# Efficient Reconstruction of Random Multilinear Formulas

Ankit Gupta
*Microsoft Research India*
t-ankitg@microsoft.com

Neeraj Kayal
*Microsoft Research India*
neeraka@microsoft.com

Satya Lokam
*Microsoft Research India*
satya@microsoft.com

**Abstract**— In the reconstruction problem for a multivariate polynomial $f$, we have blackbox access to $f$ and the goal is to efficiently reconstruct a representation of $f$ in a suitable model of computation. We give a polynomial time randomized algorithm for reconstructing *random* multilinear formulas. Our algorithm succeeds with high probability when given blackbox access to the polynomial computed by a random multilinear formula according to a natural distribution. This is the strongest model of computation for which a reconstruction algorithm is presently known, albeit efficient in a distributional sense rather than in the worst-case. Previous results on this problem considered much weaker models such as depth-3 circuits with various restrictions or read-once formulas.

Our proof uses ranks of partial derivative matrices as a key ingredient and combines it with analysis of the algebraic structure of random multilinear formulas. Partial derivative matrices have earlier been used to prove lower bounds in a number of models of arithmetic complexity, including multilinear formulas and constant depth circuits. As such, our results give supporting evidence to the general thesis that mathematical properties that capture efficient computation in a model should also enable learning algorithms for functions efficiently computable in that model.

***Keywords***-arithmetic circuits; multilinear formulas; reconstruction; learning.

## 1. Introduction

We study the problem of reconstructing a multivariate polynomial: given blackbox access to a hidden polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ over a finite[1] field $\mathbb{F}$, reconstruct a representation of $f$ in some suitable model of computation. A reconstruction algorithm can adaptively query the blackbox to evaluate $f$ on inputs of its choice from $\mathbb{F}^n$. Its efficiency is measured in terms of the number of queries and the running time. We typically assume $f$ itself to be efficiently computable in some model of computation, e.g., depth-3 circuits of polynomial size, and also require the reconstruction algorithm to produce a succinct representation of $f$ in some (possibly different) model of computation. The most obvious representation of a multivariate polynomial is its formula as a sum, weighted by coefficients from $\mathbb{F}$, of monomials, i.e., a depth-2 $\Sigma\Pi$ formula. In this case, the problem of reconstruction is more commonly referred to as *interpolation*: given blackbox access to a polynomial, produce its representation as a sum of products. However, many interesting polynomials, e.g., determinant, have exponentially long (in the number of variables) representations as a sum of products, whereas as a straight line program or an arithmetic circuit, they can be represented much more succinctly. The reconstruction problem demands such succinct representations as outputs and hence is a generalization of the interpolation problem. In its most general formulation, e.g., produce (roughly) the smallest arithmetic circuit for $f$, the reconstruction problem is extremely hard. If a circuit class $\mathcal{C}$ has a deterministic reconstruction algorithm, it is easy to see that $\mathcal{C}$ also has a deterministic (blackbox) polynomial identity testing (PIT) algorithm. On the other hand, a deterministic PIT implies superpolynomial size lower bounds against $\mathcal{C}$ for an explicit polynomial. Hence, a deterministic reconstruction algorithm for $\mathcal{C}$ is at least as hard as proving superpolynomial lower bounds against $\mathcal{C}$. Thus, much of the research in this area focusses on reconstructing polynomials efficiently computable by weaker variants of arithmetic circuits.

Previous work on the reconstruction problem focussed on polynomials computable by restricted *constant depth* arithmetic circuits and read-once formulas. In particular, depth-2 circuits [4], i.e., interpolation problem, depth-3 circuits with bounded top fan-in and multilinear depth-3 formulas with bounded top fan-in [9], [3]. See [11] for more details on previous work.

In this paper, we consider the model of multilinear formulas. An arithmetic formula, using $+$ and $\times$ operations, is *multilinear* if the formal polynomial computed by each of its subformulas is multilinear. Our main result is a randomized reconstruction algorithm for a class of random multilinear formulas. The algorithm uses as a blackbox a multilinear formula randomly chosen according to a natural distribution (see Section 2 below for details). It succeeds with high probability w.r.t. its internal randomness and the choice of the formula from the distribution. Its output is a multilinear formula of the same size as the hidden formula; it is, in fact, the smallest multilinear formula computing the hidden polynomial. This is the strongest model, and the first one of *super-constant depth*, in arithmetic complexity for which an efficient (even in a randomized or distributional sense) reconstruction algorithm is shown. We further remark that a slight variant of the problem of reconstructing multilinear formulas, even for depth three formulas, is known to be NP-hard. Specifically, Håstad [1] showed that reconstructing the smallest set-multilinear formula (an even weaker model than

[1]Many of the definitions make sense for infinite fields as well.

multilinear formulas) for a given set-multilinear polynomial is NP-hard. This indicates that without some kind of a distributional assumption, it would be unrealistic to hope for a reconstruction algorithm for multilinear formulas. Alternatively, it indicates that there is unlikely to be a *worst-case* reconstruction algorithm for multilinear formulas.

From a broad perspective, reconstructing polynomials from arithmetic complexity classes is, in some sense, analogous to learning concept classes of Boolean functions using membership and equivalence queries. (see Chapter 5 of survey by Shpilka and Yehudayoff [11] for justifying arguments for the analogy to the Boolean world and, more generally, for previous work in this area.) While research on the theory of learnability in the Boolean world has evolved into a mature discipline, thanks to fundamental notions such as PAC learning due to Valiant, research on learnability in the arithmetic world has been gaining momentum only in recent years.

A recurring theme in Boolean and arithmetic domains is that techniques used to prove lower bounds for a model of computation are often helpful in designing learning algorithms for that model. At a very high level, a lower bound proof identifies mathematical properties of a model of computation that capture efficient computation in that model. Thus functions efficiently computable in that model should possess the same or similar properties and they should also be useful in learning such functions. This thesis has been borne out in the Boolean world by several examples, e.g., Fourier approximability of $AC^0$ circuits is useful in both lower bounds and learning algorithms. In the arithmetic world, we see a similar trend, but there are still an abundant number of open questions suggested by this general theme.

Our results in this paper, and in this direction in general, are guided by, and provide supporting evidence to, the thesis mentioned above. One of the key ingredients of our proof is the use of *partial derivative matrices* of polynomials computed in a multilinear formula. We note that properties of partial derivatives of a polynomial have been an important tool in proving lower bounds in a variety of models. In particular, Raz [7] used them to prove lower bounds on multilinear formulas and Raz and Shpilka used them for lower bounds on constant depth circuits. Nisan [6] also used them to prove lower bounds in the noncommutative setting. Thus it is to be expected that properties of partial derivatives of polynomials are useful in reconstruction algorithms. Indeed, Klivans and Shpilka [5] prove that whenever the space of partial derivatives has polynomial dimension, one has polynomial time reconstruction algorithms. This implies reconstruction algorithms for some restricted versions of depth-3 circuits and Arithmetic Branching Programs (ABP's) since their partial derivatives span low-dimensional spaces. This approach, however, cannot be used for multilinear formulas since there are multilinear formulas whose partial derivatives span spaces of exponential dimension. Nevertheless, Raz [7] combines rank arguments about partial derivative matrices and combinatorial arguments based on random restrictions to prove quasipolynomial lower bounds on the multilinear formula complexity of the determinant and permanent polynomials. In this paper, we also exploit rank arguments about partial derivative matrices of polynomials computed in a multilinear formula and combine them with additional structural properties of random multilinear formulas to derive our reconstruction algorithm.

## 2. Definitions and Main Result

We recall that an **arithmetic formula** is a binary tree such that (i) each leaf is labeled by either a variable from $X = \{x_1, \ldots, x_n\}$ or an element of the field $\mathbb{F}$, (ii) each internal node is either $+$ gate or $\times$ gate, and (iii) The incoming edges of a $+$ gate are also labeled by constants from $\mathbb{F}$. A $+$ gate computes the linear combination of its inputs with coefficients given by the constants on the incoming edges of the gate. A $\times$ gate computes the product of its inputs. Each gate $v$ in the formula is naturally associated to a polynomial $p_v \in \mathbb{F}[X]$ computed at $v$. In particular, the polynomial computed at the root (output node) is the polynomial computed by the formula. The *size* of a formula is the number of leaves in the tree. The *(multiplicative) depth* of a node is the number of $\times$ gates on the path from that node to the root. The depth of the formula is the maximum depth of a leaf. An arithmetic formula is said to be **multilinear** if each gate in it computes a multilinear polynomial, i.e., in each of its monomials the power of every input variable is at most one.

