) \leq 1/20q$;

(ii) $\log T(n) \geq n/(\log n)^{2d+O(1)}$ *in the case of an exact compression game (i.e., $\gamma = 0$).*

Proof. We employ the polynomial approximation method, i.e., we show that if MOD_q^n admits a circuit of this form, then it can be approximated by a polynomial Q whose degree is upper bounded by a function depending on T. We then invoke Proposition 3.3 in order to obtain a lower bound on T. More details follow.

First, Proposition 3.4 guarantees that for any $\delta > 0$, each circuit C_i can be δ-approximated under the uniform distribution by a polynomial $Q_i \in \mathbb{F}_p[x_1, \ldots, x_n]$ of degree at most $(\ell \cdot \log n)^d \cdot \log(1/\delta)$, where ℓ is a fixed positive constant. We let $\delta \stackrel{\text{def}}{=} \varepsilon/T$, where $\varepsilon = \varepsilon(n)$ will be set conveniently later in the proof. Now let

$$Q(x) \stackrel{\text{def}}{=} \sum_{i \in [T]} Q_i(x).$$

We claim that $Q \in \mathbb{F}_p[x_1, \ldots, x_n]$ is a polynomial that $(\varepsilon + \gamma)$-approximates MOD_q^n under the uniform distribution. Clearly,

$$\begin{aligned} \Pr_{x \sim \{0,1\}^n}[\mathsf{MOD}_q^n(x) \neq Q(x)] &\leq \Pr\Big[\mathsf{MOD}_q^n(x) \neq \bigvee_{i \in [T(n)]} C_i(x)\Big] + \Pr\Big[\bigvee_{i \in [T(n)]} C_i(x) \neq Q(x)\Big] \\ &\leq \gamma + \Big(1 - \Pr\Big[\bigvee_{i \in [T(n)]} C_i(x) = Q(x)\Big]\Big). \end{aligned}$$

For each $i \in [T]$, let $S_i \stackrel{\text{def}}{=} \{x \in \{0,1\}^n \mid Q_i(x) \neq C_i(x)\}$ be the set of bad inputs for Q_i, and set $S \stackrel{\text{def}}{=} \bigcup_{i \in [T]} S_i$. In order to complete the proof of our claim, we argue next that for every $y \notin S$, $Q(y) = \bigvee_{i \in [T(n)]} C_i(y)$.

First, if $\bigvee_{i \in [T(n)]} C_i(y) = 0$, then $Q_i(y) = 0$ for every $i \in [T]$, and we get $Q(y) = 0$. On the other hand, if $\bigvee_{i \in [T(n)]} C_i(y) = 1$, using the disjointness assumption for the family of circuits, it follows that there is exactly one circuit with $C_i(y) = 1$. Since $y \notin S$, we get that $Q_i(y) = 1$, while $Q_j(y) = 0$ for every $j \neq i$. Consequently, we have $Q(y) = 1$. (Observe that the extra assumption over the family of circuits is crucial for this case, since the original circuits produce Boolean values, while Q is an $\mathbb{F}_p$-polynomial.) Overall, it follows that $\Pr[\bigvee_{i \in [T(n)]} C_i(x) = Q(x)] \geq (2^n - |S|) \cdot 2^{-n} \geq 1 - T \cdot \delta = 1 - \varepsilon$, which establishes our initial claim.

Therefore, for every $\varepsilon(n) > 0$, there exists a polynomial $Q \in \mathbb{F}_p[x_1, \ldots, x_n]$ that $(\varepsilon + \gamma)$-approximates the MOD_q^n function over the uniform distribution, where

$$\mathsf{deg}(Q) \leq ((\ell \cdot \log n)^d \cdot \log(1/\delta)) \leq (\ell \cdot \log n)^d \cdot (\log T + \log(1/\varepsilon)). \tag{1}$$

On the other hand, we obtain from Proposition 3.3 that for every $\varepsilon(n) \in [2^{-n}, 1/10q]$, and every large enough n,

$$\zeta \cdot \sqrt{n \cdot \log(1/(\varepsilon + \gamma))} \leq \mathsf{deg}(Q). \tag{2}$$

Our result follows by combining Equations 1 and 2. Observe that we are free to set $\varepsilon(n)$ in order to maximize our lower bound on T, depending on the value of γ. If $0 < \gamma \leq 1/20q$, the first case of Lemma 3.6 follows if we let $\varepsilon = 1/20q$. On the other hand, when $\gamma = 0$, we get that

$$\log T(n) \geq \frac{\zeta \cdot \sqrt{n \cdot \log(1/\varepsilon)} - \log(1/\varepsilon) \cdot (\ell \cdot \log n)^d}{(\ell \cdot \log n)^d},$$

and the second case of Lemma 3.6 now follows by setting $\varepsilon = \exp(-\Theta(n/\log^{2d} n))$. □

We are now ready to prove an essentially optimal communication lower bound for $\mathsf{AC}_d^0[p]$-compression games for $\mathsf{Majority}$.

Theorem 3.7. *Let p be a prime number. There exists a constant $c \in \mathbb{N}$ such that, for any $d \in \mathbb{N}$, and every $n \in \mathbb{N}$ sufficiently large, the following holds.*

(*i*) *Any* $\mathsf{AC}^0_d[p]$*-compression game for* $\mathsf{Majority}_n$ *(with any number of rounds) has communication cost at least* $n/(\log n)^{2d+c}$.

(*ii*) *There exists a single-round* AC^0_d*-compression game for* $\mathsf{Majority}_n$ *with communication cost at most* $n/(\log n)^{d-c}$.

Proof. The lower bound follows immediately from Proposition 3.2, Proposition 3.5, and Lemma 3.6 (*ii*). The upper bound is given by Lemma 3.1. □

For randomized compression games, we are able to generalize the lower bound for single-round protocols obtained by Chattopadhyay and Santhanam [CS12] to protocols with any number of rounds.

Theorem 3.8. *Let* p *and* q *be distinct primes. There exists a constant* $c \in \mathbb{N}$ *such that, for any* $d \in \mathbb{N}$*, and* $n \in \mathbb{N}$ *sufficiently large, every randomized* $\mathsf{AC}^0_d[p]$*-compression game for* MOD^n_q *with any number of rounds and error at most* $1/3$ *has communication cost at least* $\sqrt{n}/(\log n)^{d+c}$.

Proof. If there exists a randomized compression protocol with these properties, we can boost its success probability to $1 - 1/20q$ on every input by repeating it a constant number of times, and applying a majority vote. Observe that the communication increases by a constant factor only, and that the majority vote can be computed efficiently, as it is over a constant number of bits. Since any randomized protocol with this success probability provides an average-case protocol that is correct on at least a $(1 - 1/20q)$-fraction of the inputs under the uniform distribution, the result follows from Proposition 3.5 and Lemma 3.6 (*i*). □

We stress that the results in Theorems 3.7 and 3.8 hold both for $\mathsf{Majority}$ and MOD_q, but we restricted each statement to a particular function for simplicity. In order to see this, first notice that the proof of Theorem 3.7 includes the argument for MOD_q. On the other hand, in order to extend Theorem 3.8 to $\mathsf{Majority}$, we can employ a reduction through Proposition 3.2. A subtle point is that for probabilistic protocols one has to make sure that the final error probability after the reduction is bounded. However, this can be achieved during the proof by boosting the correctness probability of the initial protocol for $\mathsf{Majority}$ via repetition.

The proof of Theorem 3.7 can be generalized to an essentially optimal bound for $\mathsf{AC}^0_d[p](s(n))$-compression games computing MOD^n_q. The argument implies that this function has communication cost $n/(\log s)^{\Theta(d)}$. Observe that the original circuit size lower bounds obtained by Razborov [Raz87] and Smolensky [Smo87] follows from the analysis of communication protocols for $\mathsf{Majority}$ and MOD_q with constant communication cost. Interestingly, the polynomial method interpolates between essentially optimal communication lower bounds and circuit size lower bounds when applied with exponentially small error and constant error, respectively.

4 Multiparty Interactive Compression

4.1 The communication cost of k-party $\mathsf{AC}^0[p]$-compression games

We will prove in this section that $\mathsf{Majority}_n$ requires $\tilde{\Omega}(n^{1/2r})$ communication in the $(k+1)$-party r-round $\mathsf{AC}^0[p]$-compression game, for any $k = \mathsf{poly}(n)$. Put another way, although Alice is allowed to send roughly $n^{1/2r}$ bits to each individual Bob, even if n^{100} such parties are present, she will not be able to combine their answers in order to compute $\mathsf{Majority}_n$.

We start with the following upper bound, which can be seen as the corresponding analogue of Lemma 3.1.

Lemma 4.1. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be an arbitrary symmetric function, and p be any prime. For any $r \in \mathbb{N}$, f admits an $(\lceil n^{1/r} \rceil + 1)$-party r-round AC^0-compression game with cost $O(rn^{1/r} \log(n))$.*

Proof. We set up some notation first. Given n and r, let $T_{n,r}$ be the complete $\lceil n^{1/r} \rceil$-ary tree of depth r. We assume the leaves of $T_{n,r}$ to be ordered from left to right. Given an input x of length n, label the leaves of $T_{n,r}$ with bits of x in the natural way: the leftmost leaf is labelled with the first bit of x, the second to leftmost with the second bit, etc. Note that some leaves may remain unlabelled in this process.

Let V_d be the set of nodes at depth d in this tree, where $0 \leq d \leq r$. The protocol will proceed with Alice iteratively labelling nodes in the tree with numbers in $[n]$, each node being labelled with the sum of all the leaves in the subtree rooted at the node. Any unlabelled leaf is assumed to have label 0. After round i, where $0 \leq i \leq r$, all nodes at depth $r - i$ or greater will be labelled. Once the root is labelled, Alice can compute $f(x)$ by herself, as $f(x)$ is purely a function of the label at the root (which is the weight of the input x), and any function of $O(\log n)$ bits can be computed in AC^0_2.

We assume inductively that after round i, all nodes at depth $r-i$ or greater have been labelled. The base case $i = 0$ clearly holds, as Alice can label the leaves herself. Assume that the inductive hypothesis holds after round i, where $0 \leq i < r$. We show it holds after round $i+1$. In round $i+1$, Alice arbitrarily associates a unique Bob with each node $v \in V_{r-i-1}$. This can be done as long as the number of parties is greater than $\lceil n^{1/r} \rceil$, as assumed. We denote the Bob associated with v by $\mathrm{Bob}(v)$. For each v, Alice sends to $\mathrm{Bob}(v)$ the sequence of labels of the children of v. Note that by the inductive assumption, the children of v have already been labelled. For each v, $\mathrm{Bob}(v)$ responds with the sum of all the integer labels sent by Alice to $\mathrm{Bob}(v)$ in the $(i+1)$-th round.

This is clearly a correct protocol. In any one round, Alice sends at most $\lceil n^{1/r} \rceil \cdot \lceil \log(n+1) \rceil$ bits to any Bob, as the number of children of any node in the tree is at most $\lceil n^{1/r} \rceil$, and each labelled node has a label in $[n]$. Thus, the cost of the protocol is $O(rn^{1/r} \log n)$, as claimed. □

Our lower bound is also based on algebraic arguments, but it employs a slightly different approach to that in the previous section. In particular, it does not rely on Proposition 3.5. We will need the following result.

