A Super-Quadratic Lower Bound for Depth Four Arithmetic Circuits

Nikhil Gupta
Indian Institute of Science
nikhilg@iisc.ac.in

Chandan Saha
Indian Institute of Science
chandan@iisc.ac.in

Bhargav Thankey
Indian Institute of Science
thankeyd@iisc.ac.in

February 27, 2020

Abstract

We show an $\tilde{\Omega}(n^{2.5})$ lower bound for general depth four arithmetic circuits computing an explicit $n$-variate degree-$\Theta(n)$ multilinear polynomial over any field of characteristic zero. To our knowledge, and as stated in the survey [SY10], no super-quadratic lower bound was known for depth four circuits over fields of characteristic $\neq 2$ before this work. The previous best lower bound is $\tilde{\Omega}(n^{1.5})$ [Sha17], which is a slight quantitative improvement over the roughly $\Omega(n^{1.33})$ bound obtained by invoking the super-linear lower bound for constant depth circuits in [Raz10, SS97].

Our lower bound proof follows the approach of the almost cubic lower bound for depth three circuits in [KST16] by replacing the shifted partials measure with a suitable variant of the projected shifted partials measure, but it differs from [KST16]'s proof at a crucial step – namely, the way “heavy” product gates are handled. Loosely speaking, a heavy product gate has a relatively high fan-in. Product gates of a depth three circuit compute products of affine forms, and so, it is easy to prune $\Theta(n)$ many heavy product gates by projecting the circuit to a low-dimensional affine subspace [KST16, SW01]. However, in a depth four circuit, the second (from the top) layer of product gates compute products of polynomials having arbitrary degree, and hence it was not clear how to prune such heavy product gates from the circuit. We show that heavy product gates can also be eliminated from a depth four circuit by projecting the circuit to a low-dimensional affine subspace, unless the heavy gates together account for $\tilde{\Omega}(n^{2.5})$ size. This part of our argument is inspired by a well-known greedy approximation algorithm for the weighted set-cover problem.
1 Introduction

The arithmetic circuit model is naturally well-suited for the study of optimality of algorithms for algebraic and linear algebraic problems. An arithmetic circuit consists of addition (+) and multiplication (×) gates, it takes input \{x_1, x_2, \ldots, x_n\} and field scalars, and computes a polynomial in \{x_1, x_2, \ldots, x_n\}. Size of a circuit is the number of wires in it, and depth is the longest path from an input to an output gate. The two complexity measures – size and depth – of a circuit essentially capture the sequential and parallel complexity of the computation happening inside the circuit.

Arithmetic circuits are weaker\(^1\) than Boolean circuits: An explicit lower bound on the size of Boolean circuits implies an explicit lower bound on the size of arithmetic circuits over finite fields\(^2\) and also over fields of characteristic zero\(^3\), but the converse is not true\(^4\). Still, the best known lower bound for arithmetic circuits is \(\Omega(n \log n)\) [BS83, Str73], which is barely super-linear. For arithmetic formulas, an \(\Omega(n^2)\) lower bound is known [Kal85]. Several other better lower bounds have been shown in the past few decades for various restricted models of arithmetic circuits (see Section 1.3 for a brief history of known lower bounds). But, very few lower bounds are known for circuit models that do not impose restrictions like homogeneity, multilinearity, monotonicity, bounded coefficients, bounded reads etc. One such result is the lower bound for constant depth circuits. Shoup and Smolensky [SS97], and Raz [Raz10], showed an \(\Omega(\Delta n^{1+\frac{1}{\Delta}})\) lower bound for depth-\(\Delta\) circuits, where \(\Delta = O(\log n)\). In the special case of depth three circuits, an \(\Omega(n^2)\) lower bound was shown in [SW01], which was improved to an \(\Omega(n^3)\) lower bound in [KST16]. However, for depth four circuits\(^5\), the best lower bound was \(\bar{\Omega}(n^{1.5})\) [Sha17], which is a slight quantitative improvement over the roughly \(\Omega(n^{1.33})\) lower bound obtained by specializing the constant depth lower bound in [SS97,Raz10] to depth four circuits.

In this work, we show an \(\bar{\Omega}(n^{2.5})\) lower bound for depth four circuits. To the best of our knowledge, and as stated in the survey [SY10] (Section 1.4.2, page-13), this is the first super-quadratic lower bound for this model over fields of characteristic \(\neq 2\).

\(\footnote{\text{1}This is not to mean that arithmetic circuits cannot simulate Boolean circuits. It is easy to see that a Boolean circuit can be efficiently simulated by an arithmetic circuit over any field that contains the additive and the multiplicative identities 0 and 1 respectively.}
\footnote{\text{2}This is because an arithmetic circuit over a finite field can be simulated by a Boolean circuit with only a slight blow-up in size and depth (see [vzGS91]).}
\footnote{\text{3}It was shown in [Bür00] (Corollary 4.6 in Chapter 4) that NC\(^3\)/poly \(\neq\) NP/poly implies VP \(\neq\) VNP over fields of characteristic zero, assuming the Generalized Riemann Hypothesis. The circuit classes VP and VNP are arithmetic analogues of non-uniform P and NP respectively [Val79].}
\footnote{\text{4}as computing a Boolean function is a weaker requirement than computing a specific polynomial representation of the function.}
\footnote{\text{5}Depth four circuits form a natural circuit class as the “optimal” circuit for an arbitrary polynomial turns out to be a depth four circuit: The multiplicative complexity of a polynomial \(f\), denoted \(M(f)\), is the minimum number of multiplication gates required to compute \(f\). It is known that there exists an \(n\)-variate degree-\(d\) polynomial \(f\) for which \(M(f) = \Omega(\sqrt{\frac{n+d}{d}})\) [CKW11, HY11]. On the other hand, every \(n\)-ivariate degree-\(d\) polynomial can be computed by a depth four circuit having \(\sqrt{\frac{n+d}{d}} \cdot \text{poly}(n,d)\) many multiplication gates [Lov11].}
1.1 Our Result

We state our result formally now. Without loss of generality, we will assume that a depth four circuit is a $\Sigma \Pi \Sigma \Pi$ circuit, i.e., the circuit has a $+$-gate on top followed by second layer of $\times$-gates, then a third layer of $+$-gates and finally a bottom layer of $\times$-gates.

**Theorem 1** (Lower bound for depth four circuits). Over any field of characteristic zero$^6$, there exists a family of multilinear polynomials $\{f_n\}_{n\geq 1}$ in $\text{VNP}$, where $f_n$ is a polynomial in $\Theta(n)$ variables and of degree $\Theta(n)$ such that any depth four circuit computing $f_n$ has $\Omega\left(\frac{n^{2.5}}{(\log n)^6}\right)$ many wires/edges and $\Omega\left(\frac{n^{1.5}}{(\log n)^4}\right)$ many gates.

A word about the polynomial family. The polynomial family $\{f_n\}_{n\geq 1}$ is a variant of the Nisan-Wigderson design polynomial family, which has been used in proving several other previous lower bounds [Raz10, KSS14, KLSS17, KS17, KS16a, KS16b, KST16, Sha17]. The $n$-th member of the family, i.e., $f_n$ is a polynomial in $x = \{x_1, ..., x_{3m}\}$ and $y = \{y_1, ..., y_{3m}\}$ variables, where $m$ is an integer in $[n^2, 2n]$. The degree of the polynomial in $y$ variables is $\deg_y(f_n) = m$, while its degree in $x$ variables is $\deg_x(f_n) = d_x = \Theta(\sqrt[m]{\ln m})$. Informally, $f_n$ contains multiple ‘copies’ of the design polynomial in different subsets of the $x$ variables, while the $y$ variables are used as ‘prefixes’ to uniquely identify each such copy. While the reason for having multiple copies is similar to [KST16], as we shall see in the next section, handling them is a little trickier in our case. Note that because of the way we have defined $m$ and $d_x$, proving that any depth four circuit computing $f_n$ has $\Omega\left(\frac{m^2d_x}{(\ln m)^6}\right)$ many edges and $\Omega\left(\frac{md_x}{(\ln m)^3}\right)$ many gates would establish the theorem.$^7$ The exact description of $f_n$ is given in Section 4.

1.2 Proof Idea

Let $C$ be a depth four circuit over a field $\mathbb{F}$. Like many other works on arithmetic circuit lower bounds, we use a rank based complexity measure to obtain our result. The measure we apply is a variant of the projected shifted partials measure, which has been used before in [KLSS17, KS17, KS16a, Sha17] and other works. Our proof can be divided into four steps; the first three show a ‘small’ upper bound on the measure of $C$ while the last step shows a ‘large’ lower bound on the measure of the hard polynomial $f_n$ described above. We now briefly outline each of these steps.

Step 1: Restricting the bottom support of $C$. We begin by removing all monomials computed by the bottom layer of $C$ that have a ‘large’ support. Such restrictions have been used in previous works [KS14, KLSS17, KS17, KS16a, Sha17] to control the degree of the (sparse) polynomials computed at the third layer of a depth four circuit so that a ‘small’ upper bound on the measure of the circuit can be obtained. However, in our work, this step plays an even more significant role by enabling us to remove ‘heavy’ gates (see below). While previous works use random restrictions, we use a deterministic procedure for restricting the bottom support – we briefly explain our reasons for doing so towards the end of this section.

---

$^6$The lower bound holds even if the characteristic is sufficiently large (see Section 4).

$^7$We use log base $e$ in the proof rather than log base 2 as it simplifies the analysis.
Heavy gates. We call a product gate in the second layer of $C$ heavy if the number of distinct third layer gates (computing sparse polynomials) feeding into it is $\Omega(n^{1.5})$. The presence of heavy gates makes the task of obtaining a ‘small’ upper bound on the measure of $C$ difficult. The problem of dealing with heavy gates was also faced by previous works on depth three circuits [SW01,KST16], and was dealt with by removing all heavy gates from the circuit before applying the measure to it. We too remove all heavy gates from $C$, but our way of doing so differs from [SW01,KST16]. Since a depth three circuit computes a sum of product of affine forms, [SW01,KST16] were able to remove all heavy gates by going modulo affine factors of these gates thereby restricting the circuit to an affine subspace. While going modulo the sparse factors of heavy gates is a natural generalization of this technique for depth four circuits, we do not know how to adopt this method as the quotient ring so obtained might not be a polynomial ring. In the next step, we outline a technique of removing heavy gates from $C$ which, in spirit, also restricts $C$ to an affine subspace.

Step 2: Removing heavy gates from $C$. We remove heavy gates from $C$ by sequentially evaluating exactly one sparse factor of each heavy gate to zero. This can be done if $F$ is algebraically closed, which one can assume without loss of generality (as argued in Section 5). In particular, we use the following greedy procedure: While there exists a heavy gate in $C$, pick a sparse polynomial for which the ratio of the number of heavy gates connected to it to its fan-in is maximum, and evaluate it to zero. Intuitively, this allows us to remove a large number of heavy gates at the cost of evaluating a few monomials (computed by the bottom layer) to field constants. Then, as we have restricted the bottom support of $C$ in Step 1, we are able to show that we can remove $\Theta(n)$ many heavy gates at a cost of setting only a few variables to constants (unless the already removed heavy gates account for $\tilde{\Omega}(n^{2.5})$ size). Note that $\Theta(n)$ many heavy gates would immediately imply the desired lower bound. This greedy procedure is inspired by an approximation algorithm for the weighted set cover problem [Vaz01] (Section 2.1, page-16), however its analysis here is tailored to our needs. This step, which is the main contribution of this work, plays a crucial role in enabling us to prove a super-quadratic lower bound and we provide more details about it in Section 3.2.2.

Step 3: Analyzing the measure of $C$. After all the heavy gates have been removed, a ‘small’ upper bound on the measure of $C$ is derived by closely following the “grouping” argument made in [KST16]. However, we replace the shifted partials measure by the projected shifted partials measure as the latter is suitable for controlling the degree of the sparse polynomials computed at the third layer of $C$. In this step, we divide the factors of a polynomial $T$ computed by a $\times$-gate in the second layer of $C$ into ‘groups’ of suitable sizes, and multiply out factors in the same group to reduce the effective number of factors of $T$. This then helps obtain a ‘small’ upper bound on the projected shifted partials measure of $T$ which, by sub-additivity, implies a ‘small’ upper bound on the projected shifted partials measure of $C$.