**Definition 2.1.** ***Syntactic Multilinear Formulas:*** *Let $\Phi$ be an arithmetic formula over $X = \{x_1, \ldots, x_n\}$. Let $\Phi_v$ denote the subformula rooted at a node $v$ and $X_v$ be the set of variables that appear in $\Phi_v$. Then, $\Phi$ is said to be* syntactic multilinear *if for every product gate $v = v_1 \times v_2$ of $\Phi$, the sets $X_{v_l}$ and $X_{v_2}$ are disjoint.*

Note that for any multilinear formula, there exists a syntactic multilinear formula of the same size that computes the same polynomial (see [7]). Hence, we often omit the word "syntactic" while referring to multilinear formulas. Moreover, any syntactic multilinear formula can be converted into a syntactic multilinear formula with alternating layers of $+$ and $\times$ gates with only a polynomial blow-up in size (see [8]).

*A Natural Distribution on the set of Multilinear Formulas:*

Our reconstruction algorithm uses, as a blackbox, a random multilinear formula drawn according to a distribution as defined below. Informally, this distribution constructs a binary[2] tree with $+$ and $\times$ gates at alternating levels (with a $+$ gate at the root). Each $+$ gate computes a random linear

[2]we assume this for the clarity of presentation and handle the $k$-ary case in Section A.

combination of its inputs over $\mathbb{F}$. Moving down the tree, at each $\times$ gate, we partition the variables into two equal-sized sets and recursively build a subformula rooted at each of this $\times$ gate. We stop the recursion when the number of variables is small enough (we choose this to be about $\log^3 n$ for technical reasons and ensure an error probability of $1/\mathrm{poly}(n)$.)

A formal definition of the distribution follows:

Let $\mathcal{M}(X, \mathbb{F})$ be the set of all possible syntactic multilinear formulas over the variable set $X = \{x_1, \ldots, x_n\}$ and a (sufficiently large) finite field $\mathbb{F}$. We propose the following method $\texttt{SAMPLE}(X, \mathbb{F})$ to sample a random syntactic multilinear formula from the set $\mathcal{M}(X, \mathbb{F})$, thereby inducing a natural P-samplable distribution $\mathcal{D}(X, \mathbb{F})$ on the set $\mathcal{M}(X, \mathbb{F})$. This distribution also depends on an integer parameter $\beta_n$, which we assume to be $\Theta(\log^3 n)$.

**Sampling Method** $\texttt{SAMPLE}(X, \mathbb{F})$:

> Step 1: $\Psi \leftarrow \texttt{CONSTRUCT}(X, +)$, where $\texttt{CONSTRUCT}(X, op)$ is defined below.
> Step 2: Let $W$ be the set of wires in $\Psi$ incident to a $+$ gate. Let $\Phi$ be the syntactic multilinear arithmetic formula obtained by labeling each $w_i \in W$ by a randomly and independently chosen $c_i \in_R \mathbb{F}$.
> Step 3: **return**$(\Phi)$.

$\texttt{CONSTRUCT}(X, op)$:

> Case 1: $|X| \leq \beta_n$. Let $\Psi$ be the formula with a $+$ gate at the root that has wires incident to it from each $x_i \in X$.
> Case 2: $|X| > \beta_n$ and $op = \times$. Partition $X$ randomly into two equal sized sets $X_1, X_2$ and let $\Psi_1 \leftarrow \texttt{CONSTRUCT}(X_1, +)$, $\Psi_2 \leftarrow \texttt{CONSTRUCT}(X_2, +)$. Let $\Psi$ be the formula with a $\times$ gate at the root and $\Psi_1, \Psi_2$ as its two children.
> Case 3: $|X| > \beta_n$ and $op = +$. Let $\Psi_1 \leftarrow \texttt{CONSTRUCT}(X, \times)$, $\Psi_2 \leftarrow \texttt{CONSTRUCT}(X, \times)$. Let $\Psi$ be the formula with a $+$ gate at the root and $\Psi_1, \Psi_2$ as its two children.
> Step: **return**$(\Psi)$.

We now state our main reconstruction result for multilinear formulas.

**Theorem 2.2.** *Let $\Phi \sim \mathcal{D}(X, \mathbb{F})$ be a random multilinear formula sampled as above and let $\hat{\Phi} \in \mathbb{F}[X]$ be the polynomial computed by $\Phi$. Then, there is an $n^{O(1)}$-time randomized algorithm $\mathcal{A}$ which, given blackbox access to $\hat{\Phi}$, constructs a syntactic multilinear formula $\Phi_{\mathcal{A}}$ of size at most size$(\Phi)$ and such that*

$$\Pr[\hat{\Phi}_{\mathcal{A}} \neq \hat{\Phi}] \leq \frac{n^{O(1)}}{|\mathbb{F}|} + \frac{1}{n^{\Omega(1)}},$$

*where the probability is taken over the randomness in the choice of $\Phi$ and the internal randomness of $\mathcal{A}$.*

## 3. Basic Idea and approach

Suppose we have blackbox access to the output polynomial $f$ of a random multilinear formula $\Phi$. By querying $f$ at points of our choice, we want to recover $\Phi$. How do we do so? We give an overview of our approach to do this.

**Determining the nature of the output gate:** Observe that if the output node was a $\times$ gate then the output would be a reducible polynomial [3]. The converse is not true in general. That is, it can happen that the output gate is a $+$ gate and $f$ is reducible as well. At this point we invoke the assumption that the formula $\Phi$ is chosen randomly and deduce that with high probability over the random choice of $\Phi$ the output node is a $\times$ node if and only if $f$ is reducible (Lemma 5.9). Thus, we can use the blackbox factoring algorithm of Kaltofen [2] (or alternatively the multilinear factoring algorithm of Shpilka and Volkovich [10]) to determine whether $f$ is reducible and this helps us answer our first question. The next thing that we would like to do is get blackbox access to the two children. Once we have that we can recursively do the reconstruction of the two subformulas. There are two cases depending on the nature of the output gate.

**Case I: Output node is a $\times$ gate.** In this case we factor $f$ using Kaltofen's algorithm. Now it can happen (in rare circumstances) that the number of factors of $f$ is larger than the number of children of the output node. For a generic (i.e. randomly chosen) formula $\Phi$ these two quantities will however be equal (Lemma 5.9) so that Kaltofen's algorithm provides blackbox access to the two children of the output node. We then recursively compute the formulas for the two children.

**Case II: Output node is a $+$ gate.** In this case we need to go one level deeper. The two children of the output node are $\times$ gates (except when we are in the base case) so that the output polynomial $f$ is of the form

$$f = A \cdot B + C \cdot D.$$

Our aim will be to obtain blackbox access to the four 'grandchildren' $A, B, C$ and $D$. If we can do that then we can recursively compute formulas for these polynomials and we would be done. At this point we use the fact that we are dealing with (syntactic) multilinear formulas. It means that there exists a partition of the set of variables into four (disjoint) subsets $\bar{u}, \bar{v}, \bar{x}$ and $\bar{y}$ such that

$$f(\bar{u}, \bar{v}, \bar{x}, \bar{y}) = A(\bar{u}, \bar{v}) \cdot B(\bar{x}, \bar{y}) + C(\bar{v}, \bar{x}) \cdot D(\bar{u}, \bar{y}). \quad (1)$$

In general this partition of the set of variables can be arbitrary in which case it becomes much more difficult to find $\Phi$. However, when $\Phi$ is random then with high probability all these sets are roughly of the same size

[3] If one of the children was a constant then the subtree rooted at that node can be discarded and we would have a smaller formula computing the same polynomial

(Lemma 5.1). Now it turns out that we can exploit the ideas in the lower bound proof of Raz [7] to find this partition of the set of variables. Very roughly, the idea is that for the right partition the rank of a certain related matrix will be very small whereas for every other partition the rank of this matrix will be much larger. This is the one of the key technical arguments (Theorem 5.6) in our work and is described in its proof sketch. For now assume that we know the subsets $\bar{u}, \bar{v}, \bar{x}$ and $\bar{y}$. Knowing these subsets, how do we obtain blackbox access to $f$? The idea is that if in equation (1) we substitute each $\bar{u}$-variable and each $\bar{v}$-variable to some random values say $\bar{u} = \bar{a}$ and $\bar{v} = \bar{b}$ then $A(\bar{a}, \bar{b})$ becomes a constant so that the degree of $(A \cdot B)$ drops down after this substitution (with high probability, this substitution does not change the degree of $C \cdot D$). This means that the homogeneous part of largest degree of $f(\bar{a}, \bar{b}, \bar{x}, \bar{y})$ is a product of the homogeneous parts of largest degrees of $C(\bar{b}, \bar{x})$ and $D(\bar{a}, \bar{y})$. Thus factoring the homogeneous part of largest degree of $f$ gives us blackbox access to the largest degree homogeneous parts of $C(\bar{b}, \bar{x})$ and $D(\bar{a}, \bar{y})$. This idea can be extended suitably (see Lemma 5.5) to obtain blackbox access to the whole of each polynomial $A, B, C$ and $D$. This completes our brief overview of the reconstruction algorithm for multilinear formulas.