Proposition 4.2 ([Raz87])**.** *Let p be a fixed prime, and $P(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a degree-ℓ polynomial. Then,*

$$\Pr_{x \sim \{0,1\}^n}[\mathsf{Majority}_n(x) = P(x)] \leq 1/2 + O(\ell/\sqrt{n}).$$

The next lemma allows us to construct low-degree probabilistic polynomials from multiparty compression games.

Lemma 4.3. *Let $\Phi_n^{[k]}$ be a randomized $(k+1)$-party r-round $\mathsf{AC}^0_d[p](\mathsf{poly}(n))$-compression protocol with signature $(n, s_1, t_1, \ldots, s_r, t_r)$ computing a Boolean function $h\colon \{0,1\}^n \to \{0,1\}$ with error γ, where $s_i \leq n$ for each $i \in [r]$, and $r \in \mathbb{N}$. Then, for every $\delta > 0$, h admits a $(\gamma + \delta)$-error probabilistic polynomial over $\mathbb{F}_p$ with degree $O\big((\sum_{i \in [r]} s_i)^r \cdot ((\log n)^{d+r} \cdot (\log 1/\delta))^{r+1}\big)$.*

Proof. We start with a proof of the lemma for $r = 1$ and deterministic protocols that are always correct, then observe that the same proof can be generalized to randomized r-round protocols.

Suppose $\Phi_n^{[k]}$ is a $(k+1)$-party 1-round $\mathsf{AC}^0_d[p](\mathsf{poly}(n))$-compression protocol with signature (n, s_1, t_1) for a Boolean function h on inputs x of n bits. For each $i \in [k]$, let $a_1^i \dots a_{n_i}^i$ be the message bits sent by Alice to Bob_i in the first round, and let $b_1^i \dots b_{m_i}^i$ be Bob_i's response. Let a be the bit output by Alice at the conclusion of the protocol. By the definition of signature, we have that for each $i \in [k]$, $n_i \le s_1$ and $m_i \le t_1$. We also have that $a = 1$ if and only if $h(x) = 1$.

Each of the message bits sent by Alice in the first round is a function of x, and since Alice is $\mathsf{AC}^0_d[p](\mathsf{poly}(n))$-bounded, we can use Proposition 3.4 to obtain ε-error probabilistic polynomials $P_j^i \in \mathbb{F}_p[x_1, \dots, x_n]$, where $i \in [k], j \in [n_i]$, for each of these message bits. The degree of each polynomial is at most $d_1 = O((\log n)^{d-1} \cdot \log 1/\varepsilon)$, where $\varepsilon > 0$ is a parameter to be determined later. Since each message bit of each Bob_i is a function of the message bits sent by Alice to Bob_i, we can express each bit b_j^i of Bob_i as an *exact* polynomial Q_j^i in the message bits of Alice. Notice that each such polynomial has degree at most s_1. Now, again by Proposition 3.4, there is an ε-error probabilistic polynomial P of degree at most $d_2 = O((\log n)^{d-1} \cdot \log 1/\varepsilon)$ for a as a function of x, the message bits sent by Alice in the first round, and the message bits sent by each Bob in the first round.

If we set $\varepsilon = \delta/(s_1 \cdot k + 1)$, by using the union bound, we have that

$$P' \stackrel{\text{def}}{=} P(x, P_1^1(x), \dots, P_{n_k}^k(x), Q_1^1(P_1^1(x), \dots, P_{n_1}^1(x)), \dots, Q_{m_k}^k(P_1^k(x), \dots, P_{n_k}^k(x)))$$

is a δ-error probabilistic polynomial for h as a function of x. The degree of P' is at most $d_1 \cdot s_1 \cdot d_2 = O(s_1 \cdot ((\log n)^d \cdot \log 1/\delta)^2)$, where we have used that $\log 1/\varepsilon = O(\log n \cdot \log 1/\delta)$ due to the upper bound on s_1 and $k \le \mathsf{poly}(n)$. This completes the proof for (deterministic) single-round protocols.

The proof for deterministic protocols with $r \ge 2$ rounds is by induction on the number of rounds. Let $\Phi_n^{[k]}$ be a $(k+1)$-party r-round $\mathsf{AC}^0_d[p](\mathsf{poly}(n))$-compression protocol with signature $(n, s_1, t_1, \dots, s_r, t_r)$ for a Boolean function h. Observe that during the last round of the protocol, each Bob_ℓ receives a message containing at most $s \stackrel{\text{def}}{=} \sum_{i \in [r]} s_i$ bits (recall that Bob_ℓ has access to the messages he received from Alice in previous rounds, and to no other message). We can view each bit a_j^ℓ of each such message as a Boolean function computed by a $(k+1)$-party $(r-1)$-round protocol, where $\ell \in [k]$, and $j \le s$. It follows from the induction hypothesis that there is a probabilistic polynomial $P_j^\ell \in \mathbb{F}_p[z_1, \dots, z_{s'}]$ for an appropriate $s' \le s$ of degree at most

$$d_1 \;\le\; O\big(s^{r-1} \cdot ((\log n)^{d+(r-1)} \cdot (\log 1/\varepsilon))^r\big)$$

that ε-approximates a_j^ℓ, where $\varepsilon > 0$ will be set conveniently later in the proof.[2] Further, during the last round of the protocol, each bit b_j^ℓ sent by Bob_ℓ can be computed exactly by a (deterministic) polynomial Q_j^ℓ of degree at most s. Finally, the last bit output by Alice during the execution of $\Phi_n^{[k]}$ is computed by an $\mathsf{AC}^0_d[p]$ circuit over polynomially many input bits. According to Proposition 3.4, it can be ε-approximated by a probabilistic polynomial $P \in \mathbb{F}_p[y_1, \dots, y_{\mathsf{poly}(n)}]$ of degree $d_2 \le O((\log n)^{d-1} \cdot \log 1/\varepsilon)$.

We now compose these polynomials appropriately, similarly to the base case, in order to obtain a probabilistic polynomial $P' \in \mathbb{F}_p[x_1, \dots, x_n]$ that approximates the original Boolean function h

[2] Our abuse of the asymptotic notation in this inductive proof is harmless, as we are proving the result for a fixed number of rounds only.

compressed by $\Phi_n^{[k]}$. If we set $\varepsilon \stackrel{\text{def}}{=} \delta/(sk+1) = \delta/\mathsf{poly}(n)$, we get via an union bound that P' is a probabilistic polynomial that δ-approximates h. Finally, the degree of P' is upper bounded by

$$\begin{aligned} d_1 \cdot s \cdot d_2 &\leq O\big(s^{r-1} \cdot ((\log n)^{d+(r-1)} \cdot (\log 1/\varepsilon))^r \cdot s \cdot (\log n)^{d-1} \cdot \log 1/\varepsilon\big) \\ &\leq O\big(s^r \cdot ((\log n)^{d+r} \cdot (\log 1/\delta))^r \cdot (\log n)^d \cdot \log 1/\delta\big) \\ &\leq O\big((\Sigma_{i\in[r]} s_i)^r \cdot ((\log n)^{d+r} \cdot (\log 1/\delta))^{r+1}\big), \end{aligned}$$

which completes the induction step.

It remains to handle the case of randomized protocols. Observe that for every fixed setting of the randomness of Alice, we obtain a multiparty compression protocol computing some Boolean function h_r. We can apply the procedure described above to get a probabilistic polynomial $P_r \in \mathbb{F}_p[x_1, \ldots, x_n]$ that agrees with h_r on every input $x \in \{0,1\}^n$ except with probability δ. Since over the choice of r we know that $h(x) = h_r(x)$ except with probability γ, we can obtain from the family of distributions P_r a single distribution over polynomials of the same degree that agrees with h on every input x except with probability $\gamma + \delta$, which completes the proof. □

We now have all ingredients to prove the main result of this section.

Theorem 4.4. *Let $p \in \mathbb{N}$ be a fixed prime. For every $k, r, d \in \mathbb{N}$, the following holds.*

(*i*) *There exists a deterministic $n^{1/r}$-party r-round $\mathsf{AC}^0[p]$-compression game for $\mathsf{Majority}_n$ with cost $O(n^{1/r} \cdot \log n)$.*

(*ii*) *Every randomized n^k-party r-round $\mathsf{AC}^0_d[p]$-compression game for $\mathsf{Majority}_n$ has cost $\Omega\big(n^{1/2r}/(\log n)^{2(d+r)}\big)$.*

Proof. The upper bound follows from Lemma 4.1. For the lower bound, assume $\Pi_n^{[k]}$ has signature $(n, s_1, t_1, \ldots, s_r, t_r)$ and satisfies the assumption of the theorem. Since $\Pi_n^{[k]}$ is a randomized protocol, we can reduce its error probability to $1/20$ by running it in parallel and computing a majority vote during the last round. Observe that the depth of the circuits used by Alice increases by at most 1 if this computation is performed by an appropriate DNF or CNF. Setting $\delta = 1/20$ in Lemma 4.3 and fixing the randomness, we can obtain an average-case (deterministic) polynomial for $\mathsf{Majority}_n$ of the stated degree and error $1/10$ with respect to the uniform distribution. Now applying Proposition 4.2 and using $1/\delta = O(1)$, we get that

$$(s_1 + s_2 + \ldots + s_r)^r \cdot (\log n)^{(d+r)(r+1)} \;\geq\; \Omega(\sqrt{n}),$$

which completes the proof of the lower bound, since $\mathsf{cost}(\Pi_n^{[k]}) = \sum_{i\in[r]} s_i$ and $r \geq 1$. □

As opposed to the statement of Theorem 3.7, we have not tried to optimize the logarithmic factors here, since there is still a polynomial gap in the bounds as a function of r.[3]

Corollary 4.5. *For any $r, \ell, d \in \mathbb{N}$, the randomized n^ℓ-party r-round $\mathsf{AC}^0_d[p]$-compression cost of $\mathsf{Majority}_n$ is $n^{\Theta(1/r)}$.*

[3]For instance, in the proof of Lemma 4.1, it is possible to break the information passed to each Bob into multiple blocks as done in the proof of Lemma 3.1, and save an extra $(\log n)^{\Theta(d)}$ factor during each round by allowing Alice to make partial progress towards the computation of $\mathsf{Majority}$.

In addition, observe that Theorem 4.4 implies a round separation result for *multiparty* $\mathsf{AC}^0[p]$-compression games. In particular, we get the following consequence for single-round $\mathsf{AC}^0[p]$ protocols versus protocols with more rounds.

Corollary 4.6. *For every $\varepsilon > 0$ and $\ell \in \mathbb{N}$, there exists $r \in \mathbb{N}$ with $r = O(1/\varepsilon)$ for which the following holds, whenever n is sufficiently large. There exists an explicit function $f_n\colon \{0,1\}^n \to \{0,1\}$ such that: f_n admits no randomized n^ℓ-party single-round $\mathsf{AC}^0[p]$-compression games with cost $n^{1/2-\varepsilon}$, but it admits deterministic n^ε-party r-round $\mathsf{AC}^0[p]$-compression games of cost n^ε.*

4.2 Randomized versus deterministic games

Note that for two-party games we were able to obtain almost linear lower bounds for *deterministic* protocols (Theorem 3.7), while for probabilistic and average-case protocols we encountered a barrier at $c(n) \approx \sqrt{n}$ (Theorems 3.8 and 4.4). We are not aware of explicit lower bounds of the form $n^{1/2+\varepsilon}$ for a fixed $\varepsilon > 0$ for randomized two-party $\mathsf{AC}^0[p]$ games. It is natural to wonder if we can improve Theorem 4.4 in the case of *deterministic* k-party games.

