Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 29 (2018)

Average-case linear matrix factorization and reconstruction of low width Algebraic Branching Programs

Neeraj Kayal
Microsoft Research India
neeraka@microsoft.com

Vineet Nair
Indian Institute of Science
vineet@iisc.ac.in

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

November 12, 2018

Abstract

A matrix X is called a *linear matrix* if its entries are affine forms, i.e. degree one polynomials in n variables. What is a minimal-sized representation of a given matrix F as a product of linear matrices? Finding such a minimal representation is closely related to finding an optimal way to compute a given polynomial via an algebraic branching program. Here we devise an efficient algorithm for an average-case version of this problem. Specifically, given $w, d, n \in \mathbb{N}$ and blackbox access to the w^2 entries of a matrix product $F = X_1 \cdots X_d$, where each X_i is a $w \times w$ linear matrix over a given finite field $\mathbb{F}_q$, we wish to recover a factorization $F = Y_1 \cdots Y_{d'}$, where every Y_i is also a linear matrix over $\mathbb{F}_q$ (or a small extension of $\mathbb{F}_q$). We show that when the input F is sampled from a distribution defined by choosing random linear matrices $X_1, \ldots, X_d$ over $\mathbb{F}_q$ independently and taking their product and $n \geq 4w^2$ and $\mathrm{char}(\mathbb{F}_q) = (ndw)^{\Omega(1)}$ then an equivalent factorization $F = Y_1 \cdots Y_d$ can be recovered in (randomized) time $(wdn \log q)^{O(1)}$. In fact, we give a (worst-case) polynomial time randomized algorithm to factor any non-degenerate or *pure* matrix product (a notion we define in the paper) into linear matrices; a matrix product $F = X_1 \cdots X_d$ is pure with high probability when the X_i's are chosen independently at random. We also show that in this situation, if we are instead given a single entry of F rather than its w^2 correlated entries then the recovery can be done in (randomized) time $(d^{w^3} n \log q)^{O(1)}$.

ISSN 1433-8092

Contents

1 Introduction

1.1 Motivation and an overview

Polynomial matrix factorization. In this paper, we are interested in factorization of a polynomial matrix (that is a matrix with multivariate polynomial entries) into linear matrices, if such a factorization exists. A linear matrix has affine forms as entries. We call this problem *linear matrix factorization*. It is a natural generalization of the problem of factoring a multivariate polynomial into linear factors for which there is a known efficient randomized algorithm [KT90]. Motivated by applications in control theory, polynomial matrix factorization has been studied in the literature under various restrictions on input and output matrices (see [LMW17] and the references therein). To our knowledge, these restrictions are quite different from the requirement of outputting linear matrix factors of an input polynomial matrix. Our primary motivation for studying this problem stems from the problem of learning or *reconstruction* of algebraic branching programs (ABPs) – a powerful subclass of arithmetic circuits capturing determinant and iterated matrix multiplication computations (see Definition 1.1). The linear matrix factorization problem can be equivalently thought of as the problem of reconstructing ABPs.

Hardness of reconstruction. Circuit reconstruction is a notable problem in algebraic complexity theory alongside proving lower bounds and polynomial identity testing. Reconstruction of a circuit class $\mathcal{C}$ is the following problem: Given black-box access (i.e. membership query access) to a polynomial function f that is computed by an arithmetic circuit of size s from $\mathcal{C}$, output a circuit (preferably from $\mathcal{C}$) of size not much larger than s (ideally, a polynomial or quasi-polynomial function of s) computing f. Reconstruction of general arithmetic circuits is believed to be an inherently hard computational problem. An algorithm for reconstruction naturally gives an approximation of the size of the smallest circuit computing f. Thus, hardness of reconstruction is related to hardness of approximation of the minimum circuit size problem. The hardness of the minimum circuit size problem for Boolean circuits, known as MCSP, is an intensely studied problem in the literature. In the MCSP problem, we are given the truth-table of a Boolean function and a parameter s as input and the task is to determine if the function can be computed by a Boolean circuit of size at most s. Allender and Hirahara [AH17] showed that approximating the minimum circuit size within a factor of $N^{1-o(1)}$ is NP-intermediate, assuming the existence of one-way functions, where N is the size of the input truth-table [1]. Drawing analogy between the Boolean and the arithmetic worlds, we expect the reconstruction problem to be hard even if the polynomial function f is given verbosely as a list of coefficients, and it only gets harder if f is given succinctly as a circuit or a black-box holding the circuit. If we insist on a very small approximation factor and on computing an output circuit that belongs to the same class $\mathcal{C}$ as the input circuit (as in proper learning), then the problem becomes NP-hard even for simple circuit classes like set-multilinear depth three circuits and depth three powering circuits [BIJL18, SWZ17, Shi16, Hås90].