## 4. Preliminaries and Notations

**Lemma 4.1** (Chernoff's bound)**.** *Let $\zeta_1, \ldots, \zeta_n$ be independent uniform* 0-1 *random variables . Then,*

$$\Pr[\ (1-\delta)n/2 \leq \sum_i \zeta_i \leq (1+\delta)n/2\ ] \geq 1 - 2\exp(-\delta^2 n/8).$$

**Lemma 4.2** (DeMillo-Lipton-Schwartz-Zippel)**.** *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of degree $d \geq 0$. Let $S$ be a finite subset of $\mathbb{F}$ and let $r_1, \ldots, r_n$ be selected randomly from $S$. Then*

$$\Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|}$$

The above lemma automatically results in the following PIT algorithm which succeeds with probability $\geq 1 - \frac{d}{|S|}$.

**Algorithm 1** (Blackbox PIT)**.** *Given blackbox access to a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree $d$, query $f(r_1, r_2, \ldots, r_n)$ to the blackbox for $r_1, \ldots, r_n \in_R S$, where $S$ is any finite subset of $\mathbb{F}$. Conclude $f = 0$ iff $f(r_1, r_2, \ldots, r_n) = 0$.*

**Kaltofen's Blackbox Factoring:** We state the multivariate blackbox factoring algorithm by Kaltofen [2] (also see [10]) in context of multilinear polynomials,

**Lemma 4.3** (Kaltofen's Blackbox Factoring)**.** *There is a randomized polynomial-time algorithm that, given blackbox access to a multilinear polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, with probability $1 - 2^{-\Omega(n)}$, outputs blackboxes to all the irreducible factors of $f$.*

**Notation:** $[n]$ denotes the set $\{1, 2, \ldots, n\}$. For a polynomial $f$, $f^{[d]}$ denotes the homogenous degree-$d$ part of $f$. Tuples would be denoted by placing a bar over a letter, e.g. $\bar{x}$. For a tuple $\bar{\beta} = (\beta_1, \ldots, \beta_n)$, $i\bar{\beta}$ would denote the tuple $(i\beta_1, \ldots, i\beta_n)$. For an arithmetic formula $\Phi$, the polynomial computed at the root is denoted by $\hat{\Phi}$.

## 5. Reconstructing Multilinear Formulas

### 5.1. Structural Properties of Multilinear Formulas from $\mathcal{D}(X, \mathbb{F})$

Before we sketch the proof of Theorem 2.2, we state and examine some structural properties of random multilinear formulas which would be essential for our algorithm to work. Due to space constraints, the proofs of the lemmas here have been omitted.

Our first lemma says for the variables in the subformula rooted at a $+$ gate, the two partitions induced by the children ($\times$ gates) of that gate intersect more or less "transversally," i.e., each block of either partition is split nontrivially (in fact in a rather balanced way) by the other partition. Moreover, a child polynomial of a $\times$ gate (a grandchild of the $+$ gate) here is not annihilated by zeroing out either subset of its variables induced by the partition at the sibling product gate.

**Lemma 5.1.** *Let $\Phi \sim \mathcal{D}(X, \mathbb{F})$. Then, for all nodes of $\Phi$, the following hold with probability at least $1 - \frac{n^{O(1)}}{|\mathbb{F}|} - \frac{1}{n^{\Omega(1)}}$:*

1) *The polynomial computed by a node at (multiplicative) depth $h$ is a homogenous polynomial of degree $\frac{n}{\beta_n 2^h}$.*
2) *The polynomial computed at a $+$ gate is of the form $\alpha.A(\bar{v}, \bar{u})B(\bar{x}, \bar{y}) + \beta.C(\bar{v}, \bar{x})D(\bar{u}, \bar{y})$ where for all $\bar{p} \in \{\bar{v}, \bar{u}, \bar{x}, \bar{y}\}$, $|\bar{p}| \geq \frac{1}{8}|\{\bar{v} \cup \bar{u} \cup \bar{x} \cup \bar{y}\}|$.*
3) *In the above polynomial computed at a $+$ gate, for all $R \in \{A, B, C, D\}$, say $R(\bar{p}, \bar{q})$, $R(\bar{0}, \bar{q}) \neq 0$ and $R(\bar{p}, \bar{0}) \neq 0$.*

Given a multilinear polynomial $f$ over two variable sets $Y = \{y_1, \ldots, y_m\}$ and $Z = \{z_1, \ldots, z_n\}$, define $M_f$ as a $2^m \times 2^n$ matrix whose $(p, q)$ entry, $p \subseteq Y$ and $q \subseteq Z$ is the coefficient of the monomial $pq$ in $f$. The rank of $M_f$ in this case is denoted by $\mathrm{Rank}_{YZ}(f)$. We will use the following properties of the partial derivatives matrix.

**Lemma 5.2** ([7])**.** *Given two multilinear polynomials $f$ and $g$ over the variable set $Y \cup Z$,*

1) $\mathrm{Rank}_{YZ}(f + g) \leq \mathrm{Rank}_{YZ}(f) + \mathrm{Rank}_{YZ}(g)$,
2) $\mathrm{Rank}_{YZ}(f.g) = \mathrm{Rank}_{YZ}(f).\mathrm{Rank}_{YZ}(g)$ *if $f$ and $g$ are polynomials on disjoint sets of variables, and*
3) $\mathrm{Rank}_{YZ}(f) \leq 2^{min(Y(f), Z(f))}$ *where $Y(f)$ and $Z(f)$ are the number of $Y$ and $Z$ variables that occur in $f$.*

We next show that a random linear combination of two multilinear polynomials can only increase the rank w.h.p.

**Lemma 5.3.** *Let $f$ and $g$ be two multilinear polynomials over the variable set $Y \cup Z$ and field $\mathbb{F}$. Let $S \subset \mathbb{F}$. For $\alpha, \beta$ chosen uniformly at random from $S \subseteq \mathbb{F}$, let*

*$p$ be the probability that* $\mathrm{Rank}_{YZ}(\alpha \cdot f + \beta \cdot g) \geq \max\{\mathrm{Rank}_{YZ}(f), \mathrm{Rank}_{YZ}(g)\}$. *Then we have*

$$p \geq 1 - \frac{2^{\min\{|Y|,|Z|\}}}{|S|}.$$

*5.2. Simulating Blackbox Access to Subformulas*

Our reconstruction algorithm will be recursive on the structure of the (unknown, random) multilinear formula. Hence, we will need to simulate blackbox access to its components using blackbox access to the polynomial/formula itself. The next lemma shows this for the homogenous component of a given degree and the theorem below for the grandchildren of a $+$ node.