We prove next that this is unlikely without the introduction of new ideas to handle probabilistic protocols. More precisely, we observe that k-party protocols can be derandomized without increasing communication cost. The proof relies on the definition of cost for such protocols as the length of the longest message sent by Alice to any particular Bob, and on the fact that we are dealing with non-uniform protocols/circuits. The argument is based on parallel repetition and composition of k-party protocols with an approximate majority function. We provide the details next.

We say that a Boolean function $h_n\colon \{0,1\}^n \to \{0,1\}$ is an (ℓ_1, ℓ_2)-approximate majority if $h_n(x) = 0$ on every x with $|x|_1 \le \ell_1$, and $h_n(x) = 1$ on every x with $|x|_1 \ge \ell_2$.

Proposition 4.7 ([ABO84])**.** *There exists a family $h = \{h_n\}_{n\in\mathbb{N}}$ of Boolean functions in $\mathsf{AC}^0_3(\mathsf{poly}(n))$ for which every h_n is an $(0.49n, 0.51n)$-approximate majority.*

Theorem 4.8. *Let $\mathcal{C}$ be a circuit class, $d \ge 1$, and $f = \{f_n\}_{n\in\mathbb{N}}$ be a family of Boolean functions, where $f_n\colon \{0,1\}^n \to \{0,1\}$. Suppose f admits a k-party probabilistic $\mathcal{C}_d(\mathsf{poly}(n))$-compression game with cost $c(n)$ and error $\gamma(n) \le 1/3$, where $k = O(\mathsf{poly}(n))$. Then f admits a k'-party deterministic $\mathcal{C}_{d+O(1)}(\mathsf{poly}(n))$-compression game with the same cost $c(n)$ and $k' = O(\mathsf{poly}(n))$.*

Proof. By assumption, f has a k-party probabilistic $\mathcal{C}_d(\mathsf{poly}(n))$-compression protocol Π with cost $c(n)$ and error $\gamma(n) \le 1/3$, where $k = O(\mathsf{poly}(n))$. We define a new probabilistic protocol for f with the same cost but with $k' \stackrel{\text{def}}{=} \ell n \cdot k$ parties and with error $\gamma'(n) < 2^{-n}$, where $\ell > 0$ is a constant which we determine later. We then use Adleman's trick to fix the random bits used by Alice, thus making the protocol deterministic.

The new probabilistic protocol Π' for f simply simulates ℓn copies of the protocol Π in parallel. Namely, we interpret the Bobs to be partitioned into ℓn sets, each of size k, and Alice independently executes the protocol in parallel for each set of Bobs. Note that by our definition of cost, the cost for each round of Π' is the same as the cost for each round of Π. In the final step of the protocol, Π' applies the Approximate Majority function $h_{\ell n}$ to the answers of Π for the ℓn parallel executions. Using Proposition 4.7, Alice can be implemented to work in $\mathcal{C}_{d+O(1)}(\mathsf{poly}(n))$. It follows by a standard application of Proposition A.1 that if we set ℓ to be a large enough constant, the error probability of the new protocol Π' is strictly less than 2^{-n}.

Now, there must exist some setting of the random bits of Alice that yields the correct answer for *every* $x \in \{0,1\}^n$, simply by using the union bound. By fixing the random bits of Alice accordingly, we derive a *deterministic* protocol with cost $c(n)$, which completes the proof. □

5 The connection with circuits augmented with oracle gates

In this section we observe that lower bounds on interactive compressibility are closely connected to lower bounds against oracle circuits with arbitrary oracles. We first show such a connection for 2-party compression games, and then for multiparty compression games.

In order to formalize these connections, we need to define classes of oracle circuits corresponding to classes of Boolean circuits. Such a definition is especially non-obvious for bounded-depth circuit classes – should we consider oracle gates when counting the depth or not? We use a very generous notion of oracle circuits. We say that an oracle circuit C belongs to the oracle analogue of a Boolean circuit class $\mathcal{C}$ if every maximal subcircuit of C without oracle gates belongs to $\mathcal{C}$. Put another way, every subcircuit induced by a connected subgraph of the acyclic graph encoding C that does not contain an oracle gate is a circuit from $\mathcal{C}$. The generosity of this notion only makes the lower bounds we derive from the connections below stronger.

For the sake of convenience, we abuse notation and occasionally use $\mathcal{C}$ to refer both to a Boolean circuit class and its oracle analogue.

Proposition 5.1. *Let $\mathcal{C}$ be a circuit class. Let C be an oracle circuit over n variables from $\mathcal{C}(\mathsf{poly}(n))$ with oracle gates $f_i \colon \{0,1\}^{s_i} \to \{0,1\}^{t_i}$, where $i \in [r]$, for some $r = r(n)$. In addition, let $s = s_1 + \ldots + s_r$ be the total fan-in of these oracle gates, and $h \colon \{0,1\}^n \to \{0,1\}$ be the Boolean function computed by C. Then h admits a $\mathcal{C}(\mathsf{poly}(n))$-compression game with communication cost $c(n) \leq s + 1$ consisting of at most $r + 1$ rounds.*

Proof. We describe a protocol for the compression game for h in which Alice sends at most $s + 1$ bits to Bob, and where each of Alice's messages is computable by a small circuit from $\mathcal{C}$.

First Alice topologically sorts the circuit C with respect to oracle gates, namely she constructs a graph G whose nodes are the oracle gates of the circuit, and there is an edge from a node u to a node v if and only if there is a path from the oracle gate represented by u to the oracle gate represented by v in the digraph C. The graph G is a DAG, and hence its vertices can be topologically sorted. Let $g_1, g_2 \ldots g_r$ be the topological ordering of the oracle gates. Alice proceeds inductively as follows. In round i, where $i \in [r]$, she computes all inputs to the gate g_i using her input x and previous messages sent by Bob. She then sends the values of these input bits to Bob, who in turn computes the value of the gate g_i applied to these bits, and sends her the answer. Note that g_1 has no predecessors which are oracle gates, and therefore Alice can compute all the inputs to g_1 herself using circuits from $\mathcal{C}$ (which are sub-circuits of C) applied to the input x. Gate g_i only has gates $g_1 \ldots g_{i-1}$ as predecessors, and by the definition of the protocol, Alice has already received the values of these gates from Bob in previous rounds, hence she can calculate values of inputs to g_i from x and previous messages using circuits from $\mathcal{C}$. In round $r + 1$, Alice computes the value of the circuit C on x and sends it to Bob, thus completing the protocol.

The total number of bits sent by Alice to Bob is the total fan-in of the oracle gates plus one, i.e., $s + 1$, and there are $r + 1$ rounds in the protocol.

□

Note that Proposition 5.1 only gives useful information when the total fan-in of oracle gates is sub-linear. We'd like to also show lower bounds on oracle gates where the total fan-in is not bounded in this way. This is where multiparty compression games, and the modified notion of protocol cost for such games, come in useful.

We need some more terminology for oracle circuits. An oracle circuit C has r layers if the oracle gates can be partitioned into r sets such that no two gates within any set are connected by a path in C. Equivalently, there are at most r oracle gates on any path from an input of C to the output.

Proposition 5.2. *Let D be an oracle circuit over n variables from $\mathcal{C}(\mathsf{poly}(n))$ augmented with r layers of oracle gates, where for each $i \in [r]$, s_i is the maximum fan-in of a gate in the i-th layer, and where there are at most k gates in each layer. Let $s = \sum_{i\in[r]} s_i$. In addition, let $h\colon \{0,1\}^n \to \{0,1\}$ be the Boolean function computed by D. Then h admits a $(k+1)$-party $\mathcal{C}(\mathsf{poly}(n))$-compression game with r rounds and communication cost $c(n) \leq s$.*

Proof. Alice orders the layers of oracle gates topologically, so that there are no paths from gates in layer i to gates in layer j for $i > j$. The protocol proceeds with Alice inductively computing all input bis to oracle gates in the i-th layer, where $i \in [r]$, and then delegating the computations of gates in the i-th layer to the Bobs, a different Bob for each oracle gate. Since there are at most k gates in each such layer, she can successfully assign a different Bob to each oracle gate in any specific layer. Alice can compute all inputs to an oracle gate in the first layer by herself, as all of these can be computed by circuits in $\mathcal{C}(\mathsf{poly}(n))$. In the i-th round, where $i \in [r]$, Alice chooses a different Bob for each oracle gate in layer i, and sends to the corresponding Bob the values of the inputs to the corresponding gate. She can compute these values using circuits in $\mathcal{C}$, as the output bits of all oracle gates in layer $i-1$ or below are already known to her by the definition of the protocol. The Bob corresponding to a gate responds with the output values of that gate. After the r-th round, Alice computes the output value of the circuit C, and outputs it.

Notice that Alice sends at most s_i bits to any individual Bob in round i by our assumption on the fan-in of oracle gates in C. Thus the cost of the protocol is s. It is clear that the protocol operates in r rounds. □

Observe that Propositions 5.1 and 5.2, together with Theorems 3.7 and 4.4, imply strong limitations on the progress that $\mathsf{AC}^0[p]$ circuits can make towards the goal of computing the Majority function. In particular, a circuit of this form extended with arbitrary oracle gates can only compute $\mathsf{Majority}_n$ if it delegates essentially all the work to these extra gates. We can formalize this claim as follows.

Corollary 5.3. *Let $p \geq 2$ be prime, and $d \in \mathbb{N}$. There exists a constant $c \in \mathbb{N}$ such that, for every sufficiently large n, the following holds. If $\mathsf{Majority}_n$ is computed by $\mathsf{AC}^0_d[p]$ circuits of polynomial size with arbitrary oracle gates, then the total fan-in of the oracle gates is at least $n/(\log n)^{2d+c}$.*

Proof. This result follows immediately from Proposition 5.1 and Theorem 3.7. The fan-in lower bound is independent of the number of oracle gates, as Theorem 3.7 holds for protocols with any number of rounds. □

This result has an interesting consequence on the structure of $\mathsf{AC}^0[p]$ circuits computing $\mathsf{Majority}$. More precisely, Corollary 5.3 implies that in any layered circuit computing $\mathsf{Majority}_n$, at least $\lfloor n/(\log n)^{O(k)} \rfloor$ gates must be present at the k-th layer of the circuit (in order to see this, transform the circuit into an equivalent circuit with a single oracle gate at the top after the first k layers).

On the other hand, the construction in Lemma 3.1 shows that this bound is not far from optimal. A similar consequence holds for polynomial size circuits computing the MOD_q function.

Using Proposition 5.2 and Theorem 4.4, we derive lower bounds on the maximum fan-in of oracle gates in oracle circuits with a bounded number of such layers computing $\mathsf{Majority}$. The *number* of oracle gates is now allowed to be polynomially large.

Corollary 5.4. *Let $p \geq 2$ be prime, and $r, d \in \mathbb{N}$. If $\mathsf{Majority}_n$ is computed by an $\mathsf{AC}^0_d[p]$ circuit of polynomial size with arbitrary oracle gates that contains at most r layers of such gates, then there is some oracle gate with fan-in at least $n^{1/2r}/\mathsf{polylog}(n)$.*

Proposition 5.2 suggests an approach to the NP vs. $\mathsf{NC}^1/\mathsf{poly}$ problem. The key observation is that for any r, every Boolean function in $\mathsf{NC}^1/\mathsf{poly}$ has oracle circuits of polynomial size with r layers, where the maximum fan-in of any oracle gate is $n^{O(1/r)}$.

