Circuits with Medium Fan-In

Pavel Hrubeš[*] Anup Rao[†]

February 17, 2014

Abstract

We consider boolean circuits in which every gate may compute an arbitrary boolean function of k other gates, for a parameter k. We give an explicit function $f : \{0,1\}^n \to \{0,1\}$ that requires at least $\Omega(\log^2 n)$ non-input gates when $k = 2n/3$. When the circuit is restricted to being depth 2, we prove the bound of $n^{\Omega(1)}$ on the fan-in of the top gate. When the circuit is a formula, we give a lower bound $\Omega(n^2/k \log n)$ on the total number of gates, for general k.

Our model is connected to some well known approaches to proving lower bounds in complexity theory. Optimal lower bounds for the Number-On-Forehead model in communication complexity, or for bounded depth circuits in AC_0, or extractors for varieties over small fields would imply strong lower bounds in our model. On the other hand, new lower bounds for our model would prove new time-space tradeoffs for branching programs and impossibility results for (fan-in 2) circuits with linear size and logarithmic depth. In particular, our lower bound gives a different proof for the best known time-space tradeoff for oblivious branching programs.

1 Introduction

A boolean circuit is usually defined as a directed acyclic graph where vertices (called gates) have in-degree (called fan-in) at most 2. Every gate with fan-in 0 corresponds to an input variable, and all other gates compute an arbitrary boolean function of the values that feed into them. Sometimes the model is restricted to using gates from the DeMorgan basis (i.e. AND, OR, NOT) gates, but this changes the size of the circuit by at most a constant factor. The circuit computes a function $f : \{0,1\}^n \to \{0,1\}$ if some gate in the circuit evaluates to f. A formula is a circuit whose underlying graph is a tree. The depth of the circuit is the length of the longest path in the graph.

Since every algorithm with running time $T(n)$ can be simulated by circuits of size $\tilde{O}(T(n))$, one can hope to prove lower bounds on the time complexity of algorithms by proving lower bounds on circuit size. A super-polynomial lower bound on the circuit size of an NP problem would imply that $\mathsf{P} \neq \mathsf{NP}$. However, we know of no explicit function (even outside NP) for which we can prove a super-linear lower bound. In contrast, counting arguments imply that almost every function requires circuits of exponential size.

We study a *stronger* model of circuits. We allow the gates to have fan-in k, where k is a parameter that depends on n, and each gate may compute an arbitrary function of its inputs. Typically, we consider the case where k is a constant fraction of n. We write $C_k(f)$ to denote the minimum number of *non-input* gates required to compute f in this model.

These circuits are much stronger than the models usually studied in the context of proving lower bounds. Nevertheless, we show that many attempts at proving lower bounds on other models of computation can be seen as proving new lower bounds in our model. Truly exponential lower bounds for AC_0, optimal lower bounds for the Number-On-Forehead (or NOF) model of communication, or new extractors for varieties over small fields, would all improve the best lower bounds we know how to prove for $C_{n/2}(f)$. On the other hand,

[*] `pahrubes@gmail.com`. Supported by the National Science Foundation under agreement CCF-1016565.

[†] `anuprao@cs.washingtonedu`. Supported by an Alfred P. Sloan Fellowship, the National Science Foundation under agreement CCF-1016565, an NSF Career award, and by the Binational Science Foundation under agreement 2010089.

lower bounds in our model could lead to lower bounds for branching programs and circuits of logarithmic depth. Our Theorem 1 already leads to a different proof of the best known lower bounds on oblivious branching programs of [BNS92]. We elaborate on these connections in Section 4.

A similar model has been studied in past work. Circuits with arbitrary gates and *arbitrarily large* fan-in have been considered for computing *several* boolean functions simultaneously. If n boolean functions are being computed, the trivial upper bound uses n^2 wires (edges). Super-linear lower bounds on the number of wires are known for circuits of bounded depth in this scenario. Cherukhin [Che05] proved a lower bound of $\Omega(n^{1.5})$ on the number of wires required in a depth-2 circuit, and bounds of around $n \log^2 n$ were obtained earlier using graph-theoretic arguments [PRS97, RTS00] (see also the survey in [Juk01]). No non-trivial bound is known for logarithmic depth. These results do not seem to give non-trivial lower bounds on $C_k(f)$.

Clearly, $C_n(f) = 1$, if f has n variables. However, when the fan-in is restricted, the power of circuits dramatically decreases. Counting arguments show that for almost every f, $C_k(f) > 2^{(n-k)-o(n-k)}$, which is exponentially large even for k linear in n. The challenge is to obtain such a lower bound for an explicit function f. If f depends on all its n inputs, it is easy to see that $C_k(f) \geq n/k$. When k is linear in n, this trivial lower bound is just a constant.