It is also known that efficient reconstruction implies lower bounds. It was shown in [FK09] that a randomized polynomial time reconstruction algorithm for an arithmetic circuit class $\mathcal{C}$ implies the existence of a function in BPEXP that does not have polynomial size circuits from $\mathcal{C}$. Also,

[1] Another related result is the hardness of approximating minimum size DNF. Umans [Uma99] showed that there is no polynomial time algorithm to compute $n^{1-\epsilon}$ factor approximation of the minimum DNF size of an input DNF of size n, for every constant $\epsilon \in (0,1)$, assuming $\Sigma_2^p \not\subseteq \mathsf{DTIME}(n^{O(\log n)})$.

Volkovich [Vol16] showed that a deterministic polynomial time reconstruction algorithm for $\mathcal{C}$ implies the existence of an explicit polynomial h such that any circuit from $\mathcal{C}$ computing h has exponential size. In hindsight, it is no wonder that research on reconstruction has focused on interesting restricted circuit classes for which non-trivial lower bounds are known (see the survey [SY10] and the references in [KNST17]). Does lower bound imply reconstruction? Even if we believe in the existence of explicit polynomials with high circuit complexity, we may not hope to get such an implication unconditionally as reconstruction seems to be an inherently hard problem. However, the answer is less clear for lower bound proofs with additional features such as "natural proofs". Taking inspiration from [RR97], the notion of algebraic natural proofs is defined in [FSV17, GKSS17] to explore the limitations of existing techniques in proving $\mathsf{VP} \neq \mathsf{VNP}$ [2].

Does natural lower bound proofs imply reconstruction? The intuitive reason for expecting a somewhat positive answer rests on the high level view that a natural lower bound proof (in the sense of [RR97]) is able to "efficiently" check some property of polynomials computed by a circuit class, and the same property is potentially useful in designing reconstruction algorithms for the class. Indeed, for Boolean circuits, an interesting result [CIKK16] showed that the natural lower bound proof framework [RR97] for $\mathsf{AC}^0[p]$ circuits can be used to give a quasi-polynomial time PAC learning (with membership queries) algorithm for the same class. The result generalizes to any circuit class $\mathcal{C}$ containing $\mathsf{AC}^0[p]$ for some prime p, the "usefulness" parameter of a natural proof for $\mathcal{C}$ determines the efficiency of such a PAC learning algorithm for $\mathcal{C}$. This generic result is preceded by evidences that hinted at such a connection, like the learning algorithms for AC^0 circuits [LMN93] and AC^0 circuits with few majority gates [JKS02] [3]. Analogous to Boolean circuits, does an algebraically natural lower bound proof (in the sense of [FSV17, GKSS17]) for an arithmetic circuit class imply efficient reconstruction for the same class? Unlike PAC learning, in the algebraic setting we need to reconstruct a circuit that computes the input polynomial exactly instead of approximately, as two distinct polynomial functions differ at far too many points. If we insist on such exact learning in the Boolean setting (which is closely related to the compression problem for Boolean functions) then the best known output circuit size for AC^0 and $\mathsf{AC}^0[p]$ functions is exponential in the number of variables [CKK+15, Sri15, CIKK16]. In the absence of a generic connection (analogous to [CIKK16]) in the algebraic setting, we could gather more evidences for or against such a connection by focusing on restricted classes for which natural lower bound proofs are known.

There are a few favorable results, like the reconstruction algorithms for read-once oblivious ABPs, set-multilinear ABPs and non-commutative ABPs [FS13, KS06]. However, there are many other interesting arithmetic circuit classes for which we know of strong lower bounds (that are also algebraically natural), but not efficient reconstruction algorithms. Instances of such classes are homogeneous depth three circuits [NW97, KST16], homogeneous depth four circuits [KLSS17, KS17], constant depth multilinear circuits [RY09], multilinear formulas [Raz09], regular formulas [KSS14], and a few other classes [KS16a, KS16b]. Even for a more powerful model like homogeneous ABPs, it makes sense to ask – can we reconstruct sub-linear width homogeneous ABPs

[2] In another interesting work [EGdOW18], limitations of rank based lower bound methods have been shown *unconditionally* towards achieving strong lower bounds for set-multilinear depth-3 circuits and diagonal depth-3 circuits.