Proposition 5.5. *Let $f = \{f_n\}_{n\in\mathbb{N}}$ be a family of Boolean functions in $\mathsf{NC}^1/\mathsf{poly}$, and $r \in \mathbb{N}$. Then f has AC^0 oracle circuits of polynomial size with r layers, where the maximum fan-in of any oracle gate is $n^{O(1/r)}$.*

Proof. Let $\{C_n\}_{n\in\mathbb{N}}$ be a sequence of circuits for f, where each C_n has size at most n^k and depth at most $c \log n$, for fixed constants k and c. We define oracle circuits D_n as follows. Divide C_n into r equally spaced layers of gates, with the distance between any two layers being at most $(c/r)\log n$. Replace each node at a layer boundary by an oracle gate whose inputs are its predecessors on the previous layer boundary. Note that any oracle gate has at most $n^{c/r}$ inputs, since the circuit has bounded fan-in. There are clearly a polynomially bounded number of oracle gates. Also, the circuit is an AC^0 circuit, since it consists purely of inputs and oracle gates. □

Applying Proposition 5.2 yields the following corollary.

Corollary 5.6. *Let r be any positive integer. Every function in $\mathsf{NC}^1/\mathsf{poly}$ admits $\mathsf{poly}(n)$-party $\mathsf{AC}^0(\mathsf{poly}(n))$-compression games with r rounds and cost $n^{O(1/r)}$.*

Thus a stronger lower bound than in Corollary 5.4 for an explicit function in NP would imply a separation of NP and $\mathsf{NC}^1/\mathsf{poly}$. We conjecture that Clique is such a function.

6 Interactive Compression versus Computation

The results of this paper and in [CS12] show that two important techniques in circuit complexity, namely, random restrictions and approximation by low-degree polynomials, can be used to prove strong incompressibility lower bounds. It is natural to wonder if other important lower bounds in complexity theory can be extended in a similar way. A related problem is whether compression can be easier than exact computation. Our next result sheds more light into these questions.

Let $\mathsf{IP}_n \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be the Inner Product function. In other words, for $x, y \in \{0,1\}^n$, $\mathsf{IP}_n(x,y) \stackrel{\text{def}}{=} \sum_{i\in[n]} x_i \cdot y_i \pmod 2$. It is known that $\mathsf{IP}_n \notin \mathsf{THR} \circ \mathsf{MAJ}$, i.e., this function cannot be computed by polynomial size circuits consisting of a bottom layer of linear threshold functions with polynomial weights, connected to a top gate computed by an arbitrary linear threshold function ([For02, FKL+01]).[4]

[4]Recall that a function $f \colon \{0,1\}^n \to \{0,1\}$ is a linear threshold function if there exist weights $w_1, \ldots, w_n \in \mathbb{Z}$ and a threshold $\theta \in \mathbb{Z}$ such that $f(x) = \mathsf{sign}(\sum_{i\in[n]} w_i \cdot x_i - \theta)$.

We observe below that IP_n admits a $(\mathsf{MAJ} \circ \mathsf{MAJ})$-compression game with communication cost $O(\log n)$. In other words, there is a natural Boolean function that cannot be computed by certain circuits, but whose computation becomes feasible if Alice is allowed to interact with a more powerful party.

Proposition 6.1. *Let $\mathsf{IP} = \{\mathsf{IP}_n\}_{n\in\mathbb{N}}$ be the family of Inner Product functions. There exists a $(\mathsf{MAJ} \circ \mathsf{MAJ})$-compression game for IP with communication cost $c(n) = O(\log n)$.*

Proof. The protocol consists of $O(\log n)$ rounds, where in each round Alice sends a single bit, and Bob replies with a string $v \in \{0,1\}^n$. After the last round, Bob knows the sum $\sum_{i\in[n]} x_i \cdot y_i$, and therefore the transcript reveals the value $\mathsf{IP}_n(x, y)$. More details follow.

Alice's circuits are of the form $C(x, y, v)$. In the first layer of the circuit, C computes $z_i \stackrel{\text{def}}{=} x_i \wedge y_i$, for every $i \in [n]$. In the second layer, C outputs $\mathsf{sign}(\sum_{i\in[n]} z_i - v_i)$. Put another way, Alice uses the same circuit in every round, and we assume that the first bit sent by Alice during the first round is discarded. Bob does all the work, and simulates a binary search by sending to Alice an appropriate string v during each round. For instance, Bob sends $v = 0^{n/2}1^{n/2}$ during the first round, and the next bit computed by Alice reveals if $\sum_{i\in[n]} x_i \cdot y_i$ is at least $n/2$. After each round, Bob sends a string corresponding to the next step of the binary search, and so on. Clearly, after $O(\log n)$ rounds, Bob knows the value $\sum_{i\in[n]} x_i \cdot y_i$. Finally, observe that Alice communicates $O(\log n)$ bits, and that her circuits are of the form $\mathsf{MAJ} \circ \mathsf{MAJ}$. □

7 An improved round separation theorem for AC^0

Recall that Chattopadhyay and Santhanam [CS12] proved that there are Boolean functions on n variables that admit AC^0-bounded protocols with r rounds and cost $O(n^{1/r})$, but for which any correct AC^0-bounded $(r-1)$-round protocol has cost $\Omega(n^{2/r-o(1)})$. We use a different construction and refine their techniques, obtaining the following result.

Theorem 7.1. *Let $r \geq 2$ and $\varepsilon > 0$ be fixed parameters. There is an explicit family of functions $f = \{f_n\}_{n\in\mathbb{N}}$ with the following properties:*

- *(i) There exists an $\mathsf{AC}^0_2(n)$-bounded protocol Π_n for f_n with r rounds and cost $c(n) \leq n^{\varepsilon}$, for every $n \geq n_f$, where n_f is a fixed constant that depends on f.*
- *(ii) Any $\mathsf{AC}^0(\mathsf{poly}(n))$-bounded protocol Π for f with $r-1$ rounds has cost $c(n) \geq n^{1-\varepsilon}$, for every $n \geq n_\Pi$, where n_Π is a fixed constant that depends on Π.*

We will need some additional definitions and notation in order to establish this result. For any $n \in \mathbb{N}$, let $g_n \colon \{0,1\}^n \to \{0,1\}$ be the parity function on n variables, and $g = \{g_n\}_{n\in\mathbb{N}}$. Let m, ℓ, and r be positive integers. Set $n = n(m, \ell, r) \stackrel{\text{def}}{=} m + \ell \cdot r \cdot m$. We define a function $f_{m,\ell,r} \colon \{0,1\}^n \to \{0,1\}$ that will be used to prove round separation results for AC^0-compression games. For convenience, let $k \stackrel{\text{def}}{=} \log \ell$ and $v \stackrel{\text{def}}{=} m/\log \ell$. The definition of $f_{m,\ell,r}$ depends on g and a given function $h \colon \{0,1\}^k \to [\ell]$, which we assume to be some fixed one-to-one function.

Given any string $z \in \{0,1\}^n$, we write $z = (x, y^{(\cdot,1)}, \ldots, y^{(\cdot,r)})$, where $x \in \{0,1\}^m$, and $y^{(\cdot,j)} = (y^{(1,j)}, \ldots, y^{(\ell,j)})$, where $j \in [r]$, and $y^{(i,j)} \in \{0,1\}^m$, for every $i \in [\ell]$. In addition, for any string $w \in \{0,1\}^m$, we write $w = (w^{(1)}, \ldots, w^{(k)})$, where each $w^{(u)} \in \{0,1\}^v$, for $u \in [k]$. For convenience, instead of writing $y^{(i,j)(u)}$, we may also use $y^{(i,j,u)}$.

The function $f_{m,\ell,r}$ is defined by induction on r. It is simply a pointer jumping function, where h is applied to certain bits computed from the current string (initially x) using $k = \log \ell$ independent applications of g_v. After jumping from the initial x to a new string x', which will be one of the y's in $y^{(\cdot,1)}$, we recurse. After r steps, some string y from $y^{(\cdot,r)}$ will be reached. The output of $f_{m,\ell,r}$ is then set to be $g_m(y)$.

Formally, when $r = 1$, for any $z \in \{0,1\}^n$,

$$f_{m,\ell,1}(z) \stackrel{\text{def}}{=} g_m(y^{(i,1)}), \text{ where } i = h(g_v(x^{(1)}), \ldots, g_v(x^{(k)})).$$

Now let $r \geq 2$ be arbitrary. Then, for any $z \in \{0,1\}^n$,

$$f_{m,\ell,r}(z) \stackrel{\text{def}}{=} f_{m,\ell,r-1}(z'), \text{ where } z' = (x', y^{(\cdot,2)}, \ldots, y^{(\cdot,r)}),\ x' = y^{(i,1)}, \text{ and } i = h(g_v(x^{(1)}), \ldots, g_v(x^{(k)})).$$

This completes the definition of $f_{m,\ell,r}$.

Lemma 7.2 (Upper Bound)**.** *For any $m, \ell, r \geq 1$, the function $f_{m,\ell,r}$ admits an $\mathsf{AC}^0_2(m \cdot \ell)$-compression game with $r+1$ rounds and communication cost $c(n) = (r+1) \cdot m$.*

Proof. During each round j, Alice sends her current string $x' \in \{0,1\}^m$ to Bob, which replies with ℓ strings $v^{(i)} \in \{0,1\}^m$ satisfying the following property: $v^{(i)} = 1^m$ if the next round of the game is played on $y^{(i,j+1)}$, and $v^{(i)} = 0^m$ otherwise. Observe that the next message that Alice has to send is simply the m-bit string given by

$$\bigvee_{i \in [\ell]} \left(v^{(i)} \wedge y^{(i,j+1)} \right).$$

The cost and round complexity of this protocol is clear. □

We now proceed with the proof that in any AC^0-bounded protocol for $f_{m,\ell,r}$ with r rounds, Alice has to communicate roughly $\ell \cdot m$ bits, for an appropriate choice of ℓ that we would like to make as large as possible. The argument is based on random restrictions, which allow us to simplify the AC^0 circuits used by Alice considerably, while still maintaining the resulting function sufficiently hard for compression games. At a high level, we apply a round elimination technique, combined with a strong lower bound for $f_{m,\ell,1}$. More details follow.