Step 4: Lower bound on the measure of $f_n$. The choice of the hard polynomial $f_n$ is dictated by the above technique of removing heavy gates. As mentioned before, $f_n$ has multiple copies of the Nisan-Wigderson design polynomial, denoted NW. The reason for having multiple copies is that if we work with only one copy, we might end up irreparably damaging it while removing heavy gates from $C$. On the other hand, starting with multiple copies of NW, much like in [KST16], we are able to show that the procedure for removing heavy gates leaves an intact copy along with some ‘damaged’ parts from the other copies. Our use of a deterministic restriction in Step
1 makes it easier to show this. A ‘large’ lower bound on the measure of NW was obtained in [KLS17, KS17, KS16a]. However, it is not clear how to obtain such a lower bound on NW in the presence of other ‘damaged’ parts, and so we remove these parts. Although in [KST16], such parts were removed by simply setting a subset of variables to 0 and 1, here we need to augment the projected shifted measure appropriately to get rid of them.

1.3 Known lower bounds

We give a brief account of known lower bounds for some of the important classes of arithmetic circuits by drawing parallels with similar results from the Boolean circuit literature. The reader may refer to the surveys [SY10, CKW11] or the book [BCS97] for more details on arithmetic circuit lower bounds. We also state a few hardness magnification or amplification results to show that proving seemingly modest lower bounds can be quite interesting and challenging even for constant depth circuits, provided the polynomials being computed have “low” complexity.

**General circuits.** The best known lower bound for general arithmetic circuits is $\Omega(n \log d)$, which was obtained nearly four decades ago for circuits computing the power symmetric polynomial $x_1^d + x_2^d + \ldots + x_n^d$ [BS83, Str73a]$^9$. Recently, [CKSV19] has shown an $\Omega(n^2)$ lower bound for ‘layered’ algebraic branching programs (ABPs) computing the same polynomial. An $\Omega(n^2)$ lower bound for formulas computing the polynomial $\sum_{i,j \in [n]} x_i^j y_j$ was shown in [Kal85]. The situation is similar for Boolean circuits. For circuits over the DeMorgan basis and over the full binary basis, the lower bounds $5n - o(n)$ [LR01, IM02, KMM12] and $(3 + \frac{1}{86})n - o(n)$ [FGHK16, Blu84] respectively, are the best till date. For Boolean formulas, an $\tilde{\Omega}(n^2)$ lower bound is known over the full binary basis [Nec66], and an $\tilde{\Omega}(n^3)$ lower bound is known over the DeMorgan basis [Hås98, And87].

**Monotone circuits.** These are arithmetic circuits over $\mathbb{Q}$ or $\mathbb{R}$ that disallow negation. A lot more is known about this class of circuits. A near optimal $2^{\Omega(n)}$ lower bound on the monotone circuit complexity of the $n \times n$ permanent was shown in [JS82]. In fact, [Val80] showed an exponential separation between monotone and general circuits computing the perfect matching polynomial of a certain planar graph. Optimal separations are also known between monotone ABPs and monotone circuits [HY16] and between monotone formulas and monotone ABPs [SS77]. Recently, [Yeh19] gave an exponential separation between monotone VP and monotone VNP which was made stronger in [Sri19]. One of the success stories on Boolean circuit complexity in the 80s is the exponential lower bound for monotone circuits computing the clique function [Raz85b, AB87]. Building on these results, [Tar88] showed an exponential separation between monotone and general Boolean circuits. An exponential separation between monotone switching networks and monotone circuits was given in [RPRC16]$^{12}$. A separation between monotone formulas and monotone switching networks follows from the work of [GS95]. Yet another interesting result is the

---

8Borrowing terminology from [OS18].
9The bound also holds for the $d$-th elementary symmetric polynomial in $n$ variables.
10Prior to this, [Raz85a] showed a quasi-polynomial lower bound for monotone circuits computing the perfect matching function.
11We may think of monotone switching networks as the Boolean analogue of monotone ABPs.
12In fact, [RPRC16] gave an exponential lower bound for the more powerful model of monotone span programs that was introduced in [KW93].
separation of monotone-NC$^i$ from monotone-NC$^{i+1}$ for every $i \leq 1$ [RM99, KW90].

**Non-commutative and multilinear circuits.** A non-commutative circuit computes a polynomial in non-commuting variables. This model has more structure than circuits over commuting variables and so one may hope that it is “easier” to prove lower bounds for non-commutative circuits. The seminal work of [Nis91] showed an exponential separation between non-commutative ABPs and non-commutative circuits. But, proving a super-polynomial lower bound for general non-commutative circuits and showing a separation between non-commutative formulas and non-commutative ABPs continue to remain two important open problems. The techniques used to prove lower bounds for non-commutative circuits is closely related to that used to prove lower bounds for multilinear circuits. In a (syntactically) multilinear circuit, the sets of variables occurring in the subcircuits rooted at the children of a product gate are pairwise disjoint. It is an interesting model of computation as most natural families of polynomials, like the permanent, determinant, iterated matrix multiplication, elementary symmetric polynomials, design polynomials etc., are multilinear. Building on [RSY08], an $\Omega\left(\frac{n^2}{\log n}\right)$ lower bound for multilinear circuits has been recently shown in [AKV18]. Prior to this, the breakthrough work of [Raz09] culminated in an optimal separation between multilinear formulas and multilinear ABPs [DMPY12, RY08, Raz06]. We do not know of any super-polynomial lower bound for multilinear ABPs.

**Bounded coefficient circuits.** In a bounded coefficient circuit over $\mathbb{C}$, we forbid any multiplication by a field element having absolute value larger than 1. An $\Omega(n \log n)$ lower bound for bounded coefficient circuits computing the Discrete Fourier Transform of a vector $(x_1, \ldots, x_n)$ was shown in [Mor73]. Later, [Raz03] gave an $\Omega(n \log n)$ lower bound for the same class of circuits computing the product of two $\sqrt{n} \times \sqrt{n}$ matrices.

**Read-$k$ circuits.** A read-once oblivious algebraic branching program (ROABP) is a layered ABP in which every layer is indexed by a variable, and every variable indexes exactly one layer. The edges of a layer are labeled by univariate polynomials in the variable indexing the layer. In a certain sense, the ROABP model generalizes quite a few other interesting arithmetic circuit models, especially tensors. An exponential lower bound for ROABPs follows from the technique introduced in the work of [Nis91]. A read-$k$ oblivious ABP is defined similarly, with every variable indexing at most $k$ layers. In [AFS+18], an $\exp\left(\frac{n}{k \log(k)}\right)$ lower bound for read-$k$ oblivious ABP was shown. Another related result is the $\exp\left(\frac{n}{k^2}\right)$ lower bound for depth four read-$k$ formulas [ASSS16]. An exponential lower bound is also known for read-$k$ Boolean branching programs [BRS93], which is a stronger model than read-$k$ oblivious Boolean branching programs.

**Constant depth circuits**

The best lower bound for constant depth circuits is a little better than that for general circuits. Shoup and Smolensky [SS97] showed an $\Omega(\Delta n^{1+\frac{1}{\Delta}})$ lower bound for depth-$\Delta$ circuits computing

---

13Besides, there is an interesting connection between the non-commutative determinant and the commutative permanent [AS18].

14In fact, nothing better than the $\Omega(n \log d)$ lower bound (which also holds for commutative circuits) is known. The hardness of proving a sufficiently strong super-linear lower bound for non-commutative circuits is explained in a recent work [CILM18] (see the discussion below on hardness magnification).
the polynomials \( \{ \sum_{j \in [n]} x_j^j y_j, \ldots, \sum_{j \in [n]} x_j^j y_j \} \). Using an argument similar to [SS97], Raz [Raz10] showed a roughly \( \Omega(n^{1+\frac{1}{\Delta}}) \) lower bound for depth-\( \Delta \) circuits computing polynomials of degree \( \Theta(\Delta) \), i.e., the polynomials have constant degree for constant depth circuits. These bounds were essentially achieved by analyzing linear circuits. Prior to these works, a barely super-linear lower bound of \( \lambda_\Delta(n) \) was known for depth-\( \Delta \) linear circuits using super-concentrators [Val75, DDPW83, Pud94, RS03], where \( \lambda_\Delta(n) \) is a very slowly growing function. Recently, [KV19] gave a \( n^{1+\frac{1}{\Delta}} \) lower bound for depth-\( \Delta \) linear circuits computing a linear transformation that can be computed in \( \exp(n^{1-O(\frac{1}{\Delta})}) \) time. A similar lower bound was known before for bounded coefficient circuits – Pudlák [Pud00] proved an \( \Omega(n^{\frac{1}{1+\frac{1}{\Delta}}}) \) lower bound for depth-\( \Delta \) bounded coefficient circuits computing the DFT matrix. A lower bound of \( \Omega(n^{1+\frac{1}{\Delta^{1+\frac{1}{\Delta}}}}) \) was also shown in [Raz03] for depth-\( \Delta \) bounded coefficient circuits computing the product of two \( \sqrt{n} \times \sqrt{n} \) matrices.

A lot better lower bounds are known for constant depth multilinear circuits. Raz and Yehudayoff [RY09] showed an \( \exp(n^{O(\frac{1}{\Delta})}) \) lower bound for depth-\( \Delta \) multilinear circuits computing the \( n \times n \) determinant polynomial. More recently, [CL19] showed an \( \exp(n^{\frac{1}{1+\Delta}}) \) lower bound for depth-\( \Delta \) multilinear circuits computing the product of \( n \) many \( 2 \times 2 \) matrices. In fact, an exponential separation is known between depth-\( \Delta \) and depth-(\( \Delta + 1 \)) multilinear circuits [CELS18], which improved upon a previous quasi-polynomial separation [RY09].

A circuit computing an \( n \)-variate homogeneous polynomial of \( \operatorname{poly}(n) \) degree can be homogenized with only a polynomial blow-up in size [Str73b], but this process is not depth preserving. It is plausible that homogeneous depth-\( \Delta \) circuits are weaker than general depth-\( \Delta \) circuits for constant \( \Delta \). Indeed, such a statement is known to be true for \( \Delta = 3 \) and \( \Delta = 4 \). It was shown in the classical work [NW97] that any homogeneous depth three circuit computing the \( n \)-variate degree-\( d \) elementary symmetric polynomial \( \operatorname{ESym}_{n,d} \) has size \( n^{\Omega(d)} \), although \( \operatorname{ESym}_{n,d} \) has a non-homogeneous (multilinear) depth three circuit of size \( \operatorname{O}(n^2) \). A sequence of work [KS17, KLSS17, FLMS15, KSS14, GKK14, Kay12] culminated in a \( n^{\Omega(\sqrt{d})} \) lower bound for homogeneous depth four circuits computing the width-\( n \), degree-\( d \) iterated matrix multiplication polynomial \( \operatorname{IMM}_{n,d} \). On the other hand, the depth reduction result in [GKK16, Tav15], which built on [Koi12, AV08, VSB13], yields a non-homogeneous depth four circuit of size \( n^{\Omega(d^{\frac{3}{2}})} \) for \( \operatorname{IMM}_{n,d} \).

An almost cubic lower bound for general depth three circuits was shown in [KST16], which improved upon the previous quadratic bound [SW01]. The bound in [SW01] is for the elementary symmetric polynomial, whereas the bound in [KST16] is for a variant of the Nisan-Wigderson design polynomial (which is in VNP). Subsequently, [BLS16, Yau16] showed near cubic lower bounds for depth three circuits computing polynomials that have \( \operatorname{poly}(n) \)-size depth five circuits. Over fixed finite fields, an exponential lower bound is known for depth three circuits computing the

---

15 A linear circuit has only addition gates and so it computes a linear transformation (i.e., a set of linear forms) in the input variables. If a set of linear forms is computable by a circuit of size \( s \) and depth-\( \Delta \) then they are computable by a linear circuit of size \( \Omega(s) \) and depth-\( \Delta \). Thus, a super-linear lower bound for linear circuits implies a super-linear lower bound for general circuits.

16 For instance, \( \lambda_\Delta(n) = \log^* n \).

17 Construction of this circuit is attributed to Michael Ben-Or.
determinant [GK98, GR00]. For depth three circuits with low bottom fan-in\(^{18}\) (but without any homogeneity restriction), [KS16a] proved an exponential lower bound\(^{19}\). As mentioned before, the previous best lower bound for general depth four circuits is \(\Omega(n^{1.5})\) [Sha17], which is a slight improvement over the roughly \(\Omega(n^{1.33})\) bound obtained by specializing the lower bound for constant depth circuits in [SS97, Raz10] to depth four circuits\(^{20}\). Our work here improves these super-linear bounds to a super-quadratic lower bound for depth four circuits.