[3] For circuit classes whose known lower bound proofs do not fit in the natural proof framework, the situation is less clear. Examples of such classes are ACC^0 [Wil14] and monotone circuits [Raz85]. A hardness result for polynomial-time learning of monotone circuits is known assuming the existence of one-way functions [DLM+08].

efficiently? A linear width lower bound for homogeneous ABPs is known [Kum17], and this lower bound proof is also natural. Unfortunately, there is some amount of evidence that indicate that the problem remains hard *in the worst-case* even for models for which natural lower bound proofs are known. For example, a polynomial time reconstruction algorithm for homogeneous depth three circuits implies a sub-exponential time reconstruction algorithm for general circuits due to the depth reduction to depth three results [GKKS16, Tav13, Koi12, AV08]; it would also give a super-polynomial lower bound for depth three circuits (via the learning to lower bound connection in [FK09]) – proving such a lower bound is a long-standing open problem. Similarly, a polynomial time reconstruction algorithm for constant width (in fact, width-3) homogeneous ABPs implies a polynomial time reconstruction algorithm for arithmetic formulas due to the reduction from formulas to width-3 ABPs in [BC92], and this in turn would give a super-polynomial lower bound for formulas (by [FK09]) – proving such a lower bound is another challenging open problem in algebraic complexity.

Average-case reconstruction. For any one of the above-mentioned models lacking efficient worst-case reconstruction, we can attempt to make progress by asking a slightly weaker question: Can we do efficient reconstruction for *almost all* polynomials computed by the model? This amounts to studying the reconstruction problem under some distributional assumptions on the polynomials computed by the model. Such types of reconstruction are called *average-case reconstruction*[4]. Often than not, an average-case algorithm in fact gives a worst-case algorithm for inputs satisfying some natural/easy-to-state *non-degeneracy* condition (like the 'pure matrix product' condition stated in Section 1.3), which is almost surely satisfied by a random input chosen according to any reasonable distribution. We feel that it is worth knowing these non-degeneracy conditions that make worst-case reconstruction tractable for some of models mentioned above. But, even average-case reconstruction (i.e., reconstruction under non-degeneracy conditions) turns out to be quite non-trivial for these models under some natural distributions.

In [GKL11] and [GKQ13], average-case reconstruction algorithms were given for multilinear formulas and fanin-2 regular formulas respectively under intuitive input distributions. Algebraic branching programs being more powerful than formulas, the problem of efficient average-case reconstruction of ABPs was posed in our earlier work [KNST17] under a natural distribution (see Definition 1.2). A polynomial-time samplable (in short, P-samplable) input distribution is additionally interesting if it is also relevant in the context of lower bound proofs – we elaborate on this point next.

For the discussion ahead, we denote a n-variate, degree-d polynomial as a (n, d)-polynomial; a random (n, d)-polynomial denotes a (n, d)-polynomial with coefficients chosen independently and uniformly at random from $\mathbb{F}$. Assume that $\mathbb{F}$ is a sufficiently large finite field $\mathbb{F}_q$, although this requirement is not binding for the most part of our arguments.

Choosing an input distribution. A lower bound proof for a class $\mathcal{C}$ shows that an explicit (n, d)-polynomial[5] is not computable by size-s circuits from $\mathcal{C}$, for some $s > \max(n, d)$. Such a proof

[4]Similar average-case relaxations of learning problems have been studied in the Boolean setting, particularly for DNFs [LSW06, JLSW08].

[5]Typically, the explicit polynomial has degree $d \leq n$, say $d = \sqrt{n}$ or $d = \Theta(n)$ (as in determinant/permanent [Raz09,

demonstrates a weakness of the set of (n, d)-polynomials computable by size-s circuits from $\mathcal{C}$. In order to exploit the weakness of this set in an average-case reconstruction problem for $\mathcal{C}$, we should ideally define an input distribution that is supported on (n, d)-polynomials computable by size-s circuits in $\mathcal{C}$, where $s > \max(n, d)$; moreover, the distribution should be P-samplable. For many circuit classes, defining such a distribution is a bit tricky as some of the natural P-samplable distributions tend to be primarily supported on (n, d)-polynomials where d or n is closely attached to the size s of the circuits produced by these distributions (as in [GKL11, GKQ13, KNST17])[6], thereby restricting s from being much larger than $\max(n, d)$. However, for some classes, like homogeneous ABPs and homogeneous depth three circuits, these requirements from an input distribution (especially, allowing $s \gg \max(n, d)$) can be mitigated easily. We study the former model here. The latter is handled in [KS18].