In [CFL83], Chandra, Furst and Lipton the Number-on-Forehead model of communication. They proved first non-trivial results in this model, and showed that *communication* lower bounds can be used to argue about complexity of *computation*. Specifically, they proved super-constant time-space tradeoffs for branching programs computing the majority function. The lower bound is obtained via Ramsey style argument and displays a tower-like decay. A similar reduction to the NOF model can be employed in our model, yielding super-constant lower bounds on $C_{2n/3}(\text{Majority})$ and host of other functions. However, NOF lower bounds can give a stronger result. First, we will start with a function for which strong communication lower are known, as given by Babai, Nisan and Szegedy [BNS92]. Second, we use a more sophisticated reduction. We will show: [1]

Theorem 1. *There exists $f \in \mathsf{P}$ such that for every $\gamma > 0$ and n large enough, $C_{n(1-\gamma)}(f) \geq \Omega(\gamma \log^2 n)$.*

The proof is reminiscent of the approach in Beame and Vee [BV02] and [BNS92] concerning trade-offs for branching programs. Our result is however stronger, since a small branching program can be simulated by a small circuit of linear fan-in.

Next we define a quantity which is closely related to $C_k(f)$. Let $C_k^2(f)$ denote the smallest number m such that there exist boolean functions $g, f_1, \dots, f_m$ with $f = g(f_1, \dots, f_m)$, where every f_i reads at most k inputs. We prove:

Theorem 2. *There exists $f \in \mathsf{P}$ such that $C^2_{(1-\gamma)n}(f) \geq \Omega(n^{c\gamma})$, $c > 0$.*

The proof of Theorem 2 involves ideas inspired by Nechiporuk's [Nec66] lower bound on boolean formula size, to whom we pay homage in the title of this paper. We show (Proposition 4) that $C_k^2(f) \leq C_k(f) \cdot 2^{C_k(f)}$ for every f, so Theorem 2 implies a lower bound of $\Omega(\gamma \log n)$ on $C_{n(1-\gamma)}(f)$. In fact, the specific f from Theorem 2 satisfies $C_{n/2}(f) \leq O(\log n)$, showing that $C^2_{n/2}$ can be exponentially larger than $C_{n/2}$.

Finally, we observe that Nechiporuk's original proof can be easily extended to formulas with large fan-in. Write $L_k(f)$ for the smallest number of *leaves* in a formula computing f with fan-in at most k. Nechiporuk gave an explicit function f for which $L_2(f) \geq \Omega(n^2/\log n)$. We note that this can be generalised to:

Theorem 3. *There exists $f \in \mathsf{P}$ such that $L_k(f) = \Omega(n^2/k \log n)$.*

Note that for formulas we are counting leaves and not just the non-input gates. Of course, Theorem 3 implies a lower bound of $\Omega(n^2/k^2 \log n)$ on the number of *non-input* gates as well.

Let us mention that the lower bound in Theorem 1 is stronger than stated. Consider circuits where the gates can have arbitrarily large fan-in, but each gate can read at most k input variables. Define $C_k^*(f)$ as the smallest number of non-input gates which read *some* input variable in a circuit computing f. Then

[1] Abusing notation, we write $f \in \mathsf{P}$ to mean that f is obtained by restricting a polynomial time computable function to n-bit inputs.

$C^*_k(f) \leq C_k(f)$. As far as our results go, this quantity is more natural – $C^*_k(f)$ is more closely related to transfer of information rather than actual computation. Indeed, the lower bounds on C_k given in Theorem 1, and the one implied by Theorem 2, apply indiscriminately to $C^\star_k$. This, however, points to a certain weakness in our arguments. $C^\star_k(f)$ can never exceed n; in order to prove super-linear lower bounds we need techniques which essentially distinguish C_k and $C^\star_k$.

In Section 2, we discuss the quantities C_k, C^2_k and $C^\star_k$ in greater detail. In Section 3, we give the proofs our lower bounds. In Section 4, we outline the connections between our model and other problems in complexity theory.

2 Circuits of medium fan-in

Before we embark on proving our main theorems, we make some general comments about our computational model.

As mentioned in the introduction, counting arguments show that for almost every f, $C_k(f) > 2^{(n-k)-o(n-k)}$. The bound is exponential even when k is very close to n, and super-linear even when $k < n - 1.1 \log n$. It becomes sub-linear when $k > n - \log n$. However, as we count the *non-input* gates, it is still non-trivial. For example, $C_{n-1.1\log\log n}(f) = \Omega(\log n)$ for most functions f.

The trivial upper bound on the quantity $C^2_k(f)$ is n. More interestingly, this estimate is quite robust: there exists an f for which $C^2_{\lfloor n-\log n-1\rfloor}(f) = n$. Indeed, the number of choices for the functions $g, f_1, \ldots, f_m$ is at most

$$2^{2^m}\left(\binom{n}{k}2^{2^k}\right)^m \leq 2^{2^m+m2^k+nm}.$$

In order to realize all n-variate functions, we must have $2^m + m2^k + nm \geq 2^n$. If $m = n-1$ and $k = \lfloor n - \log n - 1\rfloor$, the bound is

$$2^{n-1} + (n-1)2^{n-1}/n + n^2 = 2^n(1 - 1/(2n)) + n^2 < 2^n.$$

An exercise would show that if $\ell \leq \log n$, $C^2_{n-\ell}(f) \leq 2^\ell + \ell$, showing that C^2_k decreases when k goes above $n - \log n$.

Depth-2 circuits are arguably interesting in their own right. They can also serve as a tool to understand general circuits, via the follow proposition:

Proposition 4. $C^2_k(f) \leq C_k(f) \cdot 2^{C_k(f)}$. *(In fact, this holds for $C^\star_k$ instead C_k.)*

Proof. Let $u_1, \ldots, u_s$ be the non-input gates in a circuit of size $s = C_k(f)$ computing f. For every $i \in [s]$ and every $\sigma \in \{0,1\}^s$, we define a function $f_{i,\sigma}$ that depends on at most k input variables, as follows.

We interpret σ as an assignment of bit values to the s non-input gates of the circuit. $f_{i,\sigma}$ reads the input variables that are read by u_i, and outputs 1 if and only if there exists some setting of the remaining input variables that could result in the evaluation given in σ. Define g to be the function that reads the outputs of the $f_{i,\sigma}$'s and computes f by finding the unique σ for which $f_{i,\sigma} = 1$ for every i.

If a gate u_i does not read any variable then $f_{i,\sigma}$ is constant and can be thrown away, giving the statement for $C^\star_k$. □

Proposition 4 together with Theorem 2 already gives an $\Omega(\log n)$ lower bound on $C_{2n/3}(f)$. However,the exponential loss in the transformation means that even an optimal lower bound (of n) on depth-2 circuits would give at most a logarithmic lower bound for general circuits. Proposition 5 implies that the exponential loss is inevitable.

The trivial upper bound on $C^\star_k$ is n. Indeed, we have both $C^\star_k \leq C_k$ and $C^\star_k \leq C^2_k$. A random function satisfies $C^*_k(f) = \Omega(n-k)$. This is not clear from the definition, but can be seen as follows: in the proof of Proposition 4, the construction of g does not depend on the actual circuit but only on the size s. Hence, if $C^*_k(f) = s$, f is uniquely determined by $s2^s$ functions with fan-in k, which gives $s = \Omega(n-k)$ for a random f. As noted in the introduction, our lower bounds on C_k are in fact lower bounds on $C^\star_k$.

2.1 Communication complexity

In the Number-On-Forehead model of communication complexity [CFL83], there are p parties that are trying to compute a function $f(x^1, x^2, \dots, x^p)$, where each x^i is a n/p-bit string. The i'th party can see every input except x^i. To evaluate f, the parties exchange messages (by broadcast), until one of the parties can transmit the value of f to the others. The complexity of f is the number of bits the players need to exchange in order to evaluate f. The trivial upper bound is n/p. Currently the best lower bound is due to Babai, Nisan and Szegedy [BNS92], who proved that the generalized inner product function defined by

$$\mathsf{GIP}(x^1, \dots, x^p) = \sum_{i=1}^{n/p} \prod_{j=1}^{p} x_i^j \mod 2$$

requires $\Omega(n/2^{2p})$ bits of communication.

The most straightforward connection between circuits and the NOF model is the following observation: *Suppose that a circuit computing $f(x^1, \dots, x^p)$ has the property that for every gate u there is some $i \in [p]$ such that u reads no variable from x^i. Then, if the circuit has s non-input gates, the function f can be evaluated using s bits in the NOF model.*

This does not directly imply a circuit lower bound – in a circuit of linear fan-in, gates may access a constant fraction of each of the blocks x_i. For example, GIP can be computed by a constant size circuit with fan-in $n/2$ (imagine two gates, one reading the first half of every x^i, and the other the second half). Nevertheless, this issue can be partially circumvented, as in [CFL83] or in our Theorem 1, where we use the GIP function to obtain $C_{2n/3}(f) \geq \Omega(\log^2 n)$ for a related function f. Furthermore, let us note that an optimal $\Omega(n/p)$ lower bound in the NOF model would imply $C_{2n/3}(f) \geq \Omega\left(\sqrt{n}\right)$.

The main feature of the NOF model is the extreme overlap of information between the parties. It may be convenient to think of $C_k^\star$ in terms of a different communication game. The players want to evaluate $f(x)$, and each player has access to some 50% fraction of the input. The complexity of this game exactly corresponds to the quantity $C_{n/2}^\star$: with every non-input gate, we can associate a player who can see the inputs the gate reads. The circuit then provides a $C_{n/2}^\star$ protocol to evaluate f. Conversely, any such m-bit protocol can be interpreted as a circuit with at most m non-input gates reading a part of the input. In this perspective, the difference between $C_k^\star$ and C_k^2 is essentially the difference between an interactive and non-interactive communication protocol.

3 The lower bounds

3.1 The classical Nechiporuk method applied to $L_k(f)$

The proofs of Theorems 2 and 3 are variations of Nechiporuk's lower bound on formula size, which we now discuss. Given a boolean function f, a subset of its variables S, and a $0,1$-assignment σ to the variables in S, let f_σ be the function obtained by setting the variables in S to σ in f. An *S-subfunction of f* is a function of the form f_σ, where σ is a $0,1$-assignment to the variables in the complement of S. It is a function in the variables S; the number of S-subfunctions of f is clearly at most $\min(2^{2^{|S|}}, 2^{n-|S|})$, if f has n variables.

Nechiporuk finds a function f whose n-bit input is partitioned into intervals $x_1, x_2, \dots, x_{n/\log n}$ of size $\log n$ such that, for every i, f has $2^{\Omega(n)}$ x_i-subfunctions. An example to keep in mind is the element distinctness function:

$$f(x_1, \dots, x_{n/\log n}) = \begin{cases} 1 & \text{if } x_1, \dots, x_{n/\log n} \text{ are distinct} \\ 0 & \text{otherwise.} \end{cases}$$

Observe that whenever $\sigma_2, \dots, \sigma_{n/\log n}$ are distinct $\log n$-bit strings then $f(x_1, \sigma_2, \dots, \sigma_{n/\log n})$ rejects precisely on the inputs $\sigma_2, \dots, \sigma_{n/\log n}$. Hence f has at least $\binom{n}{n\log^{-1} n-1} = 2^{\Omega(n/\log n)}$ x_1-subfunctions, and likewise for any x_i. A slightly more complicated function would give $2^{\Omega(n)}$ subfunctions.

We now prove Theorem 3, which is a straightforward extension of the argument for $k = 2$ to general k. It is however noteworthy that the bound deteriorates only polynomially with k.

Claim. *Let S be a subset of variables of f. Assume that f can be computed by a formula with fan-in $\leq k$ in which m leaves are labelled with a variable from S. Then f has at most $2^{O(mk)}$ S-subfunctions.*

Proof. Given a formula computing f, take the tree T obtained as the union of all paths going from some variable in S to the output. First, we can assume that there is no path $u_1, \dots, u_p$ in T with $p > 2$ and $u_2, \dots, u_p$ having in-degree 1 in T. For then the value of u_p is determined by the value of u_1 and the variables in the complement of S. We can then replace u_p in our formula by a single binary gate which takes as input u_1 and a function not depending on S. This may increase the overall size of the formula, but leaves m unchanged. The tree T has m leaves and, as it now has few nodes with in-degree one, it has at most $4m$ nodes. Second, in order to define an S-subfunction, we only have to specify, for every gate v in T, the values of its inputs coming from outside of T. Since the fan-in is at most k, there will be altogether at most $4mk$ such gates and so f has at most 2^{4mk} S-subfunctions. □

Applying the claim to the function above, we obtain that every formula computing f contains $\Omega(n/k)$ leaves labelled with a variable from x_i, for every $i \in \{1, \dots, n/\log n\}$. This means that any such formula contains $\Omega(\frac{n^2}{k \log n})$ leaves altogether.

3.2 Proof of Theorem 2

In order to prove our theorem, we will find a function f that has a stronger property with regards to its subfunctions. Namely, f will have many functions not just for S coming from a fixed partition of the inputs; it will have many subfunctions for *almost every* $\log n$-element set.

We define our hard function as follows. $f(x, y)$ will take as inputs $x \in \{0,1\}^{n_0 + \log n_0}$ and a roughly $\log^2 n_0$-bit string y. We view y as representing a subset $S_y \subset [n_0 + \log n_0]$ of size $\log n_0$. For $S \subseteq [n_0 + \log n_0]$, let x_S be the projection of x to the coordinates in S. If $|S| = \log n_0$, we view the $\log n_0$-bit string x_S as an element of $[n_0]$. Furthermore, $x(S)$ will be the subset of $[n_0]$ represented by the bits of x in the complement of S, namely: $i \in x(S)$ iff the i-th bit of x_{S^c} equals 1. Then define

$$f(x,y) = \begin{cases} 1 & \text{if } x_{S_y} \in x(S_y), \\ 0 & \text{otherwise.} \end{cases}$$

f has $n = n_0 + O(\log^2 n_0)$ variables. Given a fixed y, an S_y-subfunction of $f(x, y)$ is uniquely determined by a subset of $[n_0]$. Hence:

Claim 1. *For every $\log n_0$-element subset S of the variables x, f has 2^{n_0} S-subfunctions.*

To prove Theorem 2, it will be enough to show that any small circuit gives an upper bound on the number of S-subfunctions of f, for some $\log n_0$ element subset of x. Suppose that

$$f = g(f_1, \dots, f_m)\,.$$

Let A_i be the set of input variables f_i reads and assume that $|A_i| \leq (1 - \gamma)n$ for every $i \in [m]$. First, we note that:

Claim 2. *If $n_0 > 100$ and $m < n^{c\gamma}/2$, where $0 < c < 1/2$ is a suitable absolute constant. Then there exists a $\log n_0$-element subset S of the variables x which satisfies $|S \cap A_i| \leq (1 - \gamma/2) \log n_0$ for every $i \in [m]$.*

Proof. Pick $\log n_0$ variables $a_1, \dots, a_{\log n_0}$ from x, y uniformly at random. With high probabilty, they will be distinct and they will completely miss the variables y; the probability being larger that $1/2$ if $n_0 > 100$. For a given set A of size $\leq (1 - \gamma)n$, let X be the random variable $\sum_{a_i \in A} 1$. The Chernoff bound gives,

$$\Pr\left[\frac{X}{\log n} \geq 1 - \gamma/2\right] \leq e^{-\mathsf{D}(1-\gamma/2||1-\gamma)\log n_0} < n_0^{-c\gamma}\,,$$

where $D(1-\gamma/2||1-\gamma) = \gamma/2\ln(1/2) + (1-\gamma/2)\ln((1-\gamma/2)/1-\gamma)$ is the Kullback-Leibler divergence. As γ approaches 0, the divergence becomes roughly $\gamma/2\ln(1/2) + \gamma/2 > 0.15\gamma$; as γ approaches 1 it goes to infinity. Hence we indeed have $D(1-\gamma/2||1-\gamma) \geq c'\gamma$ for some constant $c' > 0$ and every $\gamma \in (0,1)$. If $m < n_0^c/2$, the union bound gives that a $\log n_0$-element set satisfies $|S \cap A_i| \leq (1-\gamma/2)$ with positive probability. □

If $m \geq n_0^{c\gamma}/2$, we have a lower bound. Otherwise let S be the set promised by Claim 2. Hence, for every $i \in [m]$, the number of S-subfunctions of f_i is at most $2^{2^{|S\cap A_i|}} \leq 2^{n_0^{1-\gamma/2}}$. Furthermore, every S-subfunction of f is uniquely determined by S-subfunctions of $f_1, \ldots, f_m$, and hence f has at most $2^{n_0^{1-\gamma/2}m}$ S-subfunctions. By Claim 1, this means that $m \geq n_0^{\gamma/2}$. Hence, $C^2_{(1-\gamma)n}(f) \geq n_0^{c\gamma}/2 = \Omega(n^{c\gamma})$, proving Theorem 2.

By Proposition 4, this also means that $C_{(1-\gamma)n} = \Omega(\gamma \log n)$.

3.2.1 A Matching Upper Bound for $f(x,y)$

We will now show that the lower bound from Theorem 2 is tight for the function $f(x,y)$ defined above[2]. The principal merit of the following proposition, however, is in showing that the exponential gap between C_k and C_k^2 from Proposition 4 is inevitable.

Proposition 5. *There exists $c > 0$ such that for every $0 < \gamma < 1/2$ and n sufficiently large, $C^2_{(1-\gamma)n}f(x,y) \leq n^{c\gamma}$ and $C_{(1-\gamma)n}f(x,y) \leq c\gamma\log n$.*

Proof. It is enough to prove the bound for $C_{(1-\gamma)n}$ and invoke Proposition 4. We will outline the construction for $\gamma = 1/2$ and then sketch how to adapt it to the general case. Divide the variables x into two equal subsets x_1 and x_2. Let g_1 be the function which, on inputs x_1 and y, outputs a $\log n$-bit string whose first bits equal x_1 restricted to S_y. Define g_2 similarly. This means that x_{S_y} can be recovered from x_2, y and the advice from g_1; likewise for x_1, y and g_2. It is now easy to see that we can write $f(x,y) = h_1(g_1, x_2, y) \vee h_2(g_2, x_1, y)$ with suitable h_1 and h_2. This gives approximately $\log n$ gates with fan-in approximately $n/2$.

In general, partition the variables x into r disjoint subsets $a_1, \ldots, a_r$ of nearly the same size. The gates will have access to the inputs y and $x \setminus a_i$ for some $i \in [r]$. Note that for any $\log n_0$ element subset S of x, there will exist two distinct a_i and a_j with $|a_i \cap S|, |a_j \cap S| \leq 2\log n_0/r$. We can recover x_{S_y} from $x \setminus a_i$ with an advice of $2\log n/r$ bits, and as above, compute $f(x,y)$ using two gates depending $y, x \setminus a_i$ and $y, x \setminus a_j$ and $2\log n/r$ bits of advice each. The advice itself can be computed by gates which have access to either $y, x \setminus a_1$ or $y, x \setminus a_2$. This gives a circuit with roughly $r\log n + 1/r$ gates of fan-in $(1-1/r)n$; this is at most $cr\log n$ gates for fixed r and large enough n. □

3.3 Proof of Theorem 1

We will deduce a lower bound on $C_k(f)$ from known results in the NOF model. The main issue with the reduction to NOF model is that we do not apriori know which variables will the gates in a circuit read. One way to simulate any circuit with linear fan-in and m gates using m parties is to associate every gate with a party and then greedily assign variables to parties, giving inputs of length $\Omega(n/m)$ for each party. We manage to reduce the parties to $O(m/\log n)$, which helps us obtain stronger lower bounds. The essence is the following Lemma:

Lemma 6. *Let $G \subseteq A \times B$ be a bipartite graph with $|A| = m$, $|B| = n$ and with every $a \in A$ having degree at least γn, where $0 < \gamma < 1/100$ and n is sufficiently large with respect to γ^{-1}. If $\log n \leq m \leq \log^2 n$, then there exists $p \leq 5m/\gamma$ and disjoint $T_1, \ldots, T_p \subseteq A$, $S_1, \ldots, S_p \subseteq B$, each S_i of size at least $n^{0.9}$, such that $A = \bigcup T_i$ and $(T_i \times S_i) \subseteq G$ for every $i \in [p]$.*

Proof. We first prove the following:

[2] In the case when γ is fixed and n grows independently.

Claim. *Assume that G satisfies $m \leq \log n$ instead. Then G contains a complete bipartite graph with at least $\gamma m/2$ vertices on the left and $2n^{0.9}$ vertices on the right.*

Proof. Remove from B all vertices with degree $\leq \gamma m/2$. The remaining set B' has size at least $\gamma n/2$. For $M \subseteq A$, let $B(M)$ be the set of $b \in B'$ such that b is connected to every $a \in M$. Hence,

$$B' = \bigcup_{|M|=\lceil \gamma m/2 \rceil} B(M)\,.$$

Since $m \leq \log n$ and $\gamma < 1/100$, the number of sets with $|M| = \lceil \gamma m/2 \rceil$ is at most $n^{0.09}$. Hence there exists an M with $|M| = \lceil \gamma m/2 \rceil$ and $B(M) \geq 0.5\gamma n \cdot n^{-0.09} \geq 2n^{0.9}$, for n large enough. □

We will apply the Claim several times. If $m > \log n$, choose an arbitrary $\log n$-element subset of A and let $T_1 \times S_1$ be the complete graph guaranteed by the Claim. If $m \leq \log n$, apply the Claim directly to G. Remove from G all the vertices T_1 and S_1, obtaining a new graph $G_2 \subseteq A_2 \times B_2$. Repeat this process p times to obtain graphs $G_2, \ldots, G_p$ until $A_p = \emptyset$. We claim that $p \leq 5m/(\gamma \log n)$ First, for such a small p, we have altogether removed $o(n)$ vertices from B and so $|B_i| \geq n(1 - o(1))$. Similarly, the degree of any $a \in A_i$ is at least $\gamma |B_i|/2$. Hence, as long as $|A_i| \geq \log |B_i|$, we remove at least $\gamma \log n/4$ vertices from A_i. After at most $4m/(\gamma \log n)$ steps, we then must have $|A_i| < \log |B_i|$. After this point, A_i decreases by at least the factor of $(1 - \gamma/2)$, and so the size drops below 1 in roughly $\log\log n/\gamma$ steps, which is much smaller that m/γ. Finally, the size of every S_i is at least $|B_i|^{0.9} = 2(n(1 - o(1)))^{0.9} \geq n^{0.9}$, if n is large enough. □

Let us now take a function which is hard to evaluate in the NOF model, such as the generalized inner product mentioned in Section 2.1. Our hard function $f(x, y)$ is defined as follows. It takes as inputs $x \in \{0, 1\}^{n_0}$ and an auxiliary string y. We think of y as defining $k_y \leq \log n_0$ disjoint subsets $S_y^1, \ldots, S_y^{k_y}$ of $[n_0]$, of equal size not exceeding $n_0^{0.9}$. Hence, y can be taken as roughly $n_0^{0.9} \log^2 n_0$-bit string. We define

$$f(x, y) := \mathsf{GIP}(x_{S_y^1}, \ldots, x_{S_y^{k_y}})\,.$$

$f(x, y)$ has $n = n_0 + O(n_0^{0.9} \log^2 n_0)$ variables.

Suppose that for a fixed $0 < \gamma$ and n sufficiently large, $f(x, y)$ can be computed using $m < \gamma \log^2 n/50$ non-input gates with fan-in $n(1 - \gamma)$. Take the graph G whose left vertices are the m gates of the circuit and the right vertices the n_0 variables x. There is an edge between a gate and a variable iff the gate does *not* read the variable. Since y is much smaller than x, the degree of a gate in G is at least $\gamma n_0/2$. To apply the Lemma, we will assume $\gamma < 1/100$ (otherwise the circuit is weaker) and that $m \geq \log n_0$ (by possibly adding dummy gates). The Lemma shows that there exist disjoint sets of variables $S_1, \ldots, S_p$ with $p \leq \log n/5$ and $S_i = \lfloor n_0^{0.9} \rfloor$ such that each gate completely misses at least one set S_i. We can fix y so that y represents $S_1, \ldots S_p$ and hence $f(x, y)$ becomes $\mathsf{GIP}(x_{S_1}, \ldots, x_{S_p})$. As observed in Section 2.1, the circuit now gives an m-bit protocol for $\mathsf{GIP}(x_{S_1}, \ldots, x_{S_p})$. By the [BNS92] lower bound, this implies $m \geq \Omega(n_0^{0.9} 2^{-2 \log n_0/5}) = \Omega(\sqrt{n_0})$, contradicting the assumption $m < \gamma \log^2 n/50$. This proves Theorem 1.

4 Connections to Other Models

Here we show how is our model is connected to several disparate problems in complexity theory.

4.1 Circuits of Linear Size and Logarithmic Depth

Obviously, $C_k(f) \leq C_2(f)$, so any super-linear lower bound in our model would give a super-linear lower bound for circuits of fan-in 2. However, even a linear lower bound on our model would give a function that cannot be computed by a linear sized logarithmic depth circuit:

Proposition 7. *If f has a fan-in 2 circuit of linear size and logarithmic depth, then for any $\epsilon > 0$, $C_{n^\epsilon}(f) < O\left(\frac{n \log(1/\epsilon)}{\log \log n}\right)$.*

Valiant [Val77] showed that any (fan-in 2) circuit of linear size and logarithmic depth contains a set T of $O\left(\frac{n\log(1/\epsilon)}{\log\log n}\right)$ gates such that every path of length $\epsilon \log n$ in the circuit must touch a gate from the set. Since every such gate in T can be computed from at most n^{ϵ} other gates from T and the inputs, we obtain Proposition 7.

4.2 Oblivious Branching Programs

An oblivious branching program of width w and length ℓ is a directed graph with vertices partitioned into ℓ layers $L_1, \ldots, L_\ell$. Each layer is associated with an input variable. Every vertex in L_i has out-degree 2, with the edges labeled $0, 1$. Every vertex of L_ℓ is labeled with an output value. The program is executed by starting at the first vertex of L_1, and reading the variables in turn to find a path through the program until the output is determined.

Barrington [Bar89] showed that every logarithmic depth circuit (of fan-in 2) can be simulated by a branching program with $w = 5, \ell = \mathsf{poly}(n)$. Thus it is very interesting to prove super-polynomial lower bounds on such programs.

A line of work has proved *time-space tradeoffs* on such programs. Alon and Maass [AM86] used reductions to Ramsey theory to show that any program for computing the majority function must have $\ell \log w \geq \omega(n \log n)$. Babai, Nisan and Szegedy [BNS92] proved a lower bound of $\ell \log w \geq \Omega(n \log^2 n)$ by reductions to the Number-on-Forehead communication model. Beame and Vee [BV02] simplified the proof of this last bound. No better lower bound on $\ell \log w$ is known, to our knowledge.

Our results give lower bounds that match those of [BNS92] via the following proposition:

Proposition 8. *If f can be computed by an oblivious branching program of width $w < 2^{\epsilon n/2}$ and length ℓ, then $C_{\epsilon n}(f) \leq \frac{2\ell \log w}{\epsilon n}$.*

The first $\log w$ gates of the circuit read the first $\epsilon n/2$ variables read by the program and together compute the name of the vertex reached after those layers. The next $\log w$ gates read the outputs of the previous gates and the next $\epsilon n/2$ variables, to compute the name of the vertex in layer $L_{\epsilon n}$. Continue in this way until all of the program has been simulated. Thus we obtain a lower bound of $\ell \log w \geq \Omega(n \log^2 n)$ on the length of the program using Proposition 8 and Theorem 1. Any lower bound of the type $C_{\epsilon n}(f) = \omega(\log^2 n)$ would give new time-space tradeoffs for branching programs.

4.3 AC_0

An AC_0 circuit is a circuit that has constant depth, and gates of unbounded fan-in that compute functions from the DeMorgan basis. Any size s AC_0 circuit can be simulated by a size s^2 circuit with gates of fan-in 2.

Beautiful methods have been developed to prove lower bounds on these circuits [Has86, Raz87, Smo87]. The best known lower bounds for a depth d circuit are of the type $2^{\Omega(n^{1/d})}$. The following proposition shows that truly exponential lower bounds would give linear lower bounds in our model.

Proposition 9. *There is a depth-3 AC_0 circuit of size $kC_k(f) \cdot 2^{C_k(f)+k}$ computing f.*

To see this, observe that the g in the proof of Proposition 4 can be computed by a formula in disjunctive normal form of size $C_k(f) \cdot 2^{C_k(f)}$. Furthermore, each $f_{i,\sigma}$ depends on k variables, and so it can be computed by a formula in conjuctive normal form, or a disjunctive normal form, of size $k \cdot 2^k$.

Propositions 7 and 9 together imply that truly exponential lower bounds on the circuit size of a depth-3 AC_0 computing f would imply that f does not have linear sized circuit of logarithmic depth, an observation already made by Valiant [Val77].

4.4 Extractors/Dispersers for Varieties

Given a field $\mathbb{F}$, a variety is a set of the form $\{x \in \mathbb{F}^n : f_1(x) = f_2(x) = \ldots f_m(x) = 0\}$, where $f_1, \ldots, f_m$ are polynomials. For a finite field $\mathbb{F}$, an *extractor for varieties* is a function $f : \mathbb{F}^n \to \{0, 1\}$ which is non-constant on any sufficiently large variety defined by low-degree polynomials.

Dvir [Dvi12] showed how to use bounds on exponential sum estimates by Deligne [Del74] to obtain extractors for varieties. Working over a prime field of size p, he shows that if $\rho > 1/2$ is a constant, and $V \subseteq \mathbb{F}^n$ is a variety of size $p^{\rho n}$ defined by polynomials of degree ρn, then there is an efficiently computable extractor for such varieties, as long as p is polynomially large in n. Here we show that such a result for $p = 2$ would imply non-trivial circuit lower bounds.

Proposition 10. *Let $p = 2$. If f is an extractor for varieties of size $2^{\rho n}$ defined by degree k polynomials, then $C_k(f) > (1 - \rho)n$.*

Proof. Suppose there is a circuit computing f with m gates of fan-in k. By averaging, there must exist some evaluation of the gates which is consistent with 2^{n-m} input strings. We now define a variety using m polynomials as follows. Each polynomial checks that the input is consistent with the evaluations of a single gate. Since every such polynomial depends on at most k variables, and it can be taken multilinear, it has degree at most k. Thus we obtained variety of size 2^{n-m} defined by degree k polynomials on which f is constant. So it must be that $n - m < \rho n \Rightarrow m > (1 - \rho)n$. □

By Proposition 7, any such extractor cannot be computed by linear sized logarithmic depth circuits of fan-in 2.

References

[AM86] Noga Alon and Wolfgang Maass. Meanders, ramsey theory and lower bounds for branching programs. In *FOCS*, pages 410–417. IEEE Computer Society, 1986.

[BNS92] László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci*, 45(2):204–232, 1992.

[Bar89] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. Syst. Sci*, 38(1):150–164, 1989.

[BV02] Paul Beame and Erik Vee. Time-space tradeoffs, multiparty communication complexity, and nearest-neighbor problems. In John H. Reif, editor, *STOC*, pages 688–697. ACM, 2002.

[CFL83] A. Chandra, M. Furst, and R. Lipton. Multi-party protocols. In *In Proceedings of the fifteenth annual ACM symposium on Theory of computing, STOC*, pages 94–99, 1983.

[Che05] D. Y. Cherukhin. The lower estimate of complexity in the class of schemes of depth 2 without restrictions on a basis. *Moscow University Mathematics Bulletin*, 60(4):42–44, 2005.

[Del74] Pierre Deligne. La conjecture de weil : I, 1974.

[Dvi12] Zeev Dvir. Extractors for varieties. *Computational Complexity*, 21(4):515–572, 2012.

[Has86] Johan Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 6–20, Berkeley, California, 28–30 May 1986.

[Juk01] S. Jukna. *Extremal combinatorics, with applications in computer science.* Springer-Verlag, 2001.

[Nec66] E. I. Nechiporuk. A boolean function. *Sov.Math.Dokl.*, 7(4):999–1000, 1966.

[PRS97] P. Pudlák, V. Rödl, and J. Sgall. Boolean circuits, tensor ranks, and communication complexity. *SIAM J. Comput.*, 26(3):605–633, 1997.

[RTS00] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discrete Math. 13(1): 2-24*, 13(1):2–24, 200.

[Raz87] Alexander Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *MATHNASUSSR: Mathematical Notes of the Academy of Sciences of the USSR*, 41, 1987.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 77–82, New York City, 25–27 May 1987.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In Jozef Gruska, editor, *MFCS*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.