Now, coming to constant depth Boolean circuits, an exponential lower bound is known for constant depth Boolean circuits over the DeMorgan basis (i.e., \(AC^0\) circuits). However, it appears to us that the ‘right’ Boolean analogue of constant depth arithmetic circuits is \(TC^0\) circuits (see [AAD00, RT92, HAB02])\(^{21}\). The exponential lower bounds for \(AC^0\) circuits [FSS84, Ajt83, Hås86] and the quasi-polynomial lower bounds for \(ACC^0\) circuits\(^{22}\) [Wil14, MW18] are two of the great achievements in Boolean circuit complexity, but we are yet to see these kind of bounds for \(TC^0\) circuits. The best known lower bound for threshold circuits is the slightly super-linear \(n^{1+\frac{c}{\Delta}}\) bound for depth-\(\Delta\) \(TC^0\) circuits computing the parity function, where \(c > 1\) is a fixed constant [IPS97]. For depth two \(TC^0\) circuits, [KW16] showed an \(\Omega(n^{2.5})\) lower bound on the number of wires and an \(\Omega(n^{1.5})\) lower bound on the number of gates.

**Hardness magnification**

There are results in the arithmetic and Boolean circuit literature that show how to obtain strong lower bounds from seemingly weak ones. We state a few of these results below with the intent of demonstrating that sufficiently strong super-linear or super-quadratic lower bounds can be quite interesting and challenging to prove even for constant depth circuits.

It follows from [Str73b] that a cubic form\(^{23}\) has a depth three powering circuit\(^{24}\) with \(\Theta(s)\) gates if it has a circuit of size \(s\). Thus, a super-linear lower bound on the number of gates of a depth three powering circuit computing an explicit cubic form implies a super-linear circuit lower bound. Stated differently, a super-linear lower bound on the (symmetric) tensor rank of an explicit (symmetric) tensor of order 3 implies a super-linear circuit lower bound\(^{25}\). In fact, Raz [Raz13] showed that an \(n^{r(1-o(1))}\) lower bound on the tensor rank of an explicit order-\(r\) tensor \(T : [n]^r \rightarrow \mathbb{F}\) implies a super-polynomial formula lower bound, assuming \(r\) is a super-constant and \(r \leq \log \log n\).

\(^{18}\)The depth reduction in [GKKS16] yields a depth three circuit with low bottom fan-in. This is reminiscent of a result in [Val77], which showed that a strong exponential lower bound for depth three Boolean circuits with low bottom fan-in implies a super-linear lower bound for Boolean circuits having logarithmic depth and bounded fan-in.

\(^{19}\)This result was extended to depth four circuits with low bottom fan-in in the works [KS16b, KS15].

\(^{20}\)For the reader’s convenience, we show how the \(\Omega(n^{1.33})\) bound can be derived from [SS97, Raz10] in Appendix C.

\(^{21}\)This is because iterated addition and multiplication of integers are in \(TC^0\), and in the converse direction, it is known that \(TC^0\) circuits can be simulated by constant depth arithmetic circuits using a single threshold gate [AAD00]. A related fact (attributed to Michael Ben-Or) is that there is an \(O(n^3)\) size depth three arithmetic circuit computing the \(n\)-variate degree-\(\frac{n}{2}\) elementary symmetric polynomial, which is the arithmetic analogue of the majority function.

\(^{22}\)also for \(ACC^0\) circuits composed with a bottom layer of threshold gates

\(^{23}\)i.e., a homogeneous degree-3 polynomial

\(^{24}\)A depth three powering circuit has a top + gate, a middle layer of powering gates, and a bottom layer of + gates.

\(^{25}\)The best known lower bound for an explicit order-3 tensor \(T : [n]^3 \rightarrow \mathbb{F}\) is roughly \(3n\) and for an explicit order-\(r\) tensor is roughly \(2n^{r+1}\) [AFT11].
Recently, Chen and Tell [CT19] showed that a super-linear lower bound for TC⁰ circuits (computing certain NC¹-complete functions) which is slightly better than the lower bound in [IPS97] would imply TC⁰ \neq NC¹. Their result builds on the work of [AK10]. By mimicking the argument in [CT19] for arithmetic circuits one gets the following statement: If IMM₂ⁿ is computable²⁶ by a depth-Δ₀ circuit of size \( n^k \) then it is computable by a depth-Δ circuit of size \( O\left(\frac{\Delta}{\Delta_0} \cdot n^{1+\exp\left(-\frac{n}{\exp^{1.8} n}\right)}\right) \). Recall that an \( O(\Delta n^{1+\frac{1}{\Delta}}) \) lower bound is already known for depth-Δ arithmetic circuits [Raz10, SS97]. If the same lower bound is shown for depth-Δ circuits computing IMM₂ⁿ then that would imply a super-polynomial lower bound for constant depth arithmetic circuits! Compare this with the best known upper bound for depth-Δ circuits computing IMM₂ⁿ which is roughly \( \exp(O(\Delta n^{\frac{1}{\Delta}})) \). Even for depth three circuits, we get the following interesting observation: An \( \Omega(n^{1.8+\epsilon}) \) lower bound on the number of gates of a depth five circuit computing IMM₂ⁿ, for any constant \( \epsilon > 0 \), implies a super-cubic lower bound for depth three circuits²⁷.

Hardness amplification results are also known for non-commutative circuits [HWY10, CILM18]. A biquadratic polynomial \( f \) in the variables \( x = \{x_1, \ldots, x_n\} \) and \( y = \{y_1, \ldots, y_n\} \) is a homogeneous degree-4 polynomial in which every monomial is of the form \( x_i x_j y_k y_l \). The bilinear complexity of a biquadratic polynomial \( f \) is the minimum \( r \) such that \( f = g_1 h_1 + \ldots + g_r h_r \), where \( g_i, h_i \) are bilinear forms in \( x \) and \( y \). It was shown in [HWY10] that Permanent requires non-commutative circuits of exponential size if there is an explicit biquadratic polynomial having bilinear complexity \( \Omega(n^{1+r}) \), for some constant \( \epsilon > 0 \). In other words, a super-cubic lower bound on the size of a homogeneous depth four (commutative) circuit computing an explicit biquadratic form implies an exponential lower bound for non-commutative circuits. In another appealing instance of hardness amplification, [CILM18] showed that an \( \Omega(n^{\frac{\omega^2}{\omega+1}}) \) lower bound, where \( \omega \) is the matrix multiplication exponent, for non-commutative circuits computing an explicit constant degree polynomial implies an exponential lower bound for non-commutative circuits; if the explicit polynomial has \( \text{poly}(n) \) degree then the lower bound is an arbitrarily large polynomial function.

## 2 Preliminaries

**Notations.** For \( r \in \mathbb{N} \), \([r] := \{1, \ldots, r\}\). We use lowercase Greek alphabets like \( \alpha, \beta \) for field constants, bold-face lowercase letters like \( x \) and \( y \) to denote sets of variables, \( f, g \) for polynomials in \( F[x, y] \), uppercase lowercase letters \( C, D \) for polynomials over \( F \), uppercase Roman alphabets \( T, Q \) for the polynomials computed by the gates of a depth four circuit and \( M, M_1, M_2 \) for subsets of natural numbers. For \( M \subseteq [3m] \), \( x_M := \{x_i : i \in M\} \), \( y_M := \{y_i : i \in M\} \). For \( z \subseteq x_M \cup y_M \) and \( r \in \mathbb{N} \), \( z^{\leq \infty} \) and \( z^{\leq r} \) denote the set of all monomials in \( z \) variables and the set of all monomials in \( z \) variables with degree at most \( r \) respectively. For \( S \subseteq F[x, y] \), \( \dim(S) \) denotes the dimension of the \( F \)-linear span of \( S \).

**Support and degree of a monomial.** The support of a monomial \( \eta \), denoted Supp(\( \eta \)), is the set of variables appearing in it. Also, for any \( z \subseteq x \cup y \) we will use \( \deg_z(\eta) \) to denote its degree in \( z \) variables. We will say that \( \eta \) is \( z \)-multilinear if the degree of every \( z \) variable in \( \eta \) is at most one.

²⁶Here \( \text{IMM}_2^n \) is a collection of four polynomials corresponding to the entries of a product of \( n \) many \( 2 \times 2 \) matrices whose entries are distinct formal variables.

²⁷We attribute this observation to Ankit Garg.
2.1 The complexity measure

Throughout this section, we will assume that $m \in \mathbb{N}$ is as stated in the paragraph following Theorem 1, $M \subseteq [3m]$, $|M| = m$, $f \in \mathbb{F}[x_M, y_M]$ and $S \subseteq \mathbb{F}[x_M, y_M]$. Note that the set $M$ is not fixed and will depend on the circuit under analysis. Before defining the measure, let us define the operations that make up the measure.

1. **Partial derivatives.** Let $\eta = x_1 \cdots x_k$ be a monomial in $x$ variables. Then, we define the partial derivative of $f$ with respect to $\eta$ as

$$\frac{\partial f}{\partial \eta} := \frac{\partial}{\partial x_1} \left( \frac{\partial}{\partial x_2} \left( \cdots \left( \frac{\partial f}{\partial x_k} \right) \right) \right).$$

If the degree of $\eta$ is $k$, then $\frac{\partial f}{\partial \eta}$ is said to be a $k$-th order partial derivative of $f$. We denote by $\partial^k f$ the set of all $k$-th order partial derivatives of $f$ taken with respect to multilinear monomials in $x$ variables.

2. **The shift operation.** Let $\eta$ be a degree $\ell$ multilinear monomial in $x_M$ variables. We say that the polynomial $\eta \cdot f$ is obtained by shifting $f$ by $\eta$. We denote by $x_M f$ the set of polynomials obtained by shifting $f$ by all degree $\ell$ multilinear monomials in $x_M$ variables and $x_M S := \{ x_M f : f \in S \}$.

3. **Multilinear projection.** We define a map $\pi_x : \mathbb{F}[x_M, y_M] \to \mathbb{F}[x_M, y_M]$ with $\pi_x(f)$ being the polynomial made up of exactly the $x$-multilinear monomials of $f$. Formally, for a monomial $\eta$, $\pi_x(\eta) = \eta$ if $\eta$ is $x$-multilinear and 0 otherwise. The map is then linearly extended for arbitrary polynomials and $\pi_x(S) := \{ \pi_x(f) : f \in S \}$.

4. **A degree based projection.** For $i \in \mathbb{N}$ and $f \in \mathbb{F}[x_M, y_M]$, we define $[f]_i$ to be the polynomial made up of only those monomials of $f$ whose $y$-degree is exactly $i$. Formally, for a monomial $\eta$, $[\eta]_i = \eta$ if $\deg_y(\eta) = i$ and 0 otherwise. It is then linearly extended for arbitrary polynomials and $[S]_i := \{ [f]_i : f \in S \}$.

5. **An evaluation map.** For $a \in \mathbb{F}$ and $z \subseteq x_M \cup y_M$, we define a map $\sigma_{z=a} : \mathbb{F}[x_M, y_M] \to \mathbb{F}[x_M \setminus z, y_M \setminus z]$ with $\sigma_{z=a}(f)$ being obtained from $f$ by setting every variable in $z$ to $a$ and $\sigma_{z=a}(S) := \{ \sigma_{z=a}(f) : f \in S \}$.

The operations given in 1, 2 and 3 constitute the projected shifted partials measure [KLSS17]. In this work, we define and use the measure $\operatorname{PSP}_{M,k,\ell}$, which is obtained by augmenting the projected shifted partials measure with the operations in 4 and 5 as follows.

**Definition 2.1 (The measure).** For $m, k, \ell \in \mathbb{N}$, $M \subseteq [3m]$, $|M| = m$ and $f \in \mathbb{F}[x_M, y_M]$,

$$\operatorname{PSP}_{M,k,\ell}(f) := \dim \left( \langle \sigma_{y_M=1} \left( \left[ \pi_x \left( x_M \partial^k \partial_x f \right) \right]_m \right) \rangle \right).$$

**Observation 2.1 (Sub-additivity of the measure).** For any two polynomials $f, g \in \mathbb{F}[x_M, y_M]$,

$$\operatorname{PSP}_{M,k,\ell}(f + g) \leq \operatorname{PSP}_{M,k,\ell}(f) + \operatorname{PSP}_{M,k,\ell}(g).$$

The above observation is easy to prove and we omit its proof here.
2.2 Some numerical estimates

Proposition 1 (Estimating Binomial Coefficients). For any \( n, k \in \mathbb{N}, k \leq n \), \( \binom{n}{k} \leq \left( \frac{e n}{k} \right)^k \).

Proposition 2 ([GKKS14, KLSS17]). Let \( a(n), f(n), g(n) : \mathbb{Z}_{>0} \to \mathbb{Z} \) be integer valued functions such that \( (|f| + |g|) = o(a) \). Then, \( \ln \left( \frac{(a+f)!}{(a-g)!} \right) = (f + g) \ln(a) + O \left( \frac{f^2 + g^2}{a} \right) \).