From now on we will also view $f_{n,\ell,r}$ as a function $f_{m,\ell,r} \colon \{0,1\}^{[n]} \to \{0,1\}$, where each input z for $f_{m,\ell,r}$ can also be interpreted as a function $z \colon [n] \to \{0,1\}$. This will give us more flexibility when manipulating restrictions. A *restriction* $\rho \in \{0,1,*\}^{[n]}$ is simply a function $\rho \colon [n] \to \{0,1,*\}$. Given a restriction ρ and a function $f \colon \{0,1\}^{[n]} \to \{0,1\}$, we let $f^\rho \colon \{0,1\}^{\rho^{-1}(*)} \to \{0,1\}$ be the following function. For every $z^- \in \{0,1\}^{\rho^{-1}(*)}$,

$$f^\rho(z^-) \stackrel{\text{def}}{=} f(z), \text{ where } z \in \{0,1\}^{[n]} \text{ is the function with } z|_{\rho^{-1}(\{*\})} = z^- \text{ and } z|_{\rho^{-1}(\{0,1\})} = \rho|_{\rho^{-1}(\{0,1\})}.$$

Let $N \stackrel{\text{def}}{=} [n]$. Recall that we write $z \in \{0,1\}^n$ as $z = (x, y^{(1,1)}, \ldots, y^{(\ell,r)})$. Similarly, we let $S^{(i,j,u)} \subseteq N$ index the variables corresponding to $y^{(i,j,u)}$, for $i \in [\ell]$, $j \in [r]$ and $u \in [k]$. We define $S^{(i,j)} \stackrel{\text{def}}{=} \bigcup_u S^{(i,j,u)}$. Further, we use $M \subseteq N$ to index the variables corresponding to x, and $M^{(1)}, \ldots, M^{(k)}$ for the corresponding variables $x^{(1)}, \ldots, x^{(k)}$. Let Γ_N be the set of all restrictions with domain N, i.e., $\Gamma_N \stackrel{\text{def}}{=} \{0,1,*\}^N$. Given $\rho_1, \rho_2 \in \Gamma_N$, we say that ρ_2 extends ρ_1 if $\rho_2^{-1}(*) \subseteq \rho_1^{-1}(*)$ and $\rho_2|_{\rho_1^{-1}(\{0,1\})} = \rho_1|_{\rho_1^{-1}(\{0,1\})}$.

Our round separation theorem will be derived from lower bounds on a class of functions $\phi_{s,d,\ell}\colon \mathbb{N}\times\mathbb{N}\times\mathbb{R}^+\to\mathbb{N}$, defined as follows:

$$\phi_{s,d,\ell}\big(m,r,\delta\big) \stackrel{\text{def}}{=} \min_{\sigma\in\Gamma_{N,\delta}} \min_{\Pi\in\mathsf{Prot}^{\sigma}_{s,d,r}} \mathsf{cost}(\Pi),$$

where:[5]

(*i*) $\Gamma_{N,\delta}\subseteq\Gamma_N$ is the set of all restrictions σ for which the following holds: there exists sets $D_j\subseteq[\ell]$ with $j\in[r]$ such that $|D_j|\leq\delta\cdot\ell$, and $\sigma^{-1}(\{0,1\}) = \bigcup_{j\in[r]}\Big(\bigcup_{i\in D_j} S^{(i,j)}\Big)$,

(*ii*) $\mathsf{Prot}^{\sigma}_{s,d,r}$ is the set of all $\mathsf{AC}^0_d(s)$-bounded r-round protocols Π solving the compression game of $f^{\sigma}_{m,\ell,r}$.

The parameters m, r, and δ will vary during our inductive proof, while s, d, and ℓ remain fixed (observe that this is reflected in our notation for ϕ). The proof of Theorem 7.1 relies on the following lemmas, whose proof we present later in this section.

Lemma 7.3 (Lower Bound: Base case)**.** *Let $s=n^{c_1}$, $d\in\mathbb{N}$, $\ell=m^{c_2}$, $\delta\in(0,1/10)$, and $r=1$, where c_1 and c_2 are fixed positive integers. Then, for every fixed $\beta\in(0,1/10)$ and m sufficiently large,*

$$\phi_{s,d,\ell}\big(m,1,\delta\big)\geq\ell\cdot m^{1-\beta}.$$

Lemma 7.4 (Lower Bound: Induction step)**.** *Let $s=n^{c_1}$, $d\in\mathbb{N}$, $\ell=m^{c_2}$, $\delta\in(0,1/10)$, and $r\geq 2$, where c_1 and c_2 are fixed positive integers. Then, for every fixed $\beta\in(0,1/10)$ and m sufficiently large,*

$$\phi_{s,d,\ell}\big(m,r,\delta\big)\geq\min\Big\{\ell\cdot m^{1-\beta}\ ,\ \phi_{s,d,\ell}\big(m^{1-\beta},r-1,\delta+\beta\big)\Big\}.$$

These lemmas imply the following result.

Proposition 7.5. *For every fixed $r\geq 1$, $c\in\mathbb{N}$, and $\zeta>0$, for m sufficiently large, we have*

$$\phi_{\mathsf{poly}(n),O(1),m^c}\big(m,r,1/(100r)\big)\geq\ell\cdot m^{1-\zeta}.$$

Proof. The result follows easily from Lemmas 7.3 and 7.4 using that r is constant and that we can take β and δ sufficiently small. □

Finally, it is not hard to derive the main lower bound of this section from these results.

Proof of Theorem 7.1. Given any $r\geq 2$ and $\varepsilon>0$, it is enough to consider an appropriate family of functions $f_{m,\ell,r-1}$, where $c=c(\varepsilon)$ is sufficiently large, and set $\ell=m^c$. The result then follows from Lemma 7.2 and Proposition 7.5. □

We proceed now with the proof of the lemmas. We will need the notion of a random restriction. Let $p\in[0,1]$ be a real number. We let Γ^p_N denote the distribution over restrictions $\rho\in\Gamma_N$ generated by independently fixing each $\rho(i)$ (where $i\in N$) as follows:

$$\Pr[\rho(i)=*]=p,\quad \Pr[\rho(i)=1]=(1-p)/2,\quad \Pr[\rho(i)=0]=(1-p)/2.$$

Given a Boolean function $f_n\colon\{0,1\}^n\to\{0,1\}$ over n variables, we let $\mathsf{DT}_{\mathsf{depth}}(f)$ be the smallest decision tree depth among all decision trees computing f_n. The next statement is independent of the number of inputs of f.

[5]For the sake of this proof, we consider circuits of size at most s (exactly), instead of $O(s)$.

Lemma 7.6 (Switching Lemma [Hås86])**.** *Let f be a Boolean function that can be written as a conjunction or disjunction of any number of depth-t decision trees. Then, for every $p \in [0,1]$ and $r \in \mathbb{N}$,*

$$\Pr_{\rho \sim \Gamma^p} [\mathsf{DT}_{\mathsf{depth}}(f^\rho) > r] \leq (5pt)^r.$$

The next result is a standard consequence of Lemma 7.6 (cf. Gopalan and Servedio [GS10]).

Proposition 7.7. *Let f be a Boolean function computed by an AC^0 circuit of size M and depth d. For every $t \in \mathbb{N}$, if $p \leq 1/(10t)^d$ then*

$$\Pr_{\rho \sim \Gamma^p} [\mathsf{DT}_{\mathsf{depth}}(f^\rho) > t] \leq M \cdot 2^{-t}.$$

Given a function $C \colon \{0,1\}^{[n]} \to \{0,1\}$, we let $\mathsf{live}(C) \subseteq [n]$ denote the set of input variables of C with influence greater than zero. It will be more convenient for us to rely on the following straightforward consequence of Lemma 7.6 and Proposition 7.7.

Lemma 7.8. *Let $C_1, \ldots, C_{s_1} \colon \{0,1\}^{n_1} \to \{0,1\}$ be functions computed by depth-d AC^0 circuits of size at most $n_1^{c_1}$, where $d, c_1 \in \mathbb{N}$ and $s_1 = m^{1-\gamma} \cdot \ell$, and these parameters satisfy $m, \ell \in \mathbb{N}$, $\gamma \in (0, 1/5)$, $\ell = m^{c_2}$, where $c_2 \in \mathbb{N}$, and $n_1 = \Theta(m \cdot \ell)$. Then, for $p = m^{-\gamma/2}$, there exists a constant c_3 such that, as $m \to \infty$,*

$$\Pr_{\rho \sim \Gamma^p_{[n_1]}} \left[\left| \bigcup_{i \in [s_1]} \mathsf{live}\left(C_i^\rho\right) \right| \leq c_3 \cdot (m^{1-\gamma} \cdot \ell) \right] \to 1.$$

Proof. Let $p = p_1 \cdot p_2$, where $p_1 = p_2 = m^{-\gamma/4}$. Observe that sampling a restriction $\rho \sim \Gamma^p_{[n_1]}$ is equivalent to first sampling some $\rho_1 \sim \Gamma^{p_1}_{[n_1]}$, followed by a restriction $\rho_2 \sim \Gamma^{p_2}_W$, where $W \stackrel{\text{def}}{=} [n_1] \setminus \rho_1^{-1}(\{0,1\})$, and finally setting $\rho = \rho_2 \circ \rho_1$, where the composition operation is defined in the natural way. Let $c = c_1 + 10$, and $t = c \cdot \log n_1$. Furthermore, we let $r = \lceil 8(1+c_2)/\gamma \rceil$, and $c_3 = 2^r$. Then,

$$\begin{aligned}
\Pr_{\rho \sim \Gamma^p_{[n_1]}} \left[\left| \bigcup_{i \in [s_1]} \mathsf{live}\left(C_i^\rho\right) \right| > c_3 \cdot (m^{1-\gamma} \cdot \ell) \right] &\leq \Pr_{\rho \stackrel{\text{def}}{=} \rho_2 \circ \rho_1} \left[\exists i \in [s_1] \text{ s.t. } |\mathsf{live}(C_i^\rho)| > 2^r \right] \\
&\leq \Pr_{\rho_1, \rho_2} \left[\exists i \in [s_1] \text{ s.t. } \mathsf{DT}_{\mathsf{depth}}(C_i^\rho) > r \right]
\end{aligned}$$

In order to conclude the proof, it is enough to show that for every $j \in [s_1]$ and sufficiently large m, $\Pr_{\rho_1, \rho_2}[\mathsf{DT}_{\mathsf{depth}}(C_j^\rho) > r] \leq (1/n_1)^2$. However, by our choice of parameters (and with room to spare), this follows from an application of Proposition 7.7 with ρ_1 and t, followed by an application of Lemma 7.6 with ρ_2 and r (notice that these statements are true with respect to any input size). □

We are now ready to prove Lemmas 7.3 and 7.4.

Proof of Lemma 7.3. Let $\sigma \colon [n] \to \{0,1,*\}$ be a restriction in $\Gamma_{N,\delta}$, where $n = m + \ell \cdot m$ and $N = [n]$, as usual. Let $N_1 \stackrel{\text{def}}{=} N \setminus \sigma^{-1}(\{0,1\})$, and set $n_1 \stackrel{\text{def}}{=} |N_1|$. Observe that $n_1 \geq (1-\delta) \cdot \ell \cdot m = \Theta(m \cdot \ell)$. In addition, let $\Pi = (C^{(1)}, g^{(1)}, E)$ be a single-round protocol for $f^\sigma_{m,\ell,1}$, where $C^{(1)} = (C_1, \ldots, C_{s_1})$, and these are AC^0 circuits of depth d and size $s = n^{c_1} \leq n_1^{2c_1}$ (for large enough m) that compute the

message in $\{0,1\}^{s_1}$ that Alice sends to Bob. By definition, for each $i \in [s_1]$, $C_i \colon \{0,1\}^{n_1} \to \{0,1\}$. We prove that if $s_1 < \ell \cdot m^{1-\beta}$, then there exists an input $z \in \{0,1\}^{n_1}$ for which $\Pi(z) \neq f^{\sigma}_{m,\ell,1}(z)$.