Choosing a distribution on homogeneous ABPs and the role of width. A well-known ABP homogenization argument [Nis91] implies the following: If a (n, d)-polynomial is computable by an ABP $\mathtt{A}$ of size s then it is also computable by an ABP $\mathtt{B}$ of width $w \leq s$ and length d. If $\mathtt{A}$ is a homogeneous ABP of size s then $\mathtt{B}$ is also a homogeneous ABP of size s. From the perspective of lower bound for homogeneous ABPs, the distribution given in Definition 1.2 for average-case ABP reconstruction (Problem 2) is quite appropriate to study as it produces (n, d)-polynomials computable by ABPs of size $s \approx wd$ that can potentially be much larger than $\max(n, d)$ with growing width w. In [Kum17], a quadratic lower bound for homogeneous ABP is given by essentially showing a linear width lower bound: Any (w, d, n)-ABP computing the power symmetric polynomial $\sum_{i=1}^{n} x_i^d$ must satisfy $w \geq \frac{n}{2}$, implying that the size of such an ABP is $s \approx wd = \Omega(nd)$. Choosing $d = \Theta(n)$ yields the quadratic bound. This means, the homogeneous ABP reconstruction problem is interesting for $w < \frac{n}{2}$. However, as mentioned before, we cannot hope to do an efficient worst-case reconstruction for homogeneous ABP of even constant width in the absence of a super-polynomial lower bound for formulas [BC92, FK09]. But, can we do average-case reconstruction for $w < \frac{n}{2}$? We answer this question partially. Before stating our contribution, let us note that average-case reconstruction beyond $w = O(n)$ seems difficult at the moment given that no $\Omega(n^{1+\epsilon})$ lower bound on w is known, for any constant $\epsilon > 0$ [7]. Such a lower bound would in turn imply a $\Omega(n^{1+\epsilon})$ lower bound on the size of general ABPs which, if shown, would be an excellent progress in the area. Thus, pushing our understanding of the ABP reconstruction complexity (in the average-case) for w up to $\Theta(n)$ seems like a worthwhile endeavor to us.

Our contribution. We make progress in this direction by giving a nontrivial average-case reconstruction algorithm for $w \leq \sqrt{n}/2$, *irrespective of d* (Theorem 2). The algorithm outputs a (w, d, n)-ABP (with high probability) for the input polynomial chosen according to the distribution in Definition 1.2. A trivial brute-force algorithm to reconstruct a (w, d, n)-ABP over $\mathbb{F}_q$ takes time $q^{\Theta(w^2 dn)}$. By 'nontrivial' reconstruction, we mean an algorithm that takes time subexponen-

RY09, GKKS14] or the Nisan-Wigderson design polynomial [KSS14] or the elementary/power symmetric polynomials [NW97, SW01, Kum17] or a variant of the design polynomial [KST16]).

[6] The result in [GKQ13] can be viewed as an average-case reconstruction algorithm for size-s, fanin-2 regular formulas computing (n, d)-polynomials, where $s = \Theta(nd^2)$. In comparison, a $n^{\Omega(\log d)}$ size lower bound is known for regular formulas [KSS14].

[7] An average-case reconstruction for width $w = n^{1+\epsilon}$ homogeneous ABP would necessarily show that polynomials computed by such ABPs do not form a pseudo-random family which in turn opens up the possibility of having a natural lower bound proof for this class of ABPs.

tial in the quantity w^2dn. Note that we can interpolate a polynomial computed by a (w,d,n)-ABP in $(d^n \log q)^{O(1)}$ time, but knowing the coefficients of the polynomial does not give us any immediate information about the (w,d,n)-ABP that computes it – this point is related to the hardness of the MCSP problem and reconstruction under verbose representation of the input polynomial mentioned before. Hence, if we want a (w,d,n)-ABP representation for the input polynomial then even a $(d^n \log q)^{O(1)}$ time reconstruction algorithm is nontrivial as d^n is subexponential in w^2dn for any $d = n^{\Omega(1)}$. The complexity of our algorithm is $(d^{w^3} n \log q)^{O(1)}$ which is also subexponential in w^2dn for $w \leq \sqrt{n}/2$. For instance, if $m = w^2dn$, $w = \sqrt{n}/2$ and $d = \Theta(n)$ then the trivial complexity is $\exp(m)$ and our algorithm's time complexity is $\exp(\sqrt{m})$. For constant width homogeneous ABPs, our algorithm runs in polynomial time; if we can achieve the same complexity for worst-case reconstruction (instead of average-case) then that would imply a super-polynomial lower bound for arithmetic formulas! We also note that the main step (linear matrix factorization, Theorem 1) of our algorithm has only $(wdn \log q)^{O(1)}$ time complexity and is therefore polynomial time. The exponential dependence on w^3 comes from a step in our algorithm that solves polynomial equations. It is quite possible that this expensive solvability step can be circumvented and the overall complexity of the algorithm reduced to polynomial time – we leave this as an open question in Section 1.5. Along the way, we give an efficient equivalence test for the determinant *over finite fields* (Theorem 3) which is independently interesting.