3 Upper bounding the measure for a depth four circuit

Let \( C = \sum_{i=1}^{d} T_i \) be a depth four circuit computing the polynomial \( f = f_n \) (recall \( \deg_x(f_n) = d_x = \Theta \left( \frac{\sqrt{m}}{\ln m} \right) \) from Section 1.1); \( T_i = \prod_{j=1}^{q} Q_{ij} \) and \( Q_{ij}s \) are distinct sparse polynomials computed by the + gates in the third layer of \( C \), and \( e_i \geq 1 \). We assume that \( \mathbb{F} \) is algebraically closed and argue in Section 5 why this holds without loss of generality. For brevity, we would use the terminologies ‘product terms’, ‘sparse polynomials’ and ‘monomials’ for the \( + \)-gates, \( + \)-gates and \( \times \)-gates in the second, third and fourth layers of \( C \) respectively as shown in Figure 1a. The proof of the upper bound is divided into three steps:

**Step 1: Restricting the bottom support.** In this step, we show that if \( C \) has fewer than \( \Omega \left( \frac{m^2 d_x}{(\ln m)^{p+2}} \right) \) distinct monomials then we can remove all monomials with support more than \( \tau = \lfloor 20 \ln m \rfloor \) at a cost of setting \( m \) many \( x \) variables and \( m \) many \( y \) variables to zero. (Notice that if there are more than \( \Omega \left( \frac{m^2 d_x}{(\ln m)^{p+2}} \right) \) many monomials then there is nothing to prove.) This step is required not only to remove heavy gates in Step 2 but also in Step 3 where using the fact that all monomials have support at most \( \tau \) and multilinear projection, we will argue that the degree of all monomials is not too large. More details about this restriction are given in Section 3.2.1.

**Step 2: Removing heavy gates.** The transformed circuit \( C_1 \), obtained after Step 1, computes a polynomial in the remaining \( 2m \) many \( x \) variables and \( 2m \) many \( y \) variables. Moreover, the number of gates and the fan-in of all gates in \( C_1 \) is upper bounded by the number of gates and the fan-in of the corresponding gates in \( C \). A product term in \( C_1 \) is called a heavy gate if at least \( w = \left\lfloor \frac{m d_x}{\lambda_0 (\ln m)^{p+2}} \right\rfloor \) \( (\lambda_0 \) is a large enough constant fixed in Appendix B) many distinct sparse polynomials are connected to it. If there are more than \( m \) heavy gates, we are done. Otherwise, we remove all heavy gates using the following greedy procedure: While there is a heavy gate, evaluate a sparse polynomial that would kill the most number of heavy gates at the cost of evaluating as few monomials as possible. As we have already restricted the support of all monomials to \( \tau \), we are able to argue in Section 3.2.2 that this can be done at a cost of setting \( m \) many \( x \) and \( m \) many \( y \) variables to field constants.

These steps transform \( C \) to a ‘pruned circuit’, defined as follows and depicted in Figure 1b.

**Definition 3.1.** We say that a depth four circuit \( D \) is a pruned circuit if the support of all monomials in \( D \) is at most \( \tau = \lfloor 20 \ln m \rfloor \), and it does not contain any heavy gate; i.e. the number of distinct sparse polynomials feeding into any product term in \( D \) is less than \( w = \left\lfloor \frac{m d_x}{\lambda_0 (\ln m)^{p+2}} \right\rfloor \).

**Step 3: Analysing the measure.** In this step, we analyse the measure \( \text{PSP}_{M,k,f} \) of the pruned circuit \( D \), obtained after Steps 1 and 2, computing a polynomial in the remaining \( m \) many \( x \) and \( m \) many \( y \) variables. More details on this analysis are provided in the following section.
3.1 Upper bound on the measure of a pruned depth four circuit

Recall the definition of the measure $\text{PSP}_{M,k,\ell}$ from Section 2. In Steps 1 and 2, we will ensure that if a variable $x_i$ is set to a field constant then $y_i$ is also set to a field constant and vice versa. The set $M$ is then the set of indices of the remaining $x$ (or $y$) variables and $|M| = m$.

**Lemma 3.1.** Let $D$ be a pruned depth four circuit with top fan-in $s$ computing a polynomial in $x_M$ and $y_M$ variables, where $M \subseteq [3m], |M| = m$. Also, let $d_x, \tau, w$ be as defined earlier, $t = \left\lfloor \frac{d_x}{(\ln m)^3} \right\rfloor, \delta = \frac{1}{(\ln m)^3}, k = \left\lfloor \frac{\delta d_x}{T} \right\rfloor$ and $\ell = \left\lfloor \frac{m}{m^\ell + 1} \right\rfloor$. Then, for sufficiently large $m$,

$$\text{PSP}_{M,k,\ell}(D) \leq s \cdot m^{O(1)} \left( \frac{m}{\ell + 2k\tau} \right)^\left( \left\lfloor \frac{\ell}{T} \right\rfloor + k - 1 \right).$$

We prove the lemma at the end of this section. As $D = T_1 + \cdots + T_s$, where $T_i$ is a product term and as $\text{PSP}_{M,k,\ell}$ is sub-additive, to prove the lemma it suffices to show that for all $i \in [s],$

$$\text{PSP}_{M,k,\ell}(T_i) \leq m^{O(1)} \left( \frac{m}{\ell + 2k\tau} \right)^\left( \left\lfloor \frac{\ell}{T} \right\rfloor + k - 1 \right).$$

Consider any such product term $T = \prod_{i \in [a]} Q_i^{e_i}$, where $Q_i \in \mathbb{F}[x_M, y_M]$, and since $D$ is a pruned depth four circuit, $a \leq w$. Write $Q_i = Q_i' + Q_i''$, where $Q_i'$ is the sum of all monomials of $Q_i$ wherein the individual degree of every $x$ variable is at most two and $Q_i'' = Q_i - Q_i'$. Then,

$$T = \prod_{i \in [a]} (Q_i' + Q_i'')^{e_i} = \prod_{i \in [a]} Q_i^{e_i} + Q'',$$

where $Q''$ is a polynomial whose every monomial has a $x$ variable with degree at least three. Thus, $\text{PSP}_{M,k,\ell}(Q'') = 0$ and hence from the sub-additivity of $\text{PSP}_{M,k,\ell}$ we have that

$$\text{PSP}_{M,k,\ell}(T) \leq \text{PSP}_{M,k,\ell}\left( \prod_{i \in [a]} Q_i^{e_i} \right).$$
Let $T' = \prod_{i \in [r]} Q_i^{e_i}$. We will now upper bound $\text{PSP}_{M,k,t}(T')$. First, we assume without loss of generality that $a = w$ since if $a < w$ then we can multiply with additional sparse polynomials all of which are 1. Next we divide the sparse polynomials into disjoint sets such that each set (except perhaps the last) has size exactly $t$. Then, we have that
\[
T' = P_1 \cdots P_{\lceil \frac{w}{\ell} \rceil}, \text{ where } P_i = \prod_{j=(i-1)t+1}^{\min(it,w)} Q_j^{e_j}.
\]

Claim 3.1. Let $P = Q_1^{e_1} \cdots Q_r^{e_r}$ be one of the polynomials $P_i$. For $k \geq 0$, let $P^{(k)} := \prod_{i \in [t]} Q_i^{\max(e_i-k,0)}$. Then, $\partial_x^k P \subseteq \mathbb{F}\text{-span}\{y_M^{\leq \ell} x_M^{\leq (2\ell t-1)} P^{(k)}\}$.

The proof of the claim is straightforward and is given in Appendix A.1.

Proof of Lemma 3.1. Recall that it is enough to show the following
\[
\text{PSP}_{M,k,t}(T') \leq m^{O(1)} \left( \frac{m}{\ell + 2kt} \right)^{\left( \left\lceil \frac{w}{\ell} \right\rceil + k - 1 \right)\choose k},
\]
where $T' = P_1 \cdots P_{\lceil \frac{w}{\ell} \rceil}$. Let $v = \left\lceil \frac{w}{\ell} \right\rceil$. Now,
\[
\partial_x^k T' \subseteq \mathbb{F}\text{-span}\left\{ \partial_x^{k_1} P_1 \cdots \partial_x^{k_v} P_v : k_1 + \cdots + k_v = k \right\}
\subseteq \mathbb{F}\text{-span}\left\{ y_M^{\leq \ell} x_M^{\leq (2\ell t-1)} P^{(k)} : k_1 + \cdots + k_v = k \right\}
\subseteq \mathbb{F}\text{-span}\left\{ y_M^{\leq \ell} x_M^{\leq (2\ell t-1)} P^{(k)} : k_1 + \cdots + k_v = k \right\},
\]
where the second to last inclusion follows from Claim 3.1. Hence,
\[
x_M^{\ell} \partial_x^k T' \subseteq \mathbb{F}\text{-span}\left\{ y_M^{\leq \ell} x_M^{\leq (2\ell t-1)} P^{(k)} : k_1 + \cdots + k_v = k \right\}.
\]

In other words, the space of shifted partials of $T'$ is contained in the $\mathbb{F}$-span of polynomials of the form $Y \cdot X \cdot P_1^{(k_1)} \cdots P_v^{(k_v)}$ where $Y$ is a monomial in $y_M$ variables and $X$ is a monomial in $x_M$ variables of degree at most $\ell + k(2\ell t - 1)$. Let us analyse the effect of the operations $\sigma_{y_M=1}$, $[\cdot]_m$, and $\pi_x$ on one such polynomial. We will assume that $\deg(Y) \leq m$ and $X$ is multilinear for otherwise the polynomial will vanish after the operations are applied. Then, we have that
\[
\sigma_{y_M=1} \left( \left[ \pi_x \left( Y \cdot X \cdot P_1^{(k_1)} \cdots P_v^{(k_v)} \right) \right]_m \right) = X \cdot \sigma_{y_M=1} \left( \left[ \pi_x \left( \sigma_{\text{Supp}(X)=0} \left( P_1^{(k_1)} \cdots P_v^{(k_v)} \right) \right) \right]_{m-\deg(Y)} \right).
\]

Thus,
\[
\sigma_{y_M=1} \left( \left[ \pi_x \left( x_M^{\ell} \partial_x^k T' \right) \right]_m \right) \subseteq \mathbb{F}\text{-span}\left\{ X \cdot \sigma_{y_M=1} \left( \left[ \pi_x \left( \sigma_{\text{Supp}(X)=0} \left( P_1^{(k_1)} \cdots P_v^{(k_v)} \right) \right) \right]_{m-\deg(X)} \right) : X \text{ is a multilinear monomial in } x_M \text{ variables, } \deg(X) \text{ is at most } \ell + k(2\ell t - 1), 0 \leq i \leq m \text{ and } k_1 + \cdots + k_v = k \right\}.
\]
Once we fix $i$, $X$, and $k_1, \ldots, k_v$, $X \cdot \sigma_{y=1} \left( \left[ \tau_x \left( \sigma_{\text{Supp}(X)=0} \left( P^{(k_1)}_1 \cdots P^{(k_v)}_v \right) \right) \right]_i \right)$ is fixed. So,

$$
PSP_{M,k_1}(T') = \dim \left( \sigma_{y=1} \left( \left[ \tau_x \left( x^{k_1}_M \partial^k_x T' \right) \right]_{m_1} \right) \right)
$$

$$
\leq (m + 1) \cdot \sum_{j=0}^{\ell+2k \tau - 1} \binom{m}{j} \binom{\ell}{k} \binom{\ell + k - 1}{v + k - 1}
$$

$$
= m^{O(1)} \cdot \binom{m}{\ell + 2kt \tau} \binom{\left\lfloor \frac{\ell + k \tau}{2} \right\rfloor + k - 1}{k}
$$

where the second last inequality follows from Claim 3.2 (proved in Appendix A.1).

\[ \square \]

Claim 3.2. Let $\ell, k, t$ and $\tau$ be as defined earlier. Then, $\ell + 2kt \tau < \frac{m}{2}$.

### 3.2 Pruning a depth four circuit

As mentioned before, we will prune the circuit $C$ computing $f_n$ in two steps - first we will restrict the bottom support of $C$ and then we will get rid of all heavy gates in it.

#### 3.2.1 Step 1 - Restricting the bottom support of $C$:

If the number of monomials in $C$ is more than $\left\lfloor \frac{m^2 \delta_x}{(\ln m)^5} \right\rfloor$, there is nothing to prove. Otherwise, we show that we can get rid of all monomials with support more than $\tau = \left\lfloor 20 \ln m \right\rfloor$.

Lemma 3.2. Let the number of monomials in $C$ be at most $\left\lfloor \frac{m^2 \delta_x}{(\ln m)^5} \right\rfloor$. Then, for sufficiently large $m$, there exists $M_1 \subseteq [3m], |M_1| = m$ such that all monomials in $C_1$ obtained from $C$ by setting variables $x_{M_1}$ and $y_{M_1}$ to 0 have support at most $\tau$.