Let $D_1 \subseteq [\ell]$ be the set identifying the variables y fixed by σ (according to our definition of $\Gamma_{N,\delta}$). For any $z \in \{0,1\}^{N_1}$, we write $z = (x, y^{(i_1,1)}, \ldots, y^{(i_k,1)})$, where $[\ell] \setminus D_1 = \{i_1, \ldots, i_k\}$, $k \geq (1-\delta) \cdot \ell$, and $x \in \{0,1\}^m$. Recall that we use sets $S^{(i_1,1)}, \ldots, S^{(i_k,1)}$ and M to address the elements of $[N_1]$ corresponding to these input positions.

Now consider a random restriction $\rho \sim \Gamma^p_{N_1}$, where $p = m^{-\beta/2}$. Applying Lemma 7.8 with $\gamma = \beta$ and Proposition A.1, it follows that, for every large enough m, with high probability:

- (i) $C^{(1),\rho}$ depends on at most $O(m^{1-\beta} \cdot \ell)$ variables.
- (ii) For every $j \in [\log \ell]$, it is the case that $\rho^{-1}(*) \cap M^{(j)} \neq \emptyset$.
- (iii) $|\rho^{-1}(*) \cap (S^{(i_1,1)} \cup \ldots \cup S^{(i_k,1)})| \geq \frac{1}{2} \cdot \frac{(1-\delta) \cdot m \cdot \ell}{m^{\beta/2}} = \Omega(m^{1-\beta/2} \cdot \ell)$. In particular, from (i) we get that there exists $i \in [\ell] \setminus D_1$ for which $S^{(i,1)} \cap \left(\rho^{-1}(*) \setminus \mathsf{live}(C^{(1),\rho})\right) \neq \emptyset$.

Overall, it follows that there exists a restriction $\overline{\rho} \in \Gamma_N$ with $\overline{\rho} = \rho \circ \sigma$, for an appropriate choice of $\rho \in \Gamma_{N_1}$, such that $\overline{\rho}$ fixes the message sent by Alice, but does not fix the value of $f^{\overline{\rho}}_{m,\ell,1}$. In particular, there exists a $z \in \{0,1\}^{n_1}$ that agrees with $\overline{\rho}$ for which $\Pi(z) \neq f^{\sigma}_{m,\ell,1}(z)$, which completes the proof. □

The proof of Lemma 7.4 is not much harder than the argument used in the base case, but it has a few technicalities that need to be handled.

Proof of Lemma 7.4. Let $\sigma \in \Gamma_{N,\delta}$ and $\Pi \in \mathsf{Prot}^{\sigma}_{s,d,r}$ be a pair realizing $\phi_{s,d,\ell}(m,r,\delta)$. In other words, Π solves the compression game of $f^{\sigma}_{m,\ell,r}$, and $\mathsf{cost}(\Pi) = \phi_{s,d,\ell}(m,r,\delta)$. Assume that $\Pi = (C^{(1)}, \ldots, C^{(r)}, g^{(1)}, \ldots, g^{(r-1)}, E)$, and $\mathsf{signature}(\Pi) = (n_1, s_1, t_1, \ldots, t_{r-1}, s_r)$, where $n = m + m \cdot \ell \cdot r$, $N = [n]$, $N_1 = N \setminus \sigma^{-1}(\{0,1\})$, and $n_1 = |N_1|$. For convenience, let $C^{(1)} = (C_1, \ldots, C_{s_1})$, where each C_i is a depth-d AC^0 circuit of size at most $n^{c_1} \leq n_1^{2c_1}$ (for large m), since $n_1 \geq (1-\delta) \cdot n$.

Notice that if $\mathsf{cost}(\Pi) \geq \ell \cdot m^{1-\beta}$ then the statement of Lemma 7.4 is true. Otherwise, from $\mathsf{cost}(\Pi) < \ell \cdot m^{1-\beta}$ we get that $s_1 < \ell \cdot m^{1-\beta}$, which allows us to proceed as in the proof of Lemma 7.3. Let $p = m^{-\beta/2}$, and set $\gamma = \beta$. It follows from Lemma 7.8 that, with high probability,

$$\left|\mathsf{live}(C^{(1),\rho})\right| = O(m^{1-\beta} \cdot \ell). \tag{3}$$

Let D_j for $j \in [r]$ be the sets identifying the variables y fixed by σ. By assumption, $|D_j| \leq \delta \cdot \ell$ for every $j \in [r]$. From now on, whenever we consider a set $S^{(i,j)}$, we implicitly assume that $j \in [r]$ and $i \in [\ell] \setminus D_j$. This time we will also be concerned about how the action of ρ affects the more specific sets $S^{(i,j,u)}$, where $u \in [\log \ell]$. Observe that, with high probability (Proposition A.1), for every (i,j,u), we have:

$$\left|S^{(i,j,u)} \cap \rho^{-1}(*)\right| \geq \frac{1}{2} \cdot \frac{m}{\log \ell} \cdot p = \frac{1}{2} \cdot \frac{m^{1-\beta/2}}{c_2 \log m} \geq m^{1-(3/4)\beta}, \tag{4}$$

for any sufficiently large m. We say that a set $S^{(i,j)}$ is *bad* with respect to $C^{(1),\rho}$ if $|S^{(i,j)} \cap \mathsf{live}(C^{(1),\rho})| \geq \frac{1}{2} \cdot m^{1-(3/4)\beta}$. Otherwise, the set is said to be *good*. It follows from Equation 3 that

$$\text{Number of bad sets } S^{(i,j)} \leq \frac{O(m^{1-\beta} \cdot \ell)}{(1/2) \cdot m^{1-(3/4)\beta}} = \frac{2\ell}{m^{\beta/4}} = o(\ell), \tag{5}$$

as $m \to \infty$. In particular, since $r = O(1)$ and β is a fixed constant, with high probability, for every $j \in [r]$ there are at most $\beta \cdot \ell$ sets $S^{(i,j)}$ that are bad with respect to $C^{(1),\rho}$. Finally, with high probability over ρ, we also get that, for every $j \in [\log \ell]$,

$$\left|M^{(j)} \cap \rho^{-1}(*)\right| > 0.$$

It follows using the probabilistic method that there exists a fixed restriction $\rho_1 \in \Gamma_{N_1}$ satisfying all these properties. Let $\rho_2 = \rho_1 \circ \sigma$ be the restriction obtained by combining ρ_1 and σ in the obvious way. Observe that $\rho_2 \colon N \to \{0, 1, *\}$. Fix arbitrarily all $*$-variables in ρ_2 corresponding to bad sets $S^{(i,j)}$. On every good set $S^{(i,j)}$, fix all $*$-variables intersecting $\mathsf{live}(C^{(1),\rho_1})$, and also fix additional variables in each set $S^{(i,j,u)}$ so that the new restriction ρ_3 satisfies $|\rho_3^{-1}(*) \cap S^{(i,j,u)}| = m^{1-\beta}$, for every appropriate triple (i, j, u). This is possible for any large enough m, since these sets are good. Further, we assume that the number of variables corresponding to each $S^{(i,j,u)}$ that are set to 1 is *even*, in order not to invert the parity inside each block, which will be important later in the proof. Let $f^{\rho_3}_{m,\ell,r} \colon \{0,1\}^{\rho_3^{-1}(*)} \to \{0,1\}$ be the resulting function.

Given an input $\tilde{z} \in \{0,1\}^{\rho_3^{-1}(*)}$, write $\tilde{z} = (\tilde{x}, \{\tilde{y}^{(i,j)}\})$, and let $z = (x, \{y^{(i,j)}\}) \in \{0,1\}^n$ be the completion of $\tilde{z}$ that *agrees* with ρ_3, where this notion is defined in the natural way. Observe that $h(x)$ still depends on $\tilde{x}$. Now we set all remaining $*$-variables in M in a way that, for the new restriction $\overline{\sigma} \colon [N] \to \{0, 1, *\}$, we have $h(\overline{\sigma}(M))$ pointing to a pair $(i, 1)$ corresponding to a good set $S^{(i,1)}$. This is possible due to the properties of ρ_1. Observe that $C^{(1),\overline{\sigma}}$ computes a constant function (i.e., Alice's message $a^{(1)}$ has been fixed). Let $b^{(1)} \in \{0,1\}^{t_1}$ be the answer provided by Bob, which is also fixed.

Now let $\overline{\Pi} = (\overline{C}^{(1)}, \ldots, \overline{C}^{(r-1)}, \overline{g}^{(1)}, \ldots, \overline{g}^{(r-2)}, E)$ be a new protocol obtained by setting each $\overline{C}^{(i)}$ to be $C^{(i+1)}$ with its input corresponding to the first message sent by Bob fixed to $b^{(1)}$, and $\overline{g}^{(i)} = g^{(i+1)}$, for every appropriate i. If we also rename the input variables in $f^{\overline{\sigma}}_{m,\ell,r}$ and in the functions and circuits from $\overline{\Pi}$, truncating irrelevant variables appropriately (recall the definition of the original function as a pointer jumping function), we obtain a restriction $\sigma' \colon \{0,1\}^{N'} \to \{0,1\}$, where $n' = |N'| = m' + m' \cdot \ell \cdot r'$, $m' = m^{1-\beta}$, $r' = r - 1$, $\sigma' \in \Gamma_{N',\delta'}$, $\delta' = \delta + \beta$, and the resulting protocol $\Pi' \in \mathsf{Prot}^{\sigma'}_{s,d,r'}$. Crucially, Π' is a protocol solving the compression game of $f^{\sigma'}_{m',\ell,r'}$ in r' rounds, which implies that $\mathsf{cost}(\Pi) \geq \mathsf{cost}(\Pi') \geq \phi_{s,d,\ell}(m', r', \delta') = \phi_{s,d,\ell}(m^{1-\beta}, r - 1, \delta + \beta)$, completing the proof of Lemma 7.4. □

8 Open Problems and Further Research Directions

Our results and techniques raise a number of interesting questions, which we discuss more carefully below.

The power of interaction in two-party $\mathsf{AC}^0[p]$-compression games. Observe that the approach to obtain communication lower bounds for $\mathsf{AC}^0[p]$ games employed in the proof of Theorem 1.1 is insensitive to the number of rounds of the protocol. On the other hand, our round separation result (Theorem 1.6) holds with respect to AC^0 circuits only. Consequently, a natural question is whether a strong round separation theorem is true for $\mathsf{AC}^0[p]$ games. We conjecture that this is the case, and that a hard function can be obtained via a similar construction that uses MOD_q instead of parity.

Randomized $\mathsf{AC}^0[p]$-compression games. While we have obtained essentially optimal lower bounds for deterministic two-party $\mathsf{AC}^0[p]$-compression games, the situation is less clear with respect to randomized protocols. Modulo logarithmic factors, there is a quadratic gap between our upper and lower bounds for MOD_q and $\mathsf{Majority}$ (Theorem 1.3). On the other hand, it is known that the communication cost of these games is $n/\log^{\Theta(d)} n$ for randomized AC^0_d-compression games (Chattopadhyay and Santhanam [CS12]). We are unable to obtain better lower bounds here because our approach does not seem to tolerate the initial error probability from the protocol, as it relies on the low error regime of the polynomial approximation method.

Extending circuit lower bounds to incompressibility results. The results presented in this paper and in [CS12] show that recent extensions of the random restriction method and the polynomial approximation method can provide optimal incompressibility results. However, our construction from Section 6 implies that not every technique can be extended in this sense. Which other techniques and results from circuit complexity can be strengthened to compressibility lower bounds?