**Lemma 5.4.** *Let $\mathbb{F}$ be a field with at least $d+1$ elements and let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a degree $d$ polynomial. Given blackbox access to $f$ we can simulate blackbox access to $f^{[r]}$'s, where $f^{[r]}$ denotes the homogenous degree-$r$ part of $f$.*

**Theorem 5.5.** *Let $\{\{\bar{v}\}, \{\bar{u}\}, \{\bar{x}\}, \{\bar{y}\}\}$ be a partition of $\{x_1, \ldots, x_n\}$ and $f(\bar{v}, \bar{u}, \bar{x}, \bar{y}) = A(\bar{v}, \bar{u})B(\bar{x}, \bar{y}) + C(\bar{v}, \bar{x})D(\bar{u}, \bar{y})$ be a non-zero polynomial such that,*

1) *A,B,C,D are homogenous multilinear polynomials over the indicated variable sets,*
2) *either $\deg(AB) \neq \deg(CD)$ or $\deg(A) = \deg(B) = \deg(C) = \deg(D)$,*
3) *for all $R \in \{A, B, C, D\}$, say $R(\bar{p}, \bar{q})$, $R(\bar{0}, \bar{q}) \neq 0$ and $R(\bar{p}, \bar{0}) \neq 0$.*
4) *for all $\bar{p} \in \{\bar{v}, \bar{u}, \bar{x}, \bar{y}\}$, $|\bar{p}| \geq \delta n$, for some $\delta > 0$*

*Then there is an $n^{O(1)}$-time randomized algorithm that, given blackbox access to $f$ and the partition $\{\{\bar{v}\}, \{\bar{u}\}, \{\bar{x}\}, \{\bar{y}\}\}$, constructs blackboxes for A,B,C,D with probability at least $1 - \frac{n^{O(1)}}{|\mathbb{F}|} - \frac{1}{2^{\Omega(n)}}$.*

*Proof:* The proof follows from the algorithm TRICKLEDOWN below.

**Input:** The partition $\{\{\bar{v}\}, \{\bar{u}\}, \{\bar{x}\}, \{\bar{y}\}\}$ and an oracle for $A(\bar{v}, \bar{u})B(\bar{x}, \bar{y}) + C(\bar{v}, \bar{x})D(\bar{u}, \bar{y})$ where $A$,$B$,$C$,$D$ are polynomials satisfying the above stated properties.

**Output:** Blackboxes for $A$,$B$,$C$,$D$.

**Algorithm:** TRICKLEDOWN

**Step 1:** Using blackbox for $f = AB + CD$, construct blackboxes for $f^{[i]}$'s for all $i \in [n]$.

**Step 2:** For $i \in [n]$, using blackbox PIT, determine if $f^{[i]} \neq 0$. If there is only one such $i$ then proceed to the next step. Otherwise let $f^{[i]}$, $f^{[j]}$ be non-zero. For $f^{[i]}(\bar{v}, \bar{u}, \bar{x}, \bar{y})$, determine using blackbox PIT, if $f^{[i]}(\bar{v}, \bar{u}, \bar{0}, \bar{0})$ is 0. If yes, conclude $f^{[i]} = AB$ and $f^{[j]} = CD$, else the other way. Using Kaltofen's factoring algorithm, construct blackboxes for irreducible factors of $A(\bar{v}, \bar{u})B(\bar{x}, \bar{y})$. For each factor $h(\bar{v}, \bar{u}, \bar{x}, \bar{y})$, determine, using blackbox PIT, if $h(\bar{0}, \bar{0}, \bar{x}, \bar{y})$ is 0. If yes, conclude it is a factor of $A$ else $B$. Similarly, construct blackboxes for $C$ and $D$.

**Step 3:** Determining degrees of $A$,$B$,$C$,$D$. Using Kaltofen's factoring algorithm, gain blackbox access to irreducible factors of $f(\bar{v}, \bar{u}, \bar{0}, \bar{0}) = C(\bar{v}, \bar{0})D(\bar{u}, \bar{0})$. For each factor $h$, determine, using blackbox PIT, if $h$ becomes the zero polynomial after instantiating $\bar{v}$ to $\bar{0}$. If yes it is a factor of $C(\bar{v}, \bar{0})$ else $D(\bar{u}, \bar{0})$. Similarly, construct blackboxes for $C(\bar{0}, \bar{x})$ and $D(\bar{0}, \bar{y})$. Having constructed blackboxes for $C(\bar{v}, \bar{0})$ and $D(\bar{u}, \bar{0})$, conclude $d = \deg(C) = \log\left(\frac{C(2\bar{\alpha}, \bar{0})}{C(\bar{\alpha}, \bar{0})}\right)$ for a randomly chosen $\bar{\alpha} \in \mathbb{F}^{|\bar{v}|}$, and similarly for $D$, $A$, $B$.

**Step 4:** Constructing blackbox for $C$. To determine $C(\bar{\alpha}, \bar{\beta})$, for any $\bar{\alpha} \in \mathbb{F}^{|\bar{v}|}$, $\bar{\beta} \in \mathbb{F}^{|\bar{x}|}$ we will make the substitution $\bar{x} := \bar{\beta}$ and $\bar{y} := \bar{\gamma}$ for $\bar{\gamma} \in_R \mathbb{F}^{|\bar{y}|}$. In the ensuing discussion we shall denote by $\hat{g}(\bar{v}, \bar{u})$ the polynomial obtained by making the above substituion in a polynomial $g(\bar{v}, \bar{u}, \bar{x}, \bar{y})$. i.e. $\hat{g} \stackrel{\text{def}}{=} g(\bar{v}, \bar{u}, \bar{\beta}, \bar{\gamma})$. Then, $\hat{f} := f(\bar{v}, \bar{u}, \bar{\beta}, \bar{\gamma})$ is of the form

$$\underbrace{A(\bar{v}, \bar{u})B(\bar{\beta}, \bar{\gamma})}_{\text{only degree } \deg(A) \text{ terms}} + \underbrace{C(\bar{v}, \bar{\beta})D(\bar{u}, \bar{\gamma})}_{\text{terms can have degree} > \deg(A)}$$
$$= A(\bar{v}, \bar{u})B(\bar{\beta}, \bar{\gamma}) + \hat{C}(\bar{v})\hat{D}(\bar{u}).$$

Then,

$$\hat{C}(\bar{v}) = \hat{C}^{[d]}(\bar{v}) \ldots + \hat{C}^{[1]}(\bar{v}) + C(\bar{0}, \bar{\beta})$$

while

$$\hat{D}(\bar{u}) = \hat{D}^{[d]}(\bar{u}) \ldots + \hat{D}^{[1]}(\bar{u}) + D(\bar{0}, \bar{\gamma}).$$

Note that $\hat{f}^{[2d]} = \hat{C}^{[d]}(\bar{v}) \cdot \hat{D}^{[d]}(\bar{u})$. Using Kaltofen's algorithm, obtain blackboxes for $\hat{C}^{[d]}(\bar{v})$ and $\hat{D}^{[d]}(\bar{u})$ using the blackbox for $\hat{f}^{[2d]}$. As Kaltofen's algorithm gives blackboxes for irreducible factors of $C^{[d]}(\bar{v})D^{[d]}(\bar{u})$ and any such factor depends on either $\bar{v}$ or $\bar{u}$, to find out if $h(\bar{v}, \bar{u})$ depends on $\bar{u}$ use blackbox PIT on $h(\bar{v}, \bar{0})$.

**Step 5:** Constructing blackboxes for $\hat{C}^{[i]}(\bar{v})$ and $\hat{D}^{[i]}(\bar{u})$ for $i \in [d-1]$ . Having gained blackboxes for $\hat{C}^{[d]}(\bar{v})$ and $\hat{D}^{[d]}(\bar{u})$ we note that,

$$\hat{f}(\bar{v}, \bar{u})^{[2d-1]} = \hat{C}^{[d]} \cdot \hat{D}^{[d-1]} + \hat{C}^{[d-1]} \cdot \hat{D}^{[d]} \quad (2)$$

Also

$$\hat{f}(\bar{v}, 2\bar{u})^{[2d-1]} = 2^{d-1}\hat{C}^{[d]} \cdot \hat{D}^{[d-1]} + 2^{d}\hat{C}^{[d-1]} \cdot \hat{D}^{[d]} \quad (3)$$

Solving (2) and (3) for $\hat{C}^{[d-1]}$, we get that $\hat{C}^{[d-1]}(\bar{v})$ equals

$$\frac{1}{\hat{D}^{[d]}(\bar{u})}\left[\frac{1}{2^{d-1}}\hat{f}(\bar{v}, 2\bar{u})^{[2d-1]} - \hat{f}(\bar{v}, \bar{u})^{[2d-1]}\right]$$