Proof. We first present a greedy procedure to remove all monomials with support more than $\tau$ and then argue that it sets $m$ variables each from $x$ and $y$ to 0. In each iteration the procedure picks a pair of variables that appears in a large number of monomials with support more than $\tau$ and set them to 0.

Procedure 1 Restriction procedure

1. $M_1 \leftarrow \emptyset$, $C_1 \leftarrow C$, $H :=$ set of all monomials of $C_1$ with support more than $\tau$.
2. For $j \in [3m], e(j) :=$ number of monomials in $H$ containing $x_j$ or $y_j$.
3. while $H \neq \emptyset$ do
   4. Pick $j' \in [3m] \setminus M_1$ such that $e(j') \geq e(j)$ for all $j \in [3m]$. Set $x_{j'} = 0$ and $y_{j'} = 0$. Update $M_1 \leftarrow M_1 \cup \{j'\}$, $C_1$ circuit obtained from $C_1$ by setting $x_{j'}$ and $y_{j'}$ to 0, $H \leftarrow$ set of all monomials of $C_1$ with support more than $\tau$, and $e(j) \leftarrow$ number of monomials in $H$ containing $x_j$ or $y_j$.
5. end while
It is clear that the bottom support of $C_1$ obtained after the termination of the procedure is at most $\tau$. Also, since we are only setting variables to 0, it trivially follows that the procedure does not increase the number of gates nor does it increase the fan-in of any gate in the circuit. Claim 3.3 (proved in Appendix A.2) implies that the procedure terminates in at most $m$ iterations. If it terminates before $m$ iterations, we arbitrarily add an appropriate number of $j \in [3m]$ to $M_1$ so that $|M_1| = m$ and set $x_j$ and $y_j$ to 0 for all such $j$.

\[ \square \]

**Claim 3.3.** Procedure 1 terminates in at most $m$ iterations.

**Remark.** Procedure 1 looks similar to an approximation algorithm for the Set Cover problem [WS11] (Section 1.2, page-6). This is because the problem of removing monomials with support more than $\tau$ can be formulated as an instance of Set Cover with the universe being all such monomials and with a set corresponding to each $j \in [3m]$. For a $j \in [3m]$, the corresponding set will contain all monomials with support more than $\tau$ in which at least one of $x_j$ and $y_j$ appears.

3.2.2 Step 2 - Pruning the heavy gates from $C_1$

A sparse polynomial $Q$ in $C_1$ is said to be light if its fan-in is at most $\frac{m}{\ln m}$, i.e., at most $\frac{m}{\ln m}$ many non-zero monomials are present in $Q$. If the sum of fan-ins of all the light sparse polynomials is $\Omega \left( \frac{m^2d}{\ln m} \right)$ then there is nothing to prove. So assume that the sum of fan-ins of all the light sparse polynomials is $O \left( \frac{m^2d}{\ln m} \right)$. Recall that a product term is called heavy if it has at least $w = \left\lfloor \frac{md}{\lambda_0 \ln m} \right\rfloor$ many distinct sparse polynomials connected to it (where $\lambda_0$ is a large enough constant fixed in Appendix B). Observe that one of the following cases is true:

1. There is a heavy gate in $C_1$, that is connected to at most $\frac{m d}{2 \lambda_0 \ln m}$ light sparse polynomials.
2. Every heavy gate is connected to at least $\frac{m d}{2 \lambda_0 \ln m}$ light sparse polynomials.

Case 1 clearly implies a lower bound of $\Omega \left( \frac{m^2d}{\ln m} \right)$. Else, we prove the following lemma.

**Lemma 3.3.** Let $C_1$ be the circuit (obtained from Lemma 3.2) having at most $m$ heavy gates such that every heavy gate is connected to at least $\frac{md}{2 \lambda_0 \ln m}$ light sparse polynomials, and the sum of fan-ins of all the light sparse polynomials is at most $\frac{m^2d}{160 \lambda_0 \ln m}$. Then, there exist $M_2 \subseteq [3m] \setminus M_1$, $|M_2| = m$ and $\alpha_l, \beta_l \in \mathbb{F}$ for $l \in M_2$, such that setting $x_l = \alpha_l, y_l = \beta_l$ for all $l \in M_2$ removes all heavy gates from $C_1$.

**Proof.** We first present the pruning procedure and then argue its correctness. For any light sparse polynomial $Q_j$ in $C_1$, let $b_j$ and $c_j$ be equal to the fan-in of $Q_j$ and the number of distinct heavy gates connected to $Q_j$ in $C_1$ respectively. As $Q_j$ is a light sparse polynomial, $b_j \leq \frac{m}{\ln m}$. In this procedure, while all the heavy gates do not disappear from $C_1$, we pick a sparse polynomial whose ratio of the number of distinct heavy gates connected to it and fan-in is maximum and set that to zero by evaluating it to one of its roots in $\mathbb{F}$. This can be done as $\mathbb{F}$ is algebraically closed. The restricted support of the monomials in $C_1$ ensure that at the end of this procedure, only a small fraction of the variables are set.
We now describe the family \( \{ f_n \}_{n \geq 1} \), whose \( n \)-th member \( f_n \) is a polynomial in variables \( x = \{ x_1, ..., x_{3m} \} \) and \( y = \{ y_1, ..., y_{3m} \} \), where \( m \in \left[ \frac{q}{r}, 2n \right] \) will be fixed later.

\[
 f_n := \sum_{S \subseteq [3m], |S| = m} \left( \prod_{i \in S} y_i \right) \cdot NW_r(x_S),
\]

where \( NW_r \) is a variant of the Nisan-Wigderson design polynomial (introduced in [KSS14]), the construction of which is described later and \( r \) is a parameter fixed in this construction. Note that \( \{ f_n \}_{n \geq 1} \) is in VNP. Given a monomial, in order to find its coefficient in \( f_n \), we first check if the monomial is multilinear and of degree \( m \) in \( y \) variables. If it is so and \( S \) is the set of the indices of the \( m \) many \( y \) variables in the monomial then simply return the coefficient of the part of the monomial in \( x \) variables in \( NW_r(x_S) \) – this can be done as the Nisan-Wigderson polynomial family
is in VNP.

Let $M_1$ and $M_2$ be as in Section 3 and $M = [3m] \setminus (M_1 \cup M_2)$. Let $f_1$ be the polynomial computed by the pruned circuit $D$, which is obtained from $f = f_n$ by setting the variables $x_M$ and $y_M$ to field constants as in Section 3.2. Let us now see how $\text{PSP}_{M,k,\ell}(f_1)$ is related to $\dim \langle \pi_x \left( x_M^{\ell} \partial_x^k \text{NW}_r(x_M) \right) \rangle$.

**Lemma 4.1.** Let $f_1$ be as defined above. Then, $\text{PSP}_{M,k,\ell}(f_1) = \dim \langle \pi_x \left( x_M^{\ell} \partial_x^k \text{NW}_r(x_M) \right) \rangle$.

**Proof.** The proof follows easily from the following two observations:

1. The two operations in $y$ variables and the three operations in $x$ variables (in the definition of $\text{PSP}_{M,k,\ell}$) commute. That is, we have
   $$\sigma_{y_M=1} \left( \left[ \pi_x \left( x_M^{\ell} \partial_x^k f_1 \right) \right]_m \right) = \pi_x \left( x_M^{\ell} \partial_x^k \left( \sigma_{y_M=1} ([f_1]_m) \right) \right).$$

2. $f_1 = (\prod_{i \in M} y_i) \cdot \text{NW}_r(x_M) + f'$, where $f' \in \mathbb{F}[x_M, y_M]$ and $\deg_y(f') < m$.

From these observations we have that

$$\text{PSP}_{M,k,\ell}(f_1) = \dim \langle \pi_{y_M=1} \left( \left[ \pi_x \left( x_M^{\ell} \partial_x^k f_1 \right) \right]_m \right) \rangle$$

$$= \dim \left\langle \pi_x \left( x_M^{\ell} \partial_x^k \left( \sigma_{y_M=1} \left( \left[ \left( \prod_{i \in M} y_i \right) \cdot \text{NW}_r(x_M) + f' \right]_m \right) \right) \right) \right\rangle$$

$$= \dim \left\langle \pi_x \left( x_M^{\ell} \partial_x^k \text{NW}_r(x_M) \right) \right\rangle.$$  

The last equality follows from the fact that $(\sigma_{y_M=1} ([\prod_{i \in M} y_i] \cdot \text{NW}_r(x_M) + f')_m) = \text{NW}_r(x_M)$. 

**Construction of NW$_r$.** Let $d_x = \left\lceil \sqrt{\frac{r}{\ln n}} \right\rceil$. Pick an $\alpha$ such that $d_x \left\lceil d_x^{1+\alpha} \right\rceil \leq n \leq 2d_x \left\lceil d_x^{1+\alpha} \right\rceil$; this forces $\alpha$ to be $\Theta(\frac{\ln \ln n}{\ln n})$. Let $q$ be a prime number between $\left\lceil d_x^{1+\alpha} \right\rceil$ and $2 \left\lceil d_x^{1+\alpha} \right\rceil$ – such a prime exists [Erd32] – and let $m = d_x q$. Thus, $d_x \left\lceil d_x^{1+\alpha} \right\rceil \leq m \leq 2d_x \left\lceil d_x^{1+\alpha} \right\rceil$ and hence $\frac{n}{2} \leq m \leq 2n$; moreover, it can be easily verified that $d_x \in \left[ \frac{\sqrt{m}}{2}, \frac{2\sqrt{m}}{\ln m} \right]$; both being as required in Section 3. Also notice that this means $q = \Theta(\sqrt{n} \ln n)$. Let $\beta = \frac{1}{\ln n}$ and $r = \left\lceil \frac{\alpha + \beta}{2(1+\alpha)} \right\rceil d_x - 1$, $u = (u_{1,1}, \ldots, u_{1,q}, \ldots, u_{d_x,1}, \ldots, u_{d_x,q})$ and define

$$\text{NW}_r(u) := \sum_{h(z) \in \mathbb{F}_q[z], \deg(h) \leq r} u_{1,h(1)} \cdots u_{d_x,h(d_x)}.$$  

A lower bound on $\dim \langle \pi_x \left( x_M^{\ell} \partial_x^k \text{NW}_r(c_M) \right) \rangle$ was proved in [KS16a,Sha17]. Since their analysis continues to hold for our choice of parameters – which only slightly differ from the parameters in [Sha17] – we omit the proof of the following theorem. Moreover, while they prove this lower bound over fields of characteristic zero, the same proof also works if the characteristic is greater than $q^{(r+1)\cdot \min \left\{ \frac{(r+1)(\gamma)}{\gamma}, \frac{1}{\alpha \gamma - k} \right\}}$.

**Theorem 2** (Lemma 5.2 of [KS16a], Lemma 4.1 of [Sha17]).

$$\dim \langle \pi_x \left( x_M^{\ell} \partial_x^k \text{NW}_r(x_M) \right) \rangle \geq \frac{1}{m^{O(1)}} \min \left\{ \frac{1}{4^k}, \left( \frac{m}{\ell} \right), \left( \frac{m}{\ell + d_x - k} \right) \right\}.$$  

Hence, from Lemma 4.1 and Theorem 2 we get

**Lemma 4.2.** $\text{PSP}_{M,k,\ell}(f_1) \geq \frac{1}{m^{O(1)}} \min \left\{ \frac{1}{4^k}, \left( \frac{m}{\ell} \right), \left( \frac{m}{\ell + d_x - k} \right) \right\}$.  

16
5 Proof of Theorem 1

As before, let \( C \) be a depth four circuit computing the polynomial \( f_n \). Before proving the theorem, let us first justify the assumption that \( \mathbb{F} \) is an algebraically closed field that we made in Section 3. Suppose not. Then, let \( \overline{\mathbb{F}} \) be its algebraic closure. Since \( C \) is also a circuit over \( \overline{\mathbb{F}} \) and \( f_n \) a polynomial over \( \overline{\mathbb{F}} \), we can make all arguments assuming the underlying field to be \( \overline{\mathbb{F}} \). Since the size of a circuit does not depend on the underlying field, the lower bound so obtained will continue to hold when we treat \( C \) as a circuit over \( \mathbb{F} \).

First we will prove a lower bound on the number of wires of \( C \). If the number of monomials in \( C \) is \( \left\lceil \frac{n^2 d_x}{(\ln m)^3} \right\rceil \) then there is nothing to prove. Otherwise from Lemma 3.2, we can obtain a circuit \( C_1 \) such that the support of all the monomials of \( C_1 \) is at most \( \tau = \left\lceil 20 \ln m \right\rceil \), the number of gates in \( C_1 \) is at most the number of gates in \( C \) and the fan-in of each gate in \( C_1 \) is upper bounded by the fan-in of the corresponding gate in \( C \). Then, if \( C_1 \) does not satisfy the hypothesis of Lemma 3.3, the size of \( C_1 \) and hence the size of \( C \) is at least \( \Omega \left( \frac{n^2 d_x}{(\ln m)^3} \right) \). Otherwise, we can obtain a pruned circuit \( D \) such that the top fan-in and the bottom support of \( D \) are upper bounded by the top fan-in and bottom support of \( C_1 \) and so proving a lower bound on the top fan-in of \( D \) would suffice.