Understanding the structure of Boolean circuits. Our results shed more light into the computation of Boolean functions such as MOD_q using $\mathsf{AC}^0[p]$ circuits, as we are able to obtain information about each layer of the circuit. Similar developments appear for instance in Tarui [Tar10], Rudich and Berman [RB88], and Borodin [Bor71]. We believe that results of this form can provide important insights in algorithms and computational complexity, and it would be very interesting to see further advances in this direction.

References

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[AB09] Sanjeev Arora and Boaz Barak. *Complexity Theory: A Modern Approach*. Cambridge University Press, Cambridge, 2009.

[ABO84] Miklós Ajtai and Michael Ben-Or. A theorem on probabilistic constant depth computations. In *Symposium on Theory of Computing* (STOC), pages 471–474, 1984.

[Ajt83] Miklós Ajtai. $\sum^1_1$-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3), 2010.

[And85] Alexander E. Andreev. On a method for obtaining lower bounds for the complexity of individual monotone functions. *Soviet Math. Dokl*, 31(3):530–534, 1985.

[AS92] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley, New York, 1992.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *Transactions on Computation Theory* (TOCT), 1(1), 2009.

[BDFH09] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizatons of the $\mathsf{P} =? \mathsf{NP}$ Question. *SIAM J. Comput.*, 4(4):431–442, 1975.

[BH08] Harry Buhrman and John M. Hitchcock. NP-hard sets are exponentially dense unless $\mathsf{coNP} \subseteq \mathsf{NP/poly}$. In *Conference on Computational Complexity* (CCC), pages 1–7, 2008.

[Bor71] Allan Borodin. Horner's rule is uniquely optimal. In *International Symposium on the Theory of Machines and Computations*, pages 45–57, 1971.

[CS12] Arkadev Chattopadhyay and Rahul Santhanam. Lower bounds on interactive compressibility by constant-depth circuits. In *Symposium on Foundations of Computer Science* (FOCS), pages 619–628, 2012.

[Del14] Holger Dell. A simple proof that AND-compression of NP-complete problems is hard. ECCC Report TR14-75, 2014.

[DI06] Bella Dubrov and Yuval Ishai. On the randomness complexity of efficient sampling. In *Symposium on Theory of Computing* (STOC), pages 711–720, 2006.

[Dru12] Andrew Drucker. New limits to classical and quantum instance compression. In *Symposium on Foundations of Computer Science* (FOCS), pages 609–618, 2012.

[DvM14] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23, 2014.

[Fel43] William Feller. Generalization of a probability limit theorem of Cramér. *Transactions of the American Mathematical Society*, 54(3):361–372, 1943.

[FKL+01] Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels Schmitt, and Hans-Ulrich Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *Foundations of Software Technology and Theoretical Computer Science* (FSTTCS), pages 171–182, 2001.

[For02] Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. Syst. Sci.*, 65(4):612–625, 2002.

[FRR+10] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting circuits from leakage: the computationally-bounded and noisy cases. In *International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT), pages 135–156, 2010.

[FS11] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct pcps for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[GS10] Parikshit Gopalan and Rocco A. Servedio. Learning and lower bounds for AC^0 with threshold gates. In *International Workshop on Randomization and Computation* (RANDOM), pages 588–601, 2010.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Symposium on Theory of Computing* (STOC), pages 6–20, 1986.

[HMP+93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993.

[HN10] Danny Harnik and Moni Naor. On the compressibility of NP instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

[Khr71] V. M. Khrapchenko. A method of determining lower bounds for the complexity of π-schemes. *Math. Notes Acad. of Sci.* (USSR), 10(1):474–479, 1971.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[KS12] Swastik Kopparty and Srikanth Srinivasan. Certifying polynomials for $\mathrm{AC}^0(\oplus)$ circuits, with applications. In *Foundations of Software Technology and Theoretical Computer Science* (FSTTCS), pages 36–47, 2012.

[MV08] Jiří Matoušek and Jan Vondrák. *Lecture notes on the probabilistic method*, 2008.

[Neč66] Eduard I. Nečiporuk. On a Boolean function. *Soviet Math. Dokl.*, 7(4):999–1000, 1966.

[Oli13] Igor C. Oliveira. Algorithms versus circuit lower bounds. ECCC Report TR13-117, 2013.

[PS84] Christos H. Papadimitriou and Michael Sipser. Communication complexity. *J. Comput. Syst. Sci.*, 28(2):260–269, 1984.

[Raz85] Alexander A. Razborov. Lower bounds on monotone complexity of the logical permanent. *Mathematical Notes*, 37(6):485–493, 1985.

[Raz87] Alexander A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(4):598–607, 1987.

[RB88] Steven Rudich and Leonard Berman. Optimal circuits and transitive automorphism groups. In *International Colloquium on Automata, Languages and Programming* (ICALP), pages 516–524, 1988.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[San12] Rahul Santhanam. Ironic complicity: Satisfiability algorithms and circuit lower bounds. *Bulletin of the* EATCS, 106:31–52, 2012.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Symposium on Theory of Computing* (STOC), pages 77–82, 1987.

[Sri13] Srikanth Srinivasan. On improved degree lower bounds for polynomial approximation. In *Foundations of Software Technology and Theoretical Computer Science* (FSTTCS), pages 201–212, 2013.

[Tar10] Jun Tarui. Smallest formulas for the parity of 2^k variables are essentially unique. *Theor. Comput. Sci.*, 411(26-28):2623–2627, 2010.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Mathematical Foundations of Computer Science* (MFCS), pages 162–176, 1977.

[Val83] Leslie G. Valiant. Exponential lower bounds for restricted monotone circuits. In *Symposium on Theory of Computing* (STOC), pages 110–117, 1983.

[Vio09] Emanuele Viola. On the power of small-depth computation. *Foundations and Trends in Theoretical Computer Science*, 5(1):1–72, 2009.

[Wil14a] Ryan Williams. Algorithms for circuits and circuits for algorithms (Invited Talk). In *Conference on Computational Complexity* (CCC), pages 248–261, 2014.

[Wil14b] Ryan Williams. Faster all-pairs shortest paths via circuit complexity. In *Symposium on Theory of Computing* (STOC), pages 664–673, 2014.

[Wil14c] Ryan Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2, 2014.

[Yao85] Andrew Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *Symposium on Foundations of Computer Science* (FOCS), pages 1–10, 1985.

A Auxiliary results

We use the following standard concentration bound (cf. Alon and Spencer [AS92], Appendix A).

Proposition A.1. *Let $X_1, \ldots, X_m$ be independent $\{0,1\}$ random variables, where each X_i is 1 with probability $p \in [0,1]$. In addition, set $X \stackrel{\text{def}}{=} \sum_i X_i$, and $\mu \stackrel{\text{def}}{=} \mathbb{E}[X] = pm$. Then, for any fixed $\zeta > 0$, there exists a constant $c_\zeta > 0$ such that*

$$\Pr[\,|X - \mu| > \zeta\mu] < 2e^{-c_\zeta \mu}.$$

B The degree lower bound in the low-error regime

In this section we describe the proof of the degree lower bound for $\mathbb{F}_p$-polynomials approximating MOD_q in the low error regime. Recall that we use MOD_q^n to denote the MOD_q function over n input variables, and that a polynomial $Q \in \mathbb{F}_p[x_1, \ldots, x_n]$ $\varepsilon(n)$-approximates a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ under the uniform distribution if

$$\Pr_{x \sim \{0,1\}^n}[Q(x) = f(x)] \geq 1 - \varepsilon(n),$$

where x is viewed as an element of $\mathbb{F}_p^n$ or $\{0,1\}^n$, depending on the context.

Proposition B.1 ([Raz87, Smo87], folklore)**.** *Let $p, q \geq 2$ be distinct primes. There exist fixed constants $\delta > 0$ and $n_0 \in \mathbb{N}$ for which the following holds. For every $n \geq n_0$ and $\varepsilon(n) \in [2^{-n}, 1/10q]$, any polynomial $P \in \mathbb{F}_p[x_1, \ldots, x_n]$ that ε-approximates the MOD_q^n function with respect to the uniform distribution has degree at least $\delta \cdot \sqrt{n \cdot \log(1/\varepsilon)}$.*

The proofs that appear in the literature are concerned with large values of ε, and our goal here is to discuss the extension of the degree lower bound to very small ε, as stated in Proposition B.1. For this reason, we will focus on the case where $q = 2$ and $p > 2$, which is slightly simpler. We start with the following lemma.

Lemma B.2. *For a prime $p > 2$, let $P \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a degree-d polynomial that $\varepsilon(n)$-approximates MOD_2^n over the uniform distribution. Then there exists a polynomial $Q \in \mathbb{F}_p[y_1, \ldots, y_n]$ of degree at most d and a set $S \subseteq \{-1,1\}^n \subseteq \mathbb{F}_p^n$ with $|S| \geq (1-\varepsilon)2^n$ such that*

$$\forall y \in S, \quad Q(y) = \prod_{i=1}^{n} y_i.$$

Proof. Let $T \subseteq \{0,1\}^n \subseteq \mathbb{F}_p^n$ be a set of size at least $(1-\varepsilon)2^n$ such that

$$\forall x \in T, \quad P(x) = \mathsf{MOD}_2^n(x).$$

Consider the map $\gamma\colon \{-1,1\} \to \{0,1\}$ computed by the $\mathbb{F}_p$-polynomial $\gamma(y) \stackrel{\text{def}}{=} (1-y)2^{-1}$. Observe that $\gamma(-1) = 1$ and $\gamma(1) = 0$. Let $Q(y_1, \ldots, y_n)$ be a polynomial in $\mathbb{F}_p[y_1, \ldots, y_n]$ with $Q(y) \stackrel{\text{def}}{=} 2P(\gamma(y_1), \ldots, \gamma(y_n)) - 1$, and let

$$S \stackrel{\text{def}}{=} \{y \in \{-1,1\}^n \mid (y_1, \ldots, y_n) = (\gamma^{-1}(x_1), \ldots, \gamma^{-1}(x_n)), \text{ where } x \in T\}.$$

Then, using the definition of P, Q, S, T, and γ, it is not hard to see that

$$\forall y \in S, \quad Q(y) = \prod_{i=1}^{n} y_i.$$

Finally, observe that $|S| = |T|$ and $\mathsf{deg}(Q) \leq \mathsf{deg}(P)$, which completes the proof of the lemma. □

The next lemma shows that polynomials with this property can be very useful when computing functions defined over $S \subset \mathbb{F}_p^n$.