As we have blackbox access to $\hat{f}^{[2d-1]}$ and $\hat{D}^{[d]}(\bar{u})$, we have blackbox access to $\hat{C}^{[d-1]}(\bar{v})$ (after instantiating $\bar{u}$ randomly to avoid making the denominator vanish in the above equation). Similarly we have blackbox access

to $\hat{D}^{[d-1]}(\bar{u})$. In general, after constructing blackboxes for $\hat{C}^{[r]}(\bar{v})$ and $\hat{D}^{[r]}(\bar{u})$ for all $r \in [d'+1 : d]$ and proceeding as above, one obtains the following expression for $\hat{C}^{[d']}(\bar{v})$.

$$\hat{C}^{[d']}(\bar{v}) = E_1 \cdot (E_2 - E_3)$$

where

$$\begin{aligned} E_1 &= \frac{2^{d'}}{(2^d - 2^{d'})\hat{D}^{[d]}(\bar{u})} \\ E_2 &= \frac{\hat{f}(\bar{v}, 2\bar{u})^{[d+d']}}{2^{d'}} - \hat{f}(\bar{v}, \bar{u})^{[d+d']} \\ E_3 &= \sum_{i=d'+1}^{d-1} (2^{d-i} - 1)\hat{C}^{[i]}(\bar{v})\hat{D}^{[d+d'-i]}(\bar{u}) \end{aligned}$$

Hence using the above procedure, blackboxes for $\hat{C}^{[d']}(\bar{v})$, for all $d' \in [d]$, can be constructed. Also, using the blackbox for $C(\bar{0}, \bar{x})$ constructed in Step 3 determine $C(\bar{0}, \bar{\beta})$. This completes our blackbox for $C(\bar{v}, \bar{\beta})$.

**Step 6:** Repeat the above 3 steps similarly with the correct parameters to construct blackboxes for $A$,$B$ and $D$.

■

### *5.3. The Reconstruction Algorithm* RECONSTRUCT$(\mathcal{O}_{\hat{\Phi}}, X, \mathbb{F}, m)$

We are now ready to present the reconstruction algorithm for random multilinear formulas.

**Input:** oracle $\mathcal{O}_{\hat{\Phi}}$ for polynomial $\hat{\Phi}$ computable by a multilinear formula $\Phi$ sampled using SAMPLE$(X, \mathbb{F})$ where $X = \{x_1, \ldots, x_n\}$ and size $m$ of the seed partition[4]($m = \Theta(\log n)$).

**Output:** multilinear formula $\Psi$ such that $|\Psi| \leq |\Phi|$ and $\hat{\Psi} = \hat{\Phi}$, or else FAIL.

**Step 1:** *Determining linearity*: For any $x_i \in X$, $f_i = \hat{\Phi}|_{x_i=1} - \hat{\Phi}|_{x_i=0}$ is the coefficient polynomial of $x_i$ in $\hat{\Phi}$. For all $f_i$'s, using blackbox PIT on $f_i|_{x_j=1} - f_i|_{x_j=0}$, determine if $f_i$ depends on $x_j$. If for all $x_i$ with a non-zero $f_i$, $f_i$ does not depend on $X$, then $\hat{\Phi}$ is linear and in this case simply interpolate $\hat{\Phi}$ exactly and output a $\Sigma$-circuit for it.

**Step 2:** *Reducible* $\hat{\Phi}$: Using Kaltofen's factoring algorithm construct oracles for irreducible factors $h_i$'s, of $\hat{\Phi}$. If $\hat{\Phi}$ is irreducible proceed to the next step. Else using blackbox PIT, as described in the previous step, determine the variable sets of these factors. Recursively using RECONSTRUCT, construct formulas $\Psi_i$'s for $h_i$'s. If RECONSTRUCT fails on any $h_i$ output FAIL. Else, output a formula with $\times$ gate at the root and $\Psi_i$'s as its children.

[4]size of the seed partition is kept unchanged while recursing.

**Step 3:** *Determining a seed partition*: Let $\hat{\Phi} = A(\bar{v}, \bar{u})B(\bar{x}, \bar{y}) + C(\bar{v}, \bar{x})D(\bar{u}, \bar{y})$. Randomly choose an $m$-sized subset $S$ of $X$. In $\hat{\Phi}$, instantiate the variables in $X \setminus S$ to random values over $\mathbb{F}$ to get $\hat{\Phi}_S = A_S(\bar{v}_S, \bar{u}_S)B_S(\bar{x}_S, \bar{y}_S) + C_S(\bar{v}_S, \bar{x}_S)D_S(\bar{u}_S, \bar{y}_S)$ and interpolate it in $n^{O(1)}$ time. Iterate over all possible partitions $\{\{\bar{v}''\}, \{\bar{u}''\}, \{\bar{x}''\}, \{\bar{y}''\}\}$ of $S$ such that the size of each set in them is at least $\gamma m$ (for a small enough $\gamma$) and let $\{\{\bar{v}'\}, \{\bar{u}'\}, \{\bar{x}'\}, \{\bar{y}'\}\}$ be a partition such that $\mathrm{Rank}_{\{\bar{v}'\}\{\bar{y}'\}}(\hat{\Phi}_S|_{\bar{v}',\bar{y}'}) \leq 2$ and $\mathrm{Rank}_{\{\bar{u}'\}\{\bar{x}'\}}(\hat{\Phi}_S|_{\bar{u}',\bar{x}'}) \leq 2$ where $\hat{\Phi}_S|_{\bar{v}',\bar{y}'}$ is $\hat{\Phi}_S$ with variables in $S \setminus \{\bar{v}', \bar{y}'\}$ instantiated to random values in $\mathbb{F}$ and similarly for $\hat{\Phi}_S|_{\bar{u}',\bar{x}'}$. Having interpolated $\hat{\Phi}_S$, this can be done in $n^{O(1)}$ time using Gaussian elimination as there are $2^{O(\log n)}$ such possible partitions and the partial derivative matrix on $O(\log n)$ variables is of size at most $2^{O(\log n)}$.

**Step 4:** *Extending the seed partition* $\{\{\bar{v}'\}, \{\bar{u}'\}, \{\bar{x}'\}, \{\bar{y}'\}\}$: For all $x_i \in X \setminus S$ do the following. Let $S_i = S \cup \{x_i\}$. In $\hat{\Phi}$, instantiate the variables in $X \setminus S_i$ to random values over $\mathbb{F}$ to get $\hat{\Phi}_{S_i}$ and interpolate it in $2^{O(\log n)}$ time. Iterate over the following 4 partitions of $S_i$, $\{\{\bar{v}', x_i\}, \{\bar{u}'\}, \{\bar{x}'\}, \{\bar{y}'\}\}$, $\{\{\bar{v}'\}, \{\bar{u}', x_i\}, \{\bar{x}'\}, \{\bar{y}'\}\}$, $\{\{\bar{v}'\}, \{\bar{u}'\}, \{\bar{x}', x_i\}, \{\bar{y}'\}\}$, $\{\{\bar{v}'\}, \{\bar{u}'\}, \{\bar{x}'\}, \{\bar{y}', x_i\}\}$ and determine the partition $\{\{\bar{v}''\}, \{\bar{u}''\}, \{\bar{x}''\}, \{\bar{y}''\}\}$ such that, $\mathrm{Rank}_{\{\bar{v}''\}\{\bar{y}''\}}(\hat{\Phi}_{S_i}|_{\bar{v}'',\bar{y}''}) \leq 2$ and $\mathrm{Rank}_{\{\bar{u}''\}\{\bar{x}''\}}(\hat{\Phi}_{S_i}|_{\bar{u}'',\bar{x}''}) \leq 2$ where $\hat{\Phi}_{S_i}|_{\bar{v}'',\bar{y}''}$ is $\hat{\Phi}_{S_i}$ with variables in $S_i \setminus \{\bar{v}'', \bar{y}''\}$ instantiated to random values in $\mathbb{F}$. Attach $x_i$ to the appropriate block of the seed partition. This can be done in $2^{O(\log n)}$ time.