As \( D \) computes \( f_1 \) (defined in Section 4), \( \text{PSP}_{M,k,t}(D) = \text{PSP}_{M,k,t}(f_1) \). Lemma 3.1 and 4.2 imply

\[
s \geq \frac{1}{m^{O(1)}} \min \left\{ \frac{1}{n^{m^2}} \cdot \frac{\binom{m}{t} \binom{m}{k}}{\binom{m}{t+k}} \cdot \left( \frac{m}{\ell+2k} \right)^{\left\lceil \frac{\ell}{2} \right\rceil}, \right\}
\]

and the required lower bound follows from the next claim, which is proved in Appendix B.

Claim 5.1. The top fan-in \( s \) of \( D \) is \( \omega \left( \frac{n^2 d_x}{(\ln m)^3} \right) \).

Now let us prove the lower bound on the number of gates. Notice that if the circuit \( C \) computing \( f \) has a heavy gate as defined in Section 3 then we are done. So assume that it does not have any heavy gates. Now, if the number of monomials in \( C \) is \( \left\lceil \frac{n^2 d_x}{(\ln m)^3} \right\rceil \) then there is nothing to prove. Otherwise from Lemma 3.2, we can obtain a circuit \( C_1 \) such that the support of all the monomials of \( C_1 \) is at most \( \tau = \left\lceil 20 \ln m \right\rceil \), the number of gates in \( C_1 \) is at most the number of gates in \( C \) and the fan-in of each gate in \( C_1 \) is upper bounded by the fan-in of the corresponding gate in \( C \). Obtain a circuit \( D \) from \( C_1 \) by picking a set \( M_2 \subseteq [3m] \setminus M_1 \) (where \( M_1 \) is as in Lemma 3.2), \( |M_2| = m \) arbitrarily and setting variables in \( x_{M_2} \) and \( y_{M_2} \) to 0 (notice that the top fan-in and bottom support of \( D \) are upper bounded by the top fan-in and bottom support of \( C_1 \)). Now \( D \) computes \( f_1 \) (where \( f_1 \) is as defined in Section 4) and just as it was done in the preceding paragraph, we can show that the top fan-in of \( D \) is \( \omega \left( \frac{n^2 d_x}{(\ln m)^3} \right) \). However, we only get an \( \Omega \left( \frac{md_x}{(\ln m)^3} \right) \) lower bound on the number of gates since the definition of a heavy gate is the bottleneck.

6 Conclusion

We conclude by stating a few questions/problems, some of which may not be very hard to answer/solve.
1. **Improving the depth four lower bound.** An affirmative answer to any of the following questions will strengthen or improve the lower bound shown in this work.

- Can we prove an $\tilde{\Omega}(n^{2.5})$ lower bound on the number of gates of depth four circuits? The almost cubic lower bound in [KST16] is on the number of gates of depth three circuits.
- Can we prove an $\tilde{\Omega}(n^{2.5})$ lower bound for depth four circuits computing $\text{IMM}_{2,n}$? The same question may also be asked with regard to the $\Omega(n^3)$ lower bound for depth three circuits.
- Can we prove an $\Omega(n^3)$ lower bound for depth four circuits? The loss of a $\sqrt{n}$ factor (in Theorem 1) in comparison to the almost cubic lower bound for depth three circuits [KST16] is due to the use of the projected shifted partials measure in place of the shifted partials measure.

2. **A lower bound for depth five circuits.** As mentioned in Section 1.3, an $\Omega(n^{1.8+\epsilon})$ lower bound on the number of gates of a depth five circuit computing $\text{IMM}_{2,n}$ implies a super-cubic lower bound for depth three circuits. It is also interesting to note that the hard polynomial used in [Yau16,BLS16] to prove an $\tilde{\Omega}(n^3)$ lower bound for depth three circuits is computable by a poly$(n)$-size depth five circuit. As a natural next step, we pose the following problem:

- Prove a super-quadratic lower bound for depth five circuits.

3. **Super-linear lower bound for constant depth circuits computing $\text{IMM}_{2,n}$.** Recall that lower bounds of $\Omega(\Delta n^{1+\frac{\Delta}{2}})$ and roughly $\Omega(n^{1+\frac{1}{3\Delta}})$ are shown for depth-$\Delta$ circuits in [SS97] and [Raz10] respectively. We have argued in Section 1.3 that the same lower bound for depth-$\Delta$ circuits computing $\text{IMM}_{2,n}$ would give a super-polynomial lower bound for constant depth circuits. It is thus natural to ask:

- Can we prove a super-linear lower bound for constant depth circuits computing $\text{IMM}_{2,n}$?

4. **Hardness magnification for commutative circuits.** It was shown in [CILM18] that a sufficiently large super-linear lower bound for non-commutative circuits implies an arbitrarily large polynomial lower bound for general non-commutative circuits. It would be highly interesting to show a similar hardness magnification result for commutative circuits.

**Acknowledgments**

We would like to thank Neeraj Kayal and Ankit Garg for sitting through a presentation of this work and giving us useful feedback. Thanks specially to Ankit for bringing the work of Chen and Tell [CT19] to our notice. A part of this work is done at Microsoft Research India (MSRI), where CS is spending a sabbatical year. CS would like to thank MSRI for providing an excellent research environment and for the hospitality.
References


A Missing proofs from Section 3

A.1 Proofs from Section 3.1

Claim 3.1. Let \( P = Q_1^{e_1} \cdots Q_t^{e_t} \) be one of the polynomials \( P_i \). For \( k \geq 0 \), let \( P^{(k)} := \prod_{i \in [t]} Q_i^{\max(e_i-k,0)} \). Then, \( \partial^k P \subseteq \mathbb{F} \cdot \text{span}\{y_M \leq \infty, x_M \leq k(2t\tau - 1) P^{(k)}\} \).

Proof. We prove the claim by induction on \( k \). If \( k = 0 \), then \( \partial^{0}_{x_M} P = \{P\} = \{P^{(0)}\} \) and hence the claim is true. Assume that the claim is true for \( k \). Let \( X \) be a multilinear monomial of degree \( k + 1 \) in \( x \) variables. Then \( X = xX' \) where \( X' \) is a multilinear monomial of degree \( k \) in \( x \) variables and \( x \) one of the \( x \) variables. From the induction hypothesis we have that,

\[
\frac{\partial P}{\partial X'} = g \cdot P^{(k)}
\]

where \( g \) is a polynomial in \( \mathbb{F}[x_M, y_M] \) with \( x_M \) degree of \( g \) being at most \( k(2t\tau - 1) \) while its \( y_M \) degree can be arbitrarily large.

Let \( J := \{j \in [t] : e_j > k\} \). We have that,

\[
\frac{\partial P}{\partial X} = \frac{\partial}{\partial x} \left( g \cdot P^{(k)} \right) = \frac{\partial}{\partial x} \left( g \cdot \prod_{j \in J} Q_j^{e_j-k} \right)
\]
For large enough $m$ we will show that the ratio

$$
\frac{m - 2kt\tau}{\ell} > 1.
$$

Putting the values of $k$ and $\ell$,

$$
\frac{m}{2} - 2kt\tau = \frac{m}{2} - 2\left\lfloor \frac{d_x}{m^{\delta/t}} \right\rfloor t\tau
\geq \left(1 - \frac{2d_x\tau}{m}\right)(m^{\delta/t} + 1).
$$

So, we need to show that

$$
\frac{1}{2} - \frac{2d_x\tau}{m} < m^{\delta/t} + 1 \iff \frac{1}{2} - \frac{2d_x\tau}{m} - 1 < m^{\delta/t}
\iff 1 + \frac{4d_x\tau}{m} < m^{\delta/t}.
$$

For large enough $m$, $\frac{4d_x\tau}{m} \leq \frac{1}{2}$. Using $1 + x \leq e^x$, which holds for all $x \in \mathbb{R}$, and $\frac{1}{1-x} \leq e^{2x}$, which holds for $0 \leq x \leq \frac{1}{2}$ we get:

$$
1 + \frac{4d_x\tau}{m} \leq e^{\frac{12d_x\tau}{m}}.
$$

So showing that $e^{\frac{12d_x\tau}{m}} < m^{\delta/t}$ would suffice. Now,

$$
e^{\frac{12d_x\tau}{m}} < m^{\delta/t} \iff e^{\frac{12d_x\tau}{m}} < m.
$$

Putting the values of $d_x$, $t$ and $\tau$, we get that

$$
\frac{12d_x\tau}{m} = \frac{12d_x}{m} \left(\frac{1}{\ln m}\right) \leq \frac{12d_x^2}{m(\ln m)^2} = \Theta\left(\frac{1}{m^{\frac{1}{2}(\ln m)^2}}\right) = o(1) \text{ as } d_x = \Theta\left(\frac{\sqrt{m}}{\ln m}\right). \text{ Thus } e^{\frac{12d_x\tau}{m}} < m.
$$

Observe that as $D$ is a pruned depth four circuit, the support of all monomials of $Q'_j$ is upper bounded by $\tau$ and as in any monomial the individual degree of any $x$ variable is at most two, $\deg_x(Q'_j) \leq 2\tau$. Also, $|J| \leq t$ and hence

$$
\deg_x\left(\prod_{j \in J} Q'_j + g \prod_{j \in J} (e_j - k) Q'_j \prod_{i \in J \setminus \{j\}} Q'_i \prod_{j \in J} Q'^{(e_j - k)}_j\right) \leq (k + 1)(2\tau - 1).
$$

As $\prod_{j \in J} Q'^{(e_j - k)}_j = P^{(k+1)}$, the claim is true for $k + 1$.

Claim 3.2. Let $\ell, k, t$ and $\tau$ be as defined earlier. Then, $\ell + 2kt\tau < \frac{m}{2}$.

Proof. We will show that the ratio $\frac{m - 2kt\tau}{\ell} > 1$. Putting the values of $k$ and $\ell$,

$$
\frac{m}{2} - 2\left\lfloor \frac{d_x}{m^{\delta/t}} \right\rfloor t\tau
\geq \left(1 - \frac{2d_x\tau}{m}\right)(m^{\delta/t} + 1).
$$

So, we need to show that

$$
\frac{1}{2} - \frac{2d_x\tau}{m} < m^{\delta/t} + 1 \iff \frac{1}{2} - \frac{2d_x\tau}{m} - 1 < m^{\delta/t}
\iff 1 + \frac{4d_x\tau}{m} < m^{\delta/t}.
$$

For large enough $m$, $\frac{4d_x\tau}{m} \leq \frac{1}{2}$. Using $1 + x \leq e^x$, which holds for all $x \in \mathbb{R}$, and $\frac{1}{1-x} \leq e^{2x}$, which holds for $0 \leq x \leq \frac{1}{2}$ we get:

$$
1 + \frac{4d_x\tau}{m} \leq e^{\frac{12d_x\tau}{m}}.
$$

So showing that $e^{\frac{12d_x\tau}{m}} < m^{\delta/t}$ would suffice. Now,

$$
e^{\frac{12d_x\tau}{m}} < m^{\delta/t} \iff e^{\frac{12d_x\tau}{m}} < m.
$$

Putting the values of $d_x$, $t$ and $\tau$, we get that

$$
\frac{12d_x\tau}{m} = \frac{12d_x}{m} \left(\frac{1}{\ln m}\right) \leq \frac{12d_x^2}{m(\ln m)^2} = \Theta\left(\frac{1}{m^{\frac{1}{2}(\ln m)^2}}\right) = o(1) \text{ as } d_x = \Theta\left(\frac{\sqrt{m}}{\ln m}\right). \text{ Thus } e^{\frac{12d_x\tau}{m}} < m.
$$
A.2 Proof from Section 3.2.1

Claim 3.3. Procedure 1 terminates in at most m iterations.