Lemma B.3. *Let $\mathbb{F}$ be a finite field, and $a, b \in \mathbb{F}$ be distinct non-zero elements. Assume that $Q \in \mathbb{F}[x_1, \ldots, x_n]$ is a degree-d polynomial, and $S \subseteq \{a, b\}^n$ is a set such that*

$$\forall x \in S, \quad Q(x) = \prod_{i=1}^{n} x_i.$$

Then, for every function $f \colon S \to \mathbb{F}$, there is a polynomial $Q_f \in \mathbb{F}[x_1, \ldots, x_n]$ with degree at most $(n + d)/2$ such that

$$\forall x \in S, \quad Q_f(x) = f(x).$$

Proof. Fix a function $f \colon S \to \mathbb{F}$, and let P_f be a multilinear polynomial such that, for all $x \in S$, $P_f(x) = f(x)$. For instance, since a and b are distinct elements of $\mathbb{F}$, we can take

$$P_f(x) \stackrel{\text{def}}{=} \sum_{x \in S} f(x) \cdot \left(\prod_{i : x_i = a} (b - x_i)(b - a)^{-1} \right) \left(\prod_{i : x_i = b} (a - x_i)(a - b)^{-1} \right).$$

Now consider any monomial $M(x) \stackrel{\text{def}}{=} \prod_{i \in I} x_i$, where $I \subseteq [n]$. Since a and b are non-zero, for any $y \in S \subseteq \{a, b\}^n$, we have

$$\begin{aligned} \prod_{i \in I} y_i &= \left(\prod_{i \in [n]} y_i \right) \left(\prod_{i \notin I} y_i^{-1} \right) \\ &= Q(y) \cdot \left(\prod_{i \notin I} a^{-1}(b - y_i)(b - a)^{-1} + b^{-1}(a - y_i)(a - b)^{-1} \right), \end{aligned}$$

where Q is the polynomial granted by the statement of the lemma. Therefore, each monomial in P_f defined over a subset $I \subseteq [n]$ can be replaced by a monomial of degree at most $\min(|I|, d + n - |I|) \leq (n + d)/2$, in the sense that the new polynomial is still correct on every input in S. Consequently, there exists a polynomial Q_f for f with degree at most $(n + d)/2$, as claimed by the lemma. □

In other words, if d is small, there exist polynomials of degree much smaller than n for all functions with domain S and codomain $\mathbb{F}$. This is impossible for large sets S, via a simple counting argument. In order to formalize this argument and obtain good parameters, we rely on a certain lower bound for the binomial distribution. The next lemma follows from more general results presented in Feller [Fel43]. We follow closely the exposition in Matoušek and Vondrák [MV08].

Lemma B.4. *For an even integer $n \in \mathbb{N}$, consider independent random variables $X_1, \ldots, X_n$, where each X_i attains values 0 and 1, each with probability 1/2. Let $X \stackrel{\text{def}}{=} \sum_{i \in [n]} X_i$. Then, for any integer $t \in [0, n/8]$,*

$$\Pr\left[X \geq \frac{n}{2} + t\right] \geq \frac{1}{15} \cdot e^{-16t^2/n}.$$

Proof. For convenience, let $n = 2m$. Then,

$$\begin{aligned}
\Pr[X \geq m + t] &= 2^{-2m} \sum_{j=t}^{m} \binom{2m}{m+j} \\
&\geq 2^{-2m} \sum_{j=t}^{2t-1} \binom{2m}{m+j} \\
&= 2^{-2m} \sum_{j=t}^{2t-1} \binom{2m}{m} \frac{m}{m+j} \cdot \frac{m-1}{m+j-1} \cdots \frac{m-j+1}{m+1} \\
&\geq \frac{1}{2\sqrt{m}} \sum_{j=t}^{2t-1} \prod_{i=1}^{j} \left(1 - \frac{j}{m+i}\right) \quad \left(\text{since } \binom{2m}{m} \geq 2^{2m}/(2\sqrt{m})\right) \\
&\geq \frac{t}{2\sqrt{m}} \left(1 - \frac{2t}{m}\right)^{2t} \\
&\geq \frac{t}{2\sqrt{m}} \cdot e^{-8t^2/m} \quad (\text{since } 1 - x \geq e^{-2x} \text{ for } 0 \leq x \leq 1/2).
\end{aligned}$$

The lemma now follows depending on the value of t. Observe that if $t \geq \frac{1}{4}\sqrt{m}$ then the last expression is lower bounded by $\frac{1}{8}e^{-16t^2/n}$. On the other hand, for $0 \leq t < \frac{1}{4}\sqrt{m}$, we get that $\Pr[X \geq m + t] \geq \Pr[X \geq m + \frac{1}{4}\sqrt{m}] \geq \frac{1}{8}e^{-1/2} \geq \frac{1}{15}$, which completes the proof. □

Finally, we combine these lemmas in order to prove Proposition B.1 for primes $q = 2$ and $p > 2$.

Proof. Let $P \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a degree-d polynomial that $\varepsilon(n)$-approximates the MOD_2^n function over the uniform distribution. Assume without loss of generality that n is even, since otherwise we can obtain a polynomial $Q \in \mathbb{F}_p[x_1, \ldots, x_{n+1}]$ with degree at most $2d$ that $\varepsilon(n)$-approximates MOD_2^{n+1} with respect to $\{0,1\}^{n+1}$ (i.e., apply P to the first n variables, then compose with the appropriate function over two input variables).

It follows from Lemmas B.2 and B.3 that there exists a set $S \subseteq \{-1, 1\}^n \subseteq \mathbb{F}_p^n$ of size $(1 - \varepsilon)2^n$ such that, for every function $f \colon S \to \mathbb{F}_p$, there exists a polynomial $Q_f \in \mathbb{F}_p[x_1, \ldots, x_n]$ of degree at most $d' \stackrel{\text{def}}{=} (n + d)/2$ that agrees with f over S.

Let $\mathcal{F}$ be the set of such functions. Clearly, $|\mathcal{F}| = |\mathbb{F}_p|^{|S|}$. On the other hand, since $S \subseteq \{-1, 1\}^n$, we can assume that each polynomial Q_f is multilinear. The number of such polynomials with degree at most d' is upper bounded by $|\mathbb{F}_p|^M$, where $M \stackrel{\text{def}}{=} \sum_{i=0}^{d'} \binom{n}{i}$. Therefore, $|\mathbb{F}_p|^{|S|} \leq |\mathcal{F}| \leq |\mathbb{F}_p|^M$, and we get that

$$\sum_{i=0}^{(n+d)/2} \binom{n}{i} \geq (1 - \varepsilon) \cdot 2^n. \tag{6}$$

We use this inequality to lower bound d in terms of n and ε. First, Equation 6 can be rewritten as

$$2^{-n} \cdot \sum_{i>(n+d)/2} \binom{n}{i} \leq \varepsilon. \tag{7}$$

On the other hand, it follows from Lemma B.4 that, for any $d \in [0, n/8]$,

$$\frac{1}{15} \cdot \exp\left(-\frac{16}{n} \cdot \left(\frac{d}{2}+1\right)^2\right) \leq \Pr\left[X > \frac{n}{2}+\frac{d}{2}\right] = 2^{-n} \cdot \sum_{i>(n+d)/2} \binom{n}{i}. \tag{8}$$

Therefore, we obtain from Equations 7 and 8 that $d = \Omega(\sqrt{n \cdot \log(1/\varepsilon)})$ for any $\varepsilon(n) \in [2^{-n}, 1/20]$, which completes the proof. □

C Improved approximation of $\mathsf{AC}^0[p]$ circuits by polynomials

For convenience of the reader, we describe in this section how to approximate Boolean circuits by bounded-degree polynomials in the low-error regime. We assume the following classic result, obtained in slightly different forms by Razborov [Raz87] and Smolensky [Smo87].

Proposition C.1 ([Raz87], [Smo87])**.** *Let p be a fixed prime. There exists a constant $\beta = \beta(p) \in \mathbb{N}$ such that, for every $d = d(n) \geq 1$ and $s = s(n) \geq 1$, any $\mathsf{AC}^0_d[p](s(n))$ circuit admits an $1/(6s)$-error probabilistic polynomial $\mathbf{Q}(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n]$ of degree at most $(\beta \cdot \log \max\{s, 2\})^d$.*

We are now ready to describe the proof of the degree upper bound obtained by Kopparty and Srinivasan [KS12], which allows us to obtain better bounds when the error is sufficiently small.

Proposition C.2 ([KS12])**.** *Let p be a fixed prime. There exists a constant $\alpha = \alpha(p) \in \mathbb{N}$ such that, for every $\delta \in (0, 1/2)$ and $d(n) \geq 2$, any $\mathsf{AC}^0_d[p](s(n))$ circuit C admits a δ-error probabilistic polynomial $\mathbf{Q}(x_1, \ldots, x_n) \in \mathbb{F}_p[x_1, \ldots, x_n]$ of degree at most $(\alpha \cdot \log s)^{d-1} \cdot \log(1/\delta)$. In particular, it follows that for any distribution $\mathcal{D}$ over $\{0,1\}^n$, C is δ-approximated with respect to $\mathcal{D}$ by a polynomial of degree at most $(\alpha \cdot \log s)^{d-1} \cdot \log(1/\delta)$.*

Proof. Let C be an $\mathsf{AC}^0[p]$ circuit of size s and depth $d \geq 2$. Further, let g be the top gate of C, and assume that this gate is fed by $t \leq s$ input wires $y_1, \ldots, y_t$, where each $y_j = g_j(x_1, \ldots, x_n)$. Observe that the corresponding Boolean function over inputs $x_1, \ldots, x_n$ at each gate g_j is computed by a circuit of size at most s and depth at most $d-1$, while $g = g(y_1, \ldots, y_t)$ is computed by a circuit of size one. Let $\varepsilon \stackrel{\text{def}}{=} 1/(6s)$. Then, Proposition C.1 guarantees the existence of probabilistic polynomials $\mathbf{Q}_j(x_1, \ldots, x_n)$ which compute the corresponding functions g_j with error at most ε, where $\mathsf{deg}(\mathbf{Q}_j) \leq (\beta \cdot \log s)^{d-1}$. Similarly, since g is computed by a single gate, there exists a probabilistic polynomial $\mathbf{Q}_g(y_1, \ldots, y_t)$ that computes g with error at most $1/6$, where $\mathsf{deg}(\mathbf{Q}_g) \leq \beta$. By composing these polynomials and applying a union bound, it follows that there exists a probabilistic polynomial $\mathbf{P}(\vec{x}) \stackrel{\text{def}}{=} \mathbf{Q}_g(\mathbf{Q}_1(\vec{x}), \ldots, \mathbf{Q}_t(\vec{x}))$ with $\mathsf{deg}(\mathbf{P}) \leq (\gamma \cdot \log s)^{d-1}$ that computes C with error at most $1/3$, where $\gamma = \gamma(p)$ is a fixed constant. Further, by raising this polynomial to $p-1$ and applying Fermat's little theorem, we can assume without loss of generality that its output is always Boolean. Since $d \geq 2$, the degree becomes at most $(\gamma' \cdot \log s)^{d-1}$, where $\gamma' \leq p \cdot \gamma$.

Now let $k = c \cdot \log(1/\delta)$, for a sufficiently large constant c. Consider the probabilistic polynomial $\mathbf{M}(\vec{x}) \stackrel{\text{def}}{=} \mathsf{M}(\mathbf{P}_1(\vec{x}), \ldots, \mathbf{P}_k(\vec{x}))$, where M is a degree k polynomial that computes $\mathsf{Majority}_k$ exactly, and each $\mathbf{P}_i$ is an independent copy of $\mathbf{P}$. It follows from Proposition A.1 that $\mathbf{M}$ is a probabilistic polynomial of degree at most $(\alpha \cdot \log s)^{d-1} \cdot \log(1/\delta)$ that computes C with error at most δ, where $\alpha = \alpha(\gamma', c) = \alpha(p)$ is an appropriate constant. □

ECCC ISSN 1433-8092

http://eccc.hpi-web.de