**Step 5:** Using TRICKLEDOWN algorithm and the above determined partition $\{\{\bar{v}\}, \{\bar{u}\}, \{\bar{x}\}, \{\bar{y}\}\}$ of $X$ construct oracles for $A$,$B$,$C$,$D$. Then, recursively using RECONSTRUCT, construct formulas $\Psi_R$'s for $R \in \{A, B, C, D\}$. If RECONSTRUCT fails on for any of them output FAIL. Else, let $\Psi_{AB}$ be the formula with $\times$ gate at the root and $\Psi_A$, $\Psi_B$ as its children. Output a formula $\Psi$ with $+$ gate at the root and $\Psi_{AB}$, $\Psi_{CD}$ as its children.

This completes the description of the algorithm RECONSTRUCT. Algorithm $\mathcal{A}$ of Theorem 2.2 is now essentially RECONSTRUCT, returning $\Psi$ using blackbox calls to $\hat{\Phi}$. (If RECONSTRUCT outputs FAIL, $\mathcal{A}$ outputs a random multilinear formula.) The bound on the running time of $\mathcal{A}$ is obvious. For correctness, it's crucial to show that the partition determined by steps 3 and 4 is, w.h.p., the original partition of $\Phi$. We do this in the next section. This will complete the proof of Theorem 2.2. ■

### *5.4. Uniqueness of the Seed Partition*

Our main result for this section is Theorem 5.6, which shows that for a large $\mathbb{F}$, Steps 3 and 4 of the

RECONSTRUCT method determine the correct partition w.h.p.. We need some preliminary discussion leading up to the theorem.

*Placement of random field elements on the wires of a random multilinear formula drawn by* $\mathtt{SAMPLE}(X,\mathbb{F})$*:*

While sampling a multilinear formula from the set $\mathcal{M}(X,\mathbb{F})$ we first sampled a formula without any field elements using the method CONSTRUCT and later placed field elements, chosen independently and uniformly from $\mathbb{F}$, on its wires. Also note that, distinct wires originating from any of the $x_i$'s, have distinct independent uniform r.v.'s on them. For instance consider a multilinear formula on $X$ and that every $x_i$ has at most one wire originating from it. Let the polynomial computed by the formula be $\sum_{k=1}^{N}\alpha_k.M_k$ where $M_i$'s are multilinear monomials. Now, if we place an r.v. $r_i$ on the wire from $x_i$ then a term like $\alpha.x_1x_3x_n$ becomes $\alpha.r_1r_3r_n.x_1x_3x_n$. Hence essentially, the coefficient of a multilinear monomial $M$ on $X$, takes the form $\alpha_M.M_r$ where $M_r$ is the multilinear monomial $\Pi_{x_i\in M}r_i$ and each $\alpha_M$ is independent of $r_i$'s. These observations help us prove Lemma 5.8 showing that for every monomial $M$ there is a set of $\log N$ monomials containing $M$ such that the set of coefficients of these monomials is mutually independent. Also, it is easy to note that this remains true even after instantiating variables to random values over $\mathbb{F}$.

*Instantiating* $n-m$ *variables to random field elements in Step 3 of the* RECONSTRUCT *method:*

Let $\Phi$ be a random multilinear formula sampled using $\mathtt{SAMPLE}(X,\mathbb{F})$ and let $\hat{\Phi} = A(\bar{v},\bar{u})B(\bar{x},\bar{y}) + C(\bar{v},\bar{x})D(\bar{u},\bar{y})$. In Step 3 of the RECONSTRUCT method, we choose an $m$-sized subset $S$ of $X$ randomly and instantiate the variables in $X\setminus S$ to random values over $\mathbb{F}$ to get $\hat{\Phi}_S = A_S(\bar{v}_S,\bar{u}_S)B_S(\bar{x}_S,\bar{y}_S)+C_S(\bar{v}_S,\bar{x}_S)D_S(\bar{u}_S,\bar{y}_S)$. Using Chernoff bounds it easily follows that w.h.p., sizes of the sets $\bar{v}_S$, etc., are $\Omega(m)$. Let $Y=S$ and $Z=X\setminus S$. In the SAMPLE method, partitioning the set $Y\cup Z$ at a $\times$ gate (where $|Y|\le|Z|$) into two equal-sized sets $\{\bar{a}\},\{\bar{b}\}$ can be viewed as follows: label the $y_i$'s in $Y$ with independent uniform 0-1 values, include the $y_i$'s with label 0 in $\{\bar{a}\}$ and label 1 in $\{\bar{b}\}$, and finally, place the $Z$ variables randomly to make $|\bar{a}| = |\bar{b}|$. It is now easy to see that in the above expression of $\hat{\Phi}_S$, the polynomials $A_S,B_S,C_S,D_S$ are close in distribution to a multilinear formula sampled by the following method on their respective variable sets.

**Sampling Method** $\mathtt{SAMPLE}_2(X,\mathbb{F})$**:**

Step 1: $\Psi \leftarrow \mathtt{CONSTRUCT}_2(X,+)$.
Step 2: Let $W$ be the set of wires in $\Psi$ incident to a $+$ gate. Let $\Phi$ be the syntactic multilinear arithmetic formula obtained by labeling each $w_i\in W$ by a randomly and independently chosen $c_i\in_R\mathbb{F}$.
Step 3: **return** $\Phi$.

where $\mathtt{CONSTRUCT}_2(X,op)$:

Case 1: $X=\{x_i\}$. Let $\Psi$ be a single $+$ gate with $x_i$ as one input and the field element 1 as the other input.
Case 2: $op=\times$. Label each $x_i\in X$ with independent uniformly chosen 0-1 values. Include the $x_i$'s labeled 0 in a set $X_1$ and the rest in $X_2$. If either $X_i$ is empty then repeat. Let $\Psi_1\leftarrow\mathtt{CONSTRUCT}_2(X_1,+)$, $\Psi_2\leftarrow\mathtt{CONSTRUCT}_2(X_2,+)$. Let $\Psi$ be the formula with a $\times$ gate at the root and $\Psi_1$ and $\Psi_2$ as its two children.
Case 3: $op=+$. Let $\Psi_1\leftarrow\mathtt{CONSTRUCT}_2(X,\times)$, $\Psi_2\leftarrow\mathtt{CONSTRUCT}_2(X,\times)$. Let $\Psi$ be the formula with a $+$ gate at the root and $\Psi_1$ and $\Psi_2$ as its two children.
Step: **return** $\Psi$.

**Theorem 5.6** (Uniqueness of Partition)**.** *Let* $\{\{\bar{a}\},\{\bar{b}\}\}$ *and* $\{\{\bar{c}\},\{\bar{d}\}\}$ *be partitions of* $\{\bar{y}\}\cup\{\bar{z}\}$*, where* $|\bar{a}|$*,* $|\bar{b}|$*,* $|\bar{c}|$*,* $|\bar{d}|$*,* $|\bar{y}|$*,* $|\bar{z}|$ *are all* $\Omega(m)$*. Let* $A(\bar{a})$*,* $B(\bar{b})$*,* $C(\bar{c})$*,* $D(\bar{d})$ *be polynomials independently computed by random multilinear formulas sampled using* $\mathtt{SAMPLE}_2$ *over the indicated variable sets. Then for independent* $\alpha,\beta\in_R\mathbb{F}$*,*

$$\Pr[\mathrm{Rank}_{\{\bar{y}\}\{\bar{z}\}}(\alpha\cdot AB+\beta\cdot CD)\le 2]\le\frac{2^{O(m)}}{|\mathbb{F}|}+\frac{1}{2^{\Omega(m)}},$$

*unless*

1) *either* $\{\bar{y}\}=\{\bar{a}\}$ & $\{\bar{z}\}=\{\bar{b}\}$ *or* $\{\bar{y}\}=\{\bar{b}\}$ & $\{\bar{z}\}=\{\bar{a}\}$*, and*
2) *either* $\{\bar{y}\}=\{\bar{c}\}$ & $\{\bar{z}\}=\{\bar{d}\}$ *or* $\{\bar{y}\}=\{\bar{d}\}$ & $\{\bar{z}\}=\{\bar{c}\}$.