Proof. Let $H_i$ be the set $H$ after the $i$-th iteration of the procedure. Since each monomial in $H_i$ has support more than $\tau$, for any such monomial there are at least $\frac{\tau}{2}$ distinct $j \in [3m] \setminus M_1$ such that at least one of $x_j$ and $y_j$ appears in it. Counting the number of times at least one of $x_j$ and $y_j$ appears in a monomial in $H_i$ and summing up these counts for all $j \in [3m] \setminus M_1$, we get that

$$\sum_{j \in [3m] \setminus M_1} e(j) \geq \frac{\tau \cdot |H_i|}{2};$$

so from an averaging argument there exists a $j$ such that

$$e(j) \geq \frac{\tau \cdot |H_i|}{6m}.$$  

Hence, the size of $H_{i+1}$ is upper bounded as

$$|H_{i+1}| \leq |H_i| \cdot \left(1 - \frac{\tau}{6m}\right).$$

So after $i$ iterations of the procedure we get,

$$|H_i| \leq |H_0| \cdot \left(1 - \frac{\tau}{6m}\right)^i \leq \left|\frac{m^2 d_x}{(\ln m)^5}\right| \cdot \left(1 - \frac{|20 \ln m|}{6m}\right)^i \leq \frac{m^2 d_x}{(\ln m)^5} \cdot \left(1 - \frac{(20 \ln m - 1)}{6m}\right)^i \leq \frac{m^2 d_x}{(\ln m)^5} \cdot e^{-\frac{3 \ln m}{m}} \leq \frac{m^2 d_x}{(\ln m)^5} \cdot m^{-\frac{3}{m}}.$$ 

For $i = m, |H_i| < 1$ (for sufficiently large $m$), i.e., the procedure terminates in at most $m$ iterations. \hfill \square

A.3 Proof from Section 3.2.2

Claim 3.4. Let $\overline{M}_1 = [3m] \setminus M_1$. Procedure 2 sets at most $m$ many variables in $x_{\overline{M}_1} \cup y_{\overline{M}_1}$ to field constants and removes all the heavy gates from $C_1$.

Proof. In each iteration, we evaluate a light sparse polynomial in $C_1$ to zero. This can be done as $\mathbb{F}$ is an algebraically closed field. Since the support of every monomial in $C_1$ is at most $\tau$, we end up setting at most $\frac{\tau m}{(\ln m)^2} \leq \frac{20 m}{\ln m}$ many variables to field constants in each iteration. As we can afford to set $m$ variables, Step 2 of the procedure executes successfully. For some $i \in \mathbb{N}$, the while loop terminates in the $i$-th iteration in either of the following two cases:

\[ \]
1. All the heavy gates get eliminated after the \((i - 1)\)-th iteration, i.e., \(s_i = 0\).

2. \(\tau(b_1 + \cdots + b_i) > m\). (We show in the following subclaim that all the heavy gates are eliminated before this happens. Hence, the procedure stops only in the above case.)

**Subclaim A.1.** Let \(i \in \mathbb{N}\) be such that \(\tau(b_1 + \cdots + b_{i-1}) \leq m\) but \(\tau(b_1 + \cdots + b_i) > m\). Then, all the heavy gates in \(C_1\) get eliminated in the first \((i - 1)\) iterations of Procedure 2.

If we assume Subclaim A.1 then Claim 3.4 is proved. \(\square\)

**Proof of Subclaim A.1:**

For \(1 \leq j < i\), let \((Q_{j,1}, \ldots, Q_{j,r_j})\) be the available light sparse polynomials in \(C_1\) after the \((j - 1)\)-th iteration. Recall that \(s_j\) is the number of heavy gates in \(C_1\) after the \((j - 1)\)-th iteration. Suppose, \(s_j \geq 1\) (otherwise, we have nothing to prove). For every \(l \in [r_j]\), \(b_{j,l}\) and \(c_{j,l}\) refer to the fan-in of \(Q_{j,l}\) and the number of distinct heavy gates connected to \(Q_{j,l}\) in \(C_1\) respectively. We may assume that \(b_{j,l} \neq 0\) for every \(l \in [r_j]\). It is given that

\[
 b_{j,1} + \cdots + b_{j,r_j} \leq \frac{m^2d_x}{160 \cdot \lambda_0 \cdot (\ln m)^3}. \quad (2)
\]

Since every heavy gate is connected to at least \(\frac{md_x}{2 \lambda_0 \cdot (\ln m)^3}\) many light sparse polynomials in \(C_1\),

\[
 s_j \cdot \frac{md_x}{2 \lambda_0 \cdot (\ln m)^3} \leq c_{j,1} + \cdots + c_{j,r_j}.
\]

As \(b_{j,1}, \ldots, b_{j,r_j}\) are all non-zero, we get

\[
 s_j \cdot \frac{md_x}{2 \lambda_0 \cdot (\ln m)^3} \leq \frac{c_{j,1}}{b_{j,1}} \cdot b_{j,1} + \cdots + \frac{c_{j,r_j}}{b_{j,r_j}} \cdot b_{j,r_j}.
\]

Let \(u \in [r_j]\) be such that \(\frac{c_{j,u}}{b_{j,u}} = \max \left\{ \frac{c_{j,1}}{b_{j,1}}, \ldots, \frac{c_{j,r_j}}{b_{j,r_j}} \right\}\). Let \(c_j := c_{j,u}\) and \(b_j := b_{j,u}\). Then, the above equation implies

\[
 s_j \cdot \frac{md_x}{2 \lambda_0 \cdot (\ln m)^3} \leq \frac{c_j}{b_j} \cdot (b_{j,1} + \cdots + b_{j,r_j}).
\]

From Equation (2), we get that for every \(1 \leq j < i\),

\[
 s_j \cdot \frac{md_x}{2 \lambda_0 \cdot (\ln m)^3} \leq \frac{c_j}{b_j} \cdot \frac{m^2d_x}{160 \cdot \lambda_0 \cdot (\ln m)^3}.
\]

which implies

\[
 \frac{80 \cdot s_j \cdot b_j \cdot (\ln m)^2}{m} \leq c_j. \quad (3)
\]

Thus, by setting the light sparse polynomial \(Q_{j,u}\) to zero in the \(j\)-th iteration, we get rid of at least \(\frac{80 \cdot s_j \cdot b_j \cdot (\ln m)^2}{m}\) many heavy gates from \(C_1\). Recall that \(s_{j+1}\) is the number of available heavy gates after the \(j\)-th iteration. Then, for every \(1 \leq j < i\),

\[
 s_{j+1} \leq s_j - c_j \leq s_j \cdot \left(1 - \frac{80 \cdot b_j \cdot (\ln m)^2}{m}\right). \quad (4)
\]

30
Hence, for every $1 \leq j < i$,

$$s_{j+1} \leq s_1 \cdot \prod_{l=1}^{j} \left( 1 - \frac{80 \cdot b_l \cdot (\ln m)^2}{m} \right).$$

(5)

Also, for every $l \leq j$, we have $c_l \leq s_j$. Thus, Equation (3) implies

$$\frac{80 \cdot b_l \cdot (\ln m)^2}{m} \leq 1.$$

(6)

As $1 - a \leq e^{-\frac{a}{2}}$ for $0 \leq a \leq 1$, Equations (5) and (6) imply

$$s_{j+1} \leq s_1 \cdot \prod_{l=1}^{j} \left( e^{-\frac{40 \cdot b_l \cdot (\ln m)^2}{m}} \right) = s_1 \cdot e^{-\frac{40 \cdot (b_1 + \cdots + b_j) \cdot (\ln m)^2}{m}}.
$$

(7)

It is given that $\tau(b_1 + \cdots + b_i) > m$, which implies $\tau(b_1 + \cdots + b_{i-1}) > m - \tau \cdot b_i$. As $b_i$ is the fan-in of a light sparse polynomial, $b_i \leq \frac{m}{(\ln m)^2}$ and so

$$\tau(b_1 + \cdots + b_{i-1}) > m - \frac{m \tau}{(\ln m)^2}.
$$

(8)

On substituting $j = i - 1$, $\tau = \lfloor 20 \ln m \rfloor$ and the value of $\tau(b_1 + \cdots + b_{i-1})$ from Equation (8) in Equation (7), we get

$$s_i \leq s_1 \cdot e^{-\frac{40 \cdot (\ln m)^2}{m} \left( 2m \ln m - \frac{m}{(\ln m)^2} \right)} = s_1 \cdot e^{-20 \ln m - 40}.
$$

For large enough $m$, $s_i \leq \frac{s_1}{m^{1/4}}$. Since $s_1 \leq m$, we get $s_i = 0$ (as it is a natural number). In other words, we get rid of all the heavy gates within $(i-1)$ iterations.

### B Missing proofs from Section 5

**Claim 5.1.** The top fan-in $s$ of $D$ is $\omega \left( \frac{m^d}{(\ln m)^3} \right)$.

**Proof.** From Equation (1), we have

$$s \geq \frac{1}{m^{O(1)}} \min \left\{ \frac{\binom{m}{\ell} \cdot \binom{m}{\ell+2k+1}}{m^{O(1)} \cdot \binom{m}{\ell+2k+1}} \left( \frac{m}{(\ell+2k+1)} \right)^{\ell+2k+1} \right\}$$

$$= \frac{1}{m^{O(1)}} \min \left\{ \frac{\binom{m}{\ell} \cdot \binom{m}{\ell+1} \cdot \binom{m}{\ell+2k-1}}{m^{O(1)} \cdot \binom{m}{\ell+2k} \cdot \binom{m}{\ell+2k+1}} \right\}$$

$$= \frac{1}{m^{O(1)}} \min \left\{ \frac{\binom{m}{\ell}}{4^k \binom{m}{\ell+2k}} \cdot \frac{(m-\ell-2kt+1)!}{(m-\ell-1)!} \cdot \frac{(\ell+2kt+1)!}{(\ell+1)!} \cdot \frac{(m-\ell-2kt)!}{(m-\ell-d_k+k)!} \cdot \frac{(\ell+2kt)!}{(\ell+d_k-k)!} \right\}.$$
\[
\frac{1}{m^{O(1)}} \min \left\{ \frac{\binom{m}{k}}{4^k}, \left( \frac{m}{\ell + 1} - 1 \right)^{-2k\tau}, \left( \frac{m}{\ell} - 1 \right)^{(d_k - 2k\tau - k)} \right\}
\]  
(Using Proposition 2.)

\[
\geq \frac{1}{m^{O(1)}} \min \left\{ \frac{\binom{m}{k}}{4^k}, \left( \frac{m}{\ell + 1} - 1 \right)^{-2k\tau}, \left( \frac{m}{\ell} - 1 \right)^{(d_k - 2k\tau - k)} \right\}
\]

Since \(\frac{\binom{m}{k}}{4^k} \cdot m^{-2k\delta\tau} = \left( \frac{e^k}{k} \right)^k \cdot m^{(1-2\delta\tau - \frac{\tau}{4})k} \leq (\frac{e^k}{k})^k \cdot m^{(1-2\delta\tau - \frac{\tau}{4})k}\) for our choice of parameters \(\delta, k\) and \(t\), \(m^{\frac{4k}{\ell}} = O(1)\). Hence, \(\frac{\binom{m}{k}}{4^k} \cdot m^{-2k\delta\tau} \leq m^{(1-2\delta\tau - \frac{\tau}{4})k}\) and thus,

\[
s \geq \frac{1}{m^{O(1)}} \cdot \frac{\binom{m}{k} \cdot m^{-2k\delta\tau}}{4^k \cdot \left( \frac{\tau}{\ell} + 1 \right)^k}
\]

(Using Proposition 1.)

\[
\geq \frac{1}{m^{O(1)}} \cdot \left( \frac{m \cdot k}{4e \cdot k \cdot m^{2\delta\tau} \cdot (\frac{w}{\ell} + k)} \right)^k
\]

(Since \(kt \leq w = \left\lfloor \frac{md_k}{\lambda_0 (\ln m)^7} \right\rfloor\).)

\[
\geq \frac{1}{m^{O(1)}} \cdot \left( \frac{m \cdot t}{8e \cdot m^{2\delta\tau} \cdot \left\lfloor \frac{w}{\ell} \right\rfloor} \right)^k
\]

(Since \(k \geq \lfloor \ln m \rfloor\).)

\[
\geq \frac{1}{m^{O(1)}} \cdot \left( \frac{md_k}{(\ln m)^7} \right)^{\lambda_0 / (\ln m)^7} \ln m
\]

\[
\geq \frac{1}{m^{O(1)}} \cdot \left( \frac{md_k}{(\ln m)^7} \right)^{\lambda_0 / (\ln m)^7} \ln m
\]

\[
= \frac{1}{m^{O(1)}} \cdot \left( \frac{\lambda_0}{16e \cdot e^{O(1)}} \right)^{\ln m}
\]

if we choose \(\lambda_0\) to be a large enough constant. \(\square\)
C  A brief review of the lower bounds from [SS97] and [Raz10]

In this section, we present a short overview of the lower bounds for restricted depth arithmetic circuits with multiple output gates from [SS97] and [Raz10] and focus mainly on depth four circuits. We would use \((s, \Delta)\)-arithmetic circuit to denote an arithmetic circuit of size-\(s\) and depth-\(\Delta\) and \(y\) for the set of variables \(\{y_1, \ldots, y_n\}\).