Theorem 1 can be interpreted as a worst-case randomized polynomial time algorithm to factor a *pure* matrix product. We define the notion of purity after stating the theorem in Section 1.3. It is easy to show that the input distribution churns out a pure matrix product with high probability. Similarly, Theorem 2 can be interpreted as a worst-case randomized algorithm to reconstruct a *non-degenerate* ABP in $(d^{w^3} n \log q)^{O(1)}$ time. The non-degeneracy conditions are stated in Section 4.3.

Comparison with our previous work. In [KNST17], we gave a reconstruction algorithm for $w \leq \sqrt{\frac{n}{d}}$. Observe that, under this width constraint, the size $s \approx wd$ of an ABP is upper bounded by $\max(n,d)$. Whereas, in this paper we give a reconstruction algorithm for $w \leq \sqrt{n}/2$ (independent of d), and hence the size of the ABPs here can be $s = \Theta(\sqrt{n}d)$. To highlight this improvement, if we set $d = \Theta(n)$ (as in several lower bound results [Kum17,KST16,SW01,NW97]) then the width constraint in [KNST17] reduces to $w = O(1)$ and size becomes $\Theta(n)$, whereas the size of the ABPs in this work is $\Theta(n^{1.5})$ which is significantly closer to the best known $\Omega(n^2)$ lower bound for homogeneous ABPs. Also, it is because of the independence of d on the width constraint that we could infer that the same time complexity for worst-case reconstruction of constant width homogeneous ABP would imply a super-polynomial formula lower bound, as the process of homogenizing a non-homogeneous ABP to a homogeneous ABP (described in Section 1.4.2) bloats up the degree d. These factors underscore the importance of getting rid of the dependence on d from the width constraint. On the flip side though the running time of the algorithm in [KNST17] is $(wdn \log q)^{O(1)}$, whereas the algorithm here has time complexity $(d^{w^3} n \log q)^{O(1)}$.

Our approach. Our proof approach is quite different from that of [KNST17]. In [KNST17], the Lie algebra of the iterated matrix multiplication polynomial is analyzed to establish a connection between the layer spaces of a full-rank ABP and the irreducible invariant subspaces of the Lie algebra of the polynomial computed by the ABP. This in turn helped reduce the problem to reconstruction of a set-multilinear ABP. We cannot hope to do a similar reduction here as the number of variables is much fewer (and independent of d). Instead, our proof hinges on three steps whose proofs of correctness are somewhat technical:

Step 1. Showing the *uniqueness of the corner spaces* when $w \leq \sqrt{n}/2$, and finding these spaces. This step involves solving polynomial equations. The corner spaces are the two $\mathbb{F}$-linear spaces spanned by the affine forms in the first matrix and the affine forms in the last matrix of the ABP.

Step 2. (Main step) Recovering the intermediate matrices modulo the corner spaces and *rearranging them in the correct order*. This is the linear matrix factorization step (Theorem 1) and is the main part of our algorithm.

Step 3. Completing the affine forms in the intermediate matrices by showing *linear independence of the so-called minors* of a random ABP. See Section 4.2 for the definition of a minor.

The determinant equivalence test is used to recover the intermediate matrices (modulo the corner spaces) in Step 2. More details on the three steps are given in Section 1.4. We think that these steps give us some crucial insights into the structure of a random ABP which may find applications in other similar problems and in resolving some of the questions stated in Section 1.5.

1.2 The problems

We study two related problems in this work, *average-case matrix factorization* and *average-case ABP reconstruction*. The average-case matrix factorization problem aids us in making progress on the average-case ABP reconstruction problem, but the former is also independently interesting. The definition of an ABP given below is quite standard and similar to the one stated in [KNST17].

Definition 1.1 (*Algebraic branching program*)**.** An *algebraic branching program* (ABP) of width w and length d is a product expression $X_1 \cdot X_2 \ldots X_d$, where X_1, X_d are row, column linear matrices over $\mathbb{F}$ of length w respectively, and X_i is a $w \times w$ linear matrix over $\mathbb{F}$ for $i \in [2, d-1]$. The polynomial computed by the ABP is the entry of the 1×1 matrix obtained from the product $\prod_{i=1}^{d} X_i$. An ABP of width w, length d, and in n variables will be called a (w, d, n)-ABP over $\mathbb{F}$. An ABP $X_1 \cdot X_2 \ldots X_d$ is homogeneous, if every entry in every partial product $X_1 \cdot X_2 \ldots X_i$ is a homogeneous polynomial.

Remarks:

(a) A more general way to define an ABP is to consider matrices of varying dimensions, i.e. the i-th matrix has dimension $w_i \times w_{i+1}$, and $w_1 = w_{d+1} = 1$. Size of such an ABP is the quantity $\sum_{i=1}^{d+1} w_i$. Equivalently, an ABP can be defined as a layered directed acyclic graph, in which case size is the number of nodes in the graph.

(b) The *iterated matrix multiplication* polynomial ($\mathsf{IMM}_{w,d}$) is computed by a (w, d, n)-ABP where each entry in X_i is a distinct variable, for all $i \in [d]$, and hence $n = w^2(d-2) + 2w$.

(c) A polynomial computed by a (w, d, n)-ABP can be viewed as an entry of a product of d, $w \times w$ linear matrices $X_1, X_2, \ldots, X_d$. The $w \times w$ matrix $F = X_1 \cdot X_2 \ldots X_d$ is then called a (w, d, n)-*matrix product*. We note that in the matrix product formulation X_1, X_d are $w \times w$ linear matrices, while in the ABP formulation X_1, X_d are row and column linear matrices of length w respectively; hopefully, the context will make the dimensions of these matrices clear.

To study average-case reconstruction for ABP, [KNST17] defined a natural distribution on polynomials computed by ABPs. The distribution is expressed by a *random* (w, d, n)-ABP. In the following definition a random linear matrix is a linear matrix where the coefficients of the affine forms are chosen independently and uniformly at random from $\mathbb{F}$.

Definition 1.2 (*Random ABP and matrix product*)**.** A *random* (w, d, n)-ABP over $\mathbb{F}$ is a (w, d, n)-ABP $X_1 \cdot X_2 \ldots X_d$ over $\mathbb{F}$, where X_i is a random linear matrix chosen independently for every $i \in [d]$. Similarly, a *random* (w, d, n)-matrix product over $\mathbb{F}$ is a (w, d, n)-matrix product $F = X_1 \cdot X_2 \ldots X_d$ over $\mathbb{F}$, where X_i is a random linear matrix chosen independently for every $i \in [d]$.

Having defined the distributions, the two average-case problems can be posed as follows.

Problem 1 (*Average-case matrix factorization*)**.** Design an algorithm which when given $w, d, n \in \mathbb{N}$, and blackbox access to w^2, (n, d)-polynomials $\{f_{st}\}_{s,t\in[w]}$ that constitute the entries of a random (w, d, n)-matrix product F over $\mathbb{F}_q$, outputs d, $w \times w$ linear matrices $Y_1, \ldots, Y_d$ over $\mathbb{F}_q$ (or a small extension of $\mathbb{F}_q$) such that $F = Y_1 \cdot Y_2 \ldots Y_d$, with high probability. The desired running time of the algorithm is $(wdn \log q)^{O(1)}$.

Problem 2 (*Average-case ABP reconstruction*)**.** Design an algorithm which when given $w, d, n \in \mathbb{N}$, and blackbox access to a (n, d)-polynomial f computed by a random (w, d, n)-ABP over $\mathbb{F}_q$, outputs a (w, d, n)-ABP over $\mathbb{F}_q$ (or a small extension of $\mathbb{F}_q$) computing f, with high probability. The desired running time of the algorithm is $(wdn \log q)^{O(1)}$.

Remark: For both problems, the success probability is taken over the input distribution and the random bits used by the algorithm, if it is randomized. In Problem 1 we have blackbox access to w^2 polynomials constituting the entries of a matrix, whereas in Problem 2 we have blackbox access to a *single* polynomial. In this sense, Problem 1 is supposedly easier than Problem 2. Still, Problem 1 is of independent interest because if the coefficients of the affine forms are chosen adversarially (instead of randomly) in $X_1, X_2, \ldots, X_d$ then even for $w = 3$ the problem becomes as hard as formula