Before we sketch a proof of Theorem 5.6, let's see how it is used in the proof of Theorem 2.2. In Step 3 of RECONSTRUCT, we consider the ranks of the partial derivative matrices for $\hat{\Phi}_S|_{\bar{v}',\bar{y}'}$ and $\hat{\Phi}_S|_{\bar{u}',\bar{x}'}$ w.r.t. partitions $\{\bar{v}',\bar{y}'\}$ and $\{\bar{u}',\bar{x}'\}$, respectively. First, note that if $\bar{v}'$ etc are the correct partition of $S$, i.e., in $\Phi_S$, $v_S=\bar{v}'$ etc., then both the above matrices have rank at most 2. We use Theorem 5.6 to show that, w.h.p., the only partition of $S$ (into four parts) that satisfies these two rank conditions is the correct partition. Indeed, by the discussion preceding Theorem 5.6, we can see that $A_S|_{\bar{v}',\bar{y}'}$, $B_S|_{\bar{v}',\bar{y}'}$, $C_S|_{\bar{v}',\bar{y}'}$, and $D_S|_{\bar{v}',\bar{y}'}$ can be viewed as samples from $\mathtt{SAMPLE}_2$ on the variable set $\{\bar{v}',\bar{y}'\}$ (assigning $S\setminus\{\bar{v}'\cup\bar{y}'\}$ to random values). Similarly for $A_S|_{\bar{u}',\bar{x}'}$, etc., on $\{\bar{u}',\bar{x}'\}$. Now, Theorem 5.6 says if $\mathrm{Rank}_{\{\bar{v}'\}\{\bar{y}'\}}(\hat{\Phi}_S|_{\bar{v}',\bar{y}'})\le 2$, then, w.h.p., the variables that $A_S|_{\bar{v}',\bar{y}'}$ etc. depend on must each be either $\bar{v}'$ and $\bar{y}'$. Thus, w.l.o.g., we must have $\bar{v}_S=\bar{v}'$ and $\bar{y}_S=\bar{y}'$. By a similar argument applied to $\hat{\Phi}_S|_{\bar{u}',\bar{x}'}$, we can conclude that $\bar{u}_S=\bar{u}'$ and $\bar{x}_S=\bar{x}'$. Note that since $AB$ and $CD$ are defined on two independent partitions of $X$, it is unlikely that $A$ and $C$ depend on the same set of variables. Furthermore, we can also see that Step 4 associates each $x_i$ with correct block of the seed partition by applying this argument repeatedly for the seed partition augmented with $x_i$. This concludes the proof that Steps 3 and 4 determine the correct partition for $\Phi$.

The following technical lemmas are used in the proof of Theorem 5.6. Their proofs will appear in the full version of the paper. Throughout this paper, LI stands for "Linearly Independent" and LD for "Linearly Dependent."

**Lemma 5.7.** *Let $f$ and $g$ be two multilinear polynomials over an $n$-sized variable set $Y \cup Z$ and field $\mathbb{F}$. Then for any $S \subseteq \mathbb{F}$ and independently chosen $\alpha, \beta \in_R \mathbb{F}$,*

$$\Pr_{\alpha,\beta \in_R S}[\mathrm{Rank}_{YZ}(\alpha.f + \beta.g) > 2] \geq 1 - \frac{2^n}{|S|},$$

*unless $f$ and $g$ have one of the following forms,*

1) $f = f_1(Y)f_2(Z)$ *and* $g = g_1(Y)g_2(Z)$
2) $f = f_1(Y)f_2(Z) + f_3(Y)f_4(Z)$ *($f_1, f_3$ are LI, $f_2, f_4$ are LI) and either* $g = [a.f_1(Y) + b.f_3(Y)]g_2(Z)$ *or* $g = g_1(Y)[a.f_2(Z) + b.f_4(Z)]$
3) $f = f_1(Y)f_2(Z) + f_3(Y)f_4(Z)$ *($f_1, f_3$ are LI, $f_2, f_4$ are LI) and* $g = [a.f_1(Y) + b.f_3(Y)]g_2(Z) + [c.f_1(Y) + d.f_3(Y)]g_4(Z)$ *($g_2, g_4$ are LI and $ad \neq bc$)*
4) $f = f_1(Y)f_2(Z) + f_3(Y)f_4(Z)$ *and* $g = [a.f_1(Y) + b.f_3(Y)]g_2(Z) + g_3(Y)[c.f_2(Z) + d.f_4(Z)]$ *($f_1, f_3, g_3$ are LI, $f_2, f_4, g_4$ are LI and $ac = -bd$)*

*and their analogous cases, where $f_i$'s and $g_i$'s are some multilinear polynomials on their indicated variable sets and $a, b, c, d \in \mathbb{F}$.*

**Lemma 5.8.** *Let $S$ be a set of multilinear monomials over $\{r_1, r_2, \ldots, r_n\}$, where $r_i$'s are independent r.v.'s and each $r_i \in_R \mathbb{F}^*$. Then for every $M \in S$ there exists a set $S_M \subseteq S$ such that*

1) $|S_M| \geq \log |S| - 1$ *and*
2) $S_M \cup \{M\}$ *is a set of independent uniform r.v.'s over* $\mathbb{F}^*$.

**Lemma 5.9** (Irreducibility Lemma)**.** *Let $f_R$ be the polynomial computed by a random multilinear formula over the variables set $X = \{x_1, x_2, \ldots, x_m\}$ and field $\mathbb{F}$ sampled using $\mathtt{SAMPLE}_2$. The probability that there exists a proper partition $\{Y, Z\}$ of $X$ such that $\mathrm{Rank}_{YZ}(f_R) = 1$ is at most $\frac{2^{O(m)}}{|\mathbb{F}|}$.*

**Lemma 5.10.** *Let $\{Y, Z\}$, with $|Y| \leq |Z|$, be a partition of variable set $X = \{x_1, \ldots, x_m\}$ such that both $|Y|$, $|Z|$ are at least $\gamma m$ for some $\gamma > 0$ and $\delta$ be a sufficiently large integer constant . Let $f$ be the polynomial computed by a random multilinear formula sampled using $\mathtt{SAMPLE}_2(X, \mathbb{F})$. Then, with probability at least $1 - \frac{2^{O(m)}}{|\mathbb{F}|} - \frac{1}{2^{\gamma m/18 \log^2 \delta}}$,*

1) *there are at least $\delta$ distinct monomials multilinear in $Z$ variables such that coefficients of these are polynomials in $Y$ each containing at least $\delta$ monomials and*
2) $Rank_{YZ}(f) > 2$.

**Proof sketch for Theorem 5.6:** We first show, using Lemma 5.7, that a random linear combination $\alpha f + \beta g$ has rank $\leq 2$ w.r.t. a partition $(Y, Z)$ of the underlying variable set only under very special conditions. The most natural of these is when $f$ and $g$ are both of rank 1, i.e., $f(Y, Z) = f_1(Y) \cdot f_2(Z)$ and $g(Y, Z) = g_1(Y) \cdot g_2(Z)$. The other (degenerate) conditions arise when at least one of $f$ or $g$ has rank 2 and can be categorized into a small number of special cases. The second part of the proof is to show that when $f = AB$ and $g = CD$ and $A$, $B$, $C$, and $D$ are samples from $\mathtt{SAMPLE}_2$, the degenerate conditions are satisfied with very low probability. This will imply $AB$ and $CD$ must satisfy the natural condition and hence their supports must satisfy (1) and (2). For the second part, we use two main arguments about a random formula according to $\mathtt{SAMPLE}_2$ on $m$ variables: (i) it must have rank at least two, w.h.p., for any nontrivial partition of its variables (Irreducibility Lemma, Lemma 5.9) and (ii) for any partition $(Y, Z)$ with $|Y|, |Z| \geq \Omega(m)$, it must contain many monomials in $Z$ variables whose coefficients (which are polynomials in $Y$) must also contain many monomials in $Y$ variables (Lemma 5.10). By (i), we only need to consider when, say $f$, is of $\mathrm{Rank}$ 2 w.r.t. some partition (not necessarily $(Y, Z)$). This, combined with any of the degeneracy conditions, implies that the number of statistically independent monomials in $Y$ variables in the coefficient of a suitably chosen $Z$-monomial in $g$ must be small (since they are determined by linear combinations given by the degeneracy conditions of a small number of coefficients of $f$'s factors). But this contradicts (ii) since (by Lemma 5.8) there must be many independent monomials in $Y$ variables. ■

## 6. Conclusions and Open Problems

In this paper, we proved that multilinear arithmetic formulas can be efficiently reconstructed when the formula is chosen randomly according to some natural distributions. This is the strongest model of arithmetic complexity for which such a reconstruction algorithm is currently known even if the efficiency is in a distributional sense. A slight variant of the worst-case reconstruction of multilinear formulas is known to be intractable. Our result gives supporting evidence to the general theme that mathematical techniques useful in proving lower bounds against a model of computation should also enable learning algorithms for that model. This general view and the technical and intuitive connections among the areas of lower bounds in arithmetic complexity, polynomial identity testing, and reconstruction motivate several open questions for future research. Some examples include worst-case reconstruction of depth-3 constant top fan-in arithmetic formulas (over arbitrary fields), constant top fan-in depth-4 multilinear formulas, Arithmetic Branching Programs (non-commutative as well as suitably restricted commutative models), and so on. Worst-case reconstruction in many nontrivial models is either very hard or provably intractable. However, requiring efficiency only in a distributional sense appears to bring the reconstruction problem in several powerful models

including, for example, general arithmetic formulas, within reach and thus open up a direction of positive results in this area. This paper may be viewed as a first result in that direction.

## ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their thoughtful comments and suggestions which have significantly helped us in improving the paper. We also thank Amir Shpilka for pointing out an improvement in the field size.

## REFERENCES

[1] J. Håstad, "Tensor rank is NP-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[2] E. Kaltofen, "Factorization of polynomials given by straight-line programs," in *Randomness and Computation*. JAI Press, 1989, pp. 375–412.

[3] Z. S. Karnin and A. Shpilka, "Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in," in *IEEE Conference on Computational Complexity*, 2009, pp. 274–285.

[4] A. Klivans and D. A. Spielman, "Randomness efficient identity testing of multivariate polynomials," in *STOC*, 2001, pp. 216–223.

[5] A. R. Klivans and A. Shpilka, "Learning restricted models of arithmetic circuits," *Theory of Computing*, vol. 2, no. 1, pp. 185–206, 2006.

[6] N. Nisan, "Lower bounds for non-commutative computation," in *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, ser. STOC '91. New York, NY, USA: ACM, 1991, pp. 410–418.

[7] R. Raz, "Multi-linear formulas for permanent and determinant are of super-polynomial size," *Journal of the Association for Computing Machinery*, vol. 56, no. 2, 2009.

[8] R. Raz and A. Yehudayoff, "Lower bounds and separations for constant depth multilinear circuits," *Computational Complexity*, vol. 18, no. 2, pp. 171–207, 2009.

[9] A. Shpilka, "Interpolation of depth-3 arithmetic circuits with two multiplication gates," *SIAM J. Comput.*, vol. 38, no. 6, pp. 2130–2161, 2009.

[10] A. Shpilka and I. Volkovich, "On the relation between polynomial identity testing and finding variable disjoint factors," in *ICALP (1)*, 2010, pp. 408–419.

[11] A. Shpilka and A. Yehudayoff, "Arithmetic circuits: A survey of recent results and open questions," *Foundations and Trends in Theoretical Computer Science*, vol. 5, no. 3-4, pp. 207–388, 2010.

## APPENDIX

### HANDLING + GATES WITH FAN-IN $k$

In the previous sections we described how to reconstruct a random binary multilinear formula sampled using $\texttt{SAMPLE}(X, \mathbb{F})$. We now show that our algorithm, with a few minor modifications, can also reconstruct a random $k$-ary multilinear formula (where $k = O(1)$) i.e. a formula sampled using the method $\texttt{SAMPLE}(X, \mathbb{F})$ where, in Case 3, instead of recursively constructing 2 children, it constructs $k$ children. Hence the resulting formula would have $+$ gates of fanin $k$. As before, it can be shown that w.h.p. the polynomial computed at a $+$ gate in such a formula would be irreducible. Hence if the root was a $\times$ gate then we can gain blackbox access to the children using Kaltofen's algorithm and/or the algorithm of Shpilka and Volkovich [10]. The overall strategy of the algorithm is as in the outline given in section 3. The interesting case is where the root of the formula is a $+$ gate so that the polynomial computed is of the form

$$f = \sum_{i=1}^{k} A_i(\bar{x}_i, \ldots, \bar{x}_{i+k-1}) \cdot B_i(X \setminus \{\bar{x}_i, \ldots, \bar{x}_{i+k-1}\})$$

where $\{\{\bar{x}_i\}\}_{i\in[2k]}$ is a partition of the variable set $X$ such that $\bar{x}_i$'s are of roughly the same size and $A_i$'s, $B_i$'s are random $k$-ary multilinear formulas on their respective variable sets. With high probability, the $A_i$'s, $B_i$'s are of the same degree say $d$. As before, we obtain blackbox to the grandchildren in two steps. In the first step, we determine the partition of $\{\{\bar{x}_i\}\}_{i\in[2k]}$ of the variable set $X$. In the second step, we indicate how to obtain the evaluation of $A_i$ (respectively $B_i$) at any given point.

**Determining the partition.** As in the case of fanin two, the idea is that for the correct partition of $X$ the rank of a certain matrix will be very small whereas it will be large for every incorrect partition. As before, the problem can be reduced to the case where $|X|$ is relatively small (roughly $\Theta(k \log n)$) by first determining a seed partition and then extending it by introducing one variable at a time. With this lifting idea at hand, it suffices to determine whether a given partition is "the correct one" (w.h.p over the random choice of the formula, it will hold true that there is a unique correct partition). The idea is that if

$$X = \biguplus_{i\in[2k]} \{\bar{x}_i\}$$

is the correct partition then for any $\{\bar{x}_{i-1}\}$, $\{\bar{x}_i\}$ we have

$$\hat{f} = \hat{A}_i(\bar{x}_i)\hat{B}_i(\bar{x}_{i-1}) + \sum_{j\neq i} \alpha_j \hat{A}_j(\bar{x}_{i-1}, \bar{x}_i) + \beta_j \hat{B}_j(\bar{x}_{i-1}, \bar{x}_i),$$

where for any polynomial $g$, $\hat{g}$ denotes the polynomial obtained by setting all the variables from $X \setminus \{\bar{x}_{i-1}, \bar{x}_i\}$ to random values. This means that for any $j < d$, the polynomial $\hat{f}^{[2d-j]}$, the homogeneous part of degree $(2d-j)$ of $\hat{f}$, will be of rank $(j+1)$ with respect to the partition $\bar{x}_{i-1} \uplus \bar{x}_i$. On the other hand, $\hat{f}^{[2d-j]}$ will not satisfy the stated rank bound for any incorrect partition (w.h.p. over the random choice of the formula).

**Obtaining blackbox access to the grandchildren.** We now indicate how one can obtain blackbox access to the $A_i$'s and $B_i$'s, given blackbox access to $f$ and the partition $\{\{\bar{x}_i\}\}_{i\in[2k]}$. In order to determine say $A_1(\bar{\beta}_1, \ldots, \bar{\beta}_k)$, one substitutes all the variables except $\bar{x}_1$ and $\bar{x}_{2k}$ to appropriate

values and looks at the homogeneous components of the resulting polynomial. Specifically, make the substitution

$$\bar{x}_j := \begin{cases} \bar{\beta}_j & \text{for } j \in [2..k], \\ \bar{\alpha}_j \in_R \mathbb{F}^{|\bar{x}_j|} & \text{for } j \in [(k+1)..(2k-1)] \end{cases}$$

and let the resulting polynomial be $\hat{f}(\bar{x}_1, \bar{x}_{2k})$. It turns out then that $\hat{f}^{[2d]}$ equals $\hat{A}_1^{[d]}(\bar{x}_1) \cdot \hat{B}_1^{[d]}(\bar{x}_{2k})$. Factoring this polynomial gives us access to $A_1^{[d]}(\bar{x}_1, \bar{\beta}_2, \ldots, \bar{\beta}_k)$ and therefore to $A_1^{[d]}(\bar{\beta}_1, \ldots, \bar{\beta}_k)$ as well. As in the case of fanin two, this idea can be extended suitably to obtain $A_1(\bar{\beta}_1, \ldots, \bar{\beta}_k)$. This completes our brief description for extending the reconstruction result to formulas with higher fanin.