Lower bound from [SS97]. Let \(n \in \mathbb{N}\) and \(\Delta = O(\log n)\). Shoup and Smolensky showed that there exist \(n\) linear forms \(g_1, \ldots, g_n \in \mathbb{C}[y]\), such that the size of any depth-\(\Delta\) normal-linear circuit\(^{28}\) that computes \(g_1, \ldots, g_n\) is \(\Omega(\Delta n^{1+\frac{1}{\Delta}})\). The following proposition implies that the same lower bound holds for a depth-\(\Delta\) arithmetic circuit, that also computes \(g_1, \ldots, g_n\).

**Proposition 3.** Let \(n \in \mathbb{N}\), \(F\) be an arbitrary field and \(h_1, \ldots, h_n \in F[y]\) be linear forms computed by an \((s, \Delta)\)-arithmetic circuit. Then, there exists an \((s, \Delta)\)-normal-linear circuit that computes \(h_1, \ldots, h_n\).

This proposition is easy to prove; a proof of the same in given in Section 2 of [Raz10]. We refer the reader to Section 3 of [SS97] for more details. In case of depth four arithmetic circuits over \(\mathbb{C}\), if we substitute \(\Delta = 4\) in the above mentioned result then we get a lower bound of \(\Omega(n^{1.25})\), but we observe that this lower bound can be optimised roughly to \(\Omega(n^{1.33})\) using the following claim.

**Claim C.1.** Let \(n \in \mathbb{N}\) and \(h_1, \ldots, h_n \in F[y]\) be linear forms, computed by an \((s, 4)\)-arithmetic circuit \(C\) over \(F\). Then, there exists an \((s, 3)\)-normal-linear circuit over \(F\) that computes \(h_1, \ldots, h_n\).

**Proof.** From Proposition 3, we obtain an \((s, 4)\)-normal-linear circuit \(D\) over \(F\) that computes \(h_1, \ldots, h_n\). We now argue that linearisation ensures that the fan-in of every gate in the bottom layer of \(D\) is exactly 1. It turns out that only those product gates survive the linearisation in the bottom layer of \(C\) which are connected to exactly one variable. Let \(v\) be a gate in the bottom layer of \(C\) with children \(u_1, \ldots, u_r\). Then, there exists some \(i \in \mathbb{N}\) such that \(u_i\) is a variable and all other gates are labelled by field constants, in which case, we remove \(u_j, j \in \mathbb{N} \setminus \{i\}\) and multiply the label of the edge \((v, u_i)\) with \(\prod_{j\in\mathbb{N}\setminus\{i\}} u_j\). As every gate in the bottom layer of \(D\) has fan-in 1, we can directly connect the input of every gate in this layer to its outputs, thereby yielding an \((s, 3)\)-normal-linear circuit that computes \(h_1, \ldots, h_n\).

Lower bound from [Raz10]. Let \(n\) be a prime number and \(\Delta = O(\log n)\). Raz showed that there exist \(n\) explicit homogeneous polynomials of degree \(\Theta(\Delta)\) in \(\Theta(n)\) variables over \(F\), such that any depth-\(\Delta\) arithmetic circuit that computes these polynomials has size \(\Omega(n^{1+\frac{1}{\Delta}})\). While the lower bound for depth-\(\Delta\) arithmetic circuit given in [SS97] holds for \(n\) non-explicit linear forms over \(\mathbb{C}\), the same lower bound also holds for \(n\) explicit homogeneous polynomials of \(\Theta(n)\) degree in \(\Theta(n)\) variables over \(\mathbb{C}\). We first recall some definitions from [Raz10] and then show that the lower bound for depth-\(\Delta\) arithmetic circuits in [Raz10] can be optimized slightly.

---

\(^{28}\)An arithmetic circuit \(D\) over \(F\) is called a normal-linear circuit if all the gates in \(D\) are labelled by either variables or by \(+\). Every gate in \(D\) computes a linear form in the underlying set of variables over \(F\).
Let $n', m, t, s \in \mathbb{N}$. A polynomial mapping $f : \mathbb{F}^{n'} \to \mathbb{F}^m$ of degree $t$ is an $m$ tuple $(f_1, \ldots, f_m)$ of $n'$ variate degree $t$ polynomials over $\mathbb{F}$. The polynomial mapping $f$ eludes a polynomial mapping $\Gamma : \mathbb{F}^s \to \mathbb{F}^m$ if $\text{Image}(f) \not\subseteq \text{Image}(\Gamma)$. Moreover, $f$ is said to be $(s, t)$-elusive over $\mathbb{F}$ if it eludes every polynomial mapping $\Gamma : \mathbb{F}^s \to \mathbb{F}^m$ of degree at most $t$.

Let $n$ be a prime, $\Delta = O(\log n)$, $m := n^2$, $\Delta' := a \cdot \Delta$, where $a \in \mathbb{N}$ is a constant, $n' := \Delta' \cdot n$ and $x := \{x_{k,l} : k \in [\Delta'], l \in [n]\}$. Let $f : \mathbb{F}^{n'} \to \mathbb{F}^m$ be defined as follows:

For every $(i, j) \in [n] \times [n]$,
\[
 f_{(i,j)}(x) := \prod_{k=1}^{\Delta'} x_{k,(i+j)k} \mod n. \tag{9}
\]

Further, for every $i \in [n]$,
\[
 f_i(x, y) := \sum_{j \in [n]} y_j \cdot f_{(i,j)}(x). \tag{10}
\]

[Raz10] showed that for $a = 5$, any depth-$\Delta$ arithmetic circuit computing $\tilde{f}_1, \ldots, \tilde{f}_n$ requires size $\Omega(n^{1 + \frac{3}{\Delta^2}})$. The detailed proof is given in Section 4 of [Raz10]. Here, we show how this lower bound can be optimized to $\Omega(n^{1 + \frac{1}{2} - \epsilon_{a,\Delta}})$, where $\epsilon_{a,\Delta} := \frac{2\Delta - 1}{a \Delta^2}$. Note that as we increase the value of $a$, this lower bound gets closer to the one for depth-$\Delta$ arithmetic circuits given in [SS97]. The main ingredient of this improvement is the following optimization of Lemma 4.1 in [Raz10].

**Lemma C.1.** Let $n$ be a prime, $m = n^2$, $\Delta = O(\log n)$, $\Delta' := a \cdot \Delta$, where $a \in \mathbb{N}$ is a constant and $n' := \Delta' \cdot n$. Let $G$ be a field extension of $\mathbb{F}$ of size more than $m$ and $f : G^{n'} \to G^m$ be the polynomial mapping defined in Equation (9)\textsuperscript{29}. Then, $f$ is $(s, \Delta)$-elusive over $G$, where $s = \lfloor n^{1 + \frac{1}{2} - \epsilon_{a,\Delta}} \rfloor$ and $\epsilon_{a,\Delta} = \frac{2\Delta - 1}{a \Delta^2}$.

**Proof.** Let $U := [n] \times [n]$, $r := \frac{1}{2} \lfloor n^{1 - \frac{1}{2\Delta}} \rfloor$. For $A \subseteq U$, $f_A(x) := \prod_{(i,j) \in A} f_{ij}(x)$. $A$ is said to be retrievable if for any $A' \subseteq U$, $f_A \neq f_{A'}$ implies $A \neq A'$. It is shown in Claim 4.2 of [Raz10] that

\[
 \Pr_{A \in R(U)} [A \text{ is not retrievable}] \leq \left( \frac{|A|}{n + 1} \right)^{\Delta'} \cdot n^2,
\]

where $A \in R(U)$ means that $A$ is a subset of $U$ of size $r$ chosen uniformly at random. On plugging the value of $r$ in the above equation, we get

\[
 \Pr_{A \in R(U)} [A \text{ is retrievable}] > \frac{1}{2}. \tag{11}
\]

Let $\mathcal{L}$ be the set of degree $r$ multilinear homogeneous polynomials of the type $g : G^m \to G$, such that every monomial of $g$ corresponds to a retrievable set. Clearly, $\mathcal{L}$ is a $G$-vector space. From Equation (11), we get $\dim_G(\mathcal{L}) > \frac{1}{2} \binom{m}{r} \geq \frac{1}{2} \left( \frac{m}{r} \right)^r = \frac{1}{2} \left( 2n^{1 + \frac{1}{2\Delta}} \right)^r$. Fix a polynomial map $\Gamma : G^s \to G^m$ of degree $\Delta$. Then, for every $g \in \mathcal{L}$, $g \circ \Gamma : G^s \to G$ is a polynomial of degree $r \cdot \Delta$. Let $\mathcal{K}$ be the set of all polynomials from $G^s$ to $G$ of degree at most $r \cdot \Delta$. Then, $\mathcal{K}$ is a $G$-vector space and $\dim_G \mathcal{K} \leq \left( \frac{s + r \cdot s \cdot r \cdot \Delta}{r \cdot \Delta} \right)^{r \cdot \Delta} \leq \left( \frac{2\Delta}{r \cdot \Delta} \right)^{r \cdot \Delta} = \left( 12n^{1 + \frac{2}{\Delta} - \epsilon_{a,\Delta}} \right)^{r \cdot \Delta}$.

\textsuperscript{29} $f$ is naturally a polynomial mapping over $G$ because from every $i, j \in [n], f_{ij}(x) \in \mathbb{F}[x] \subseteq G[x].$
On substituting the values of $\Delta'$ and $\epsilon_{a,\Delta}$ in $\dim_G L$ and $\dim_G K$, we get $\dim_G K < \dim_G L$. Now, for a fixed polynomial map $\Gamma : G^s \to G^m$ of degree $\Delta$, define $\phi_{\Gamma} : L \to K : g \mapsto g \circ \Gamma$. Clearly, $\phi_{\Gamma}$ is a $G$-linear map and as $\dim_G K < \dim_G L$, $\phi_{\Gamma}$ is not an injective map. This means that there exists a non-zero $g_{\Gamma} \in L$, such that $\phi_{\Gamma}(g_{\Gamma}) = g_{\Gamma} \circ \Gamma = 0$. As $|G| > m$, Claim 4.4 in [Raz10] implies that $g_{\Gamma} \circ f : G' \to G$ is not the zero polynomial. Thus, for every polynomial mapping $\Gamma : G^s \to G^m$ of degree $\Delta$, $\text{Image}(f) \not\subset \text{Image}(\Gamma)$. Hence, $f$ is an $(s, \Delta)$-elusive polynomial map over $G$. □

The following is a corollary of Lemma C.1 and Proposition 3.11 in [Raz10].

**Corollary C.1.** Let $n$ be a prime, $\Delta = O(\log n)$ and $\Delta' = a \cdot \Delta$ for some constant $a \in \mathbb{N}$. Let $\tilde{f}_1, \ldots, \tilde{f}_n$ be $n(\Delta' + 1)$ variate degree $\Delta' + 1$ polynomials as defined in Equation (10). Then, any depth-$\Delta$ arithmetic circuit $C$ over $\mathbb{F}$ computing $\tilde{f}_1, \ldots, \tilde{f}_n$ requires size $\Omega \left( n^{1 + \frac{1}{\Delta} - \epsilon_{a,\Delta}} \right)$, where $\epsilon_{a,\Delta} = 2 \cdot \frac{\Delta - 1}{a \cdot \Delta^2}$.

**Proof idea.** In Proposition 3.11 in [Raz10], $\tilde{f}_1, \ldots, \tilde{f}_n$ and $C$ are viewed as linear polynomials in $y$ variables over the function field $\mathbb{F}(x)$ and an arithmetic circuit over $\mathbb{F}(x)$ respectively. Then, using Proposition 3, $C$ is converted to an $(s, \Delta)$-normal-linear circuit over $\mathbb{F}(x)$, that also computes $\tilde{f}_1, \ldots, \tilde{f}_n$. After that, on invoking Lemma C.1, we get the lower bound of $\Omega \left( n^{1 + \frac{1}{\Delta} - \epsilon_{a,\Delta}} \right)$ on a depth-$\Delta$ arithmetic circuit computing $\tilde{f}_1, \ldots, \tilde{f}_n$.

We now focus on depth four circuits. Let $s := n^{\frac{3}{4} - \epsilon_{a,\Delta}}$, where $\epsilon_{a,3} := \frac{5}{9a}$. In the proof of Corollary C.1, we use Claim C.1 to obtain an $(s, 3)$-normal-linear circuit over $\mathbb{F}(x)$ from an $(s, 4)$-arithmetic circuit over $\mathbb{F}(x)$, such that both circuits compute $\tilde{f}_1, \ldots, \tilde{f}_n$. As Lemma C.1 implies that the polynomial mapping $f$ is $(s, 3)$-elusive, we get a lower bound of $\Omega(n^{\frac{3}{4} - \epsilon_{a,3}})$ for depth four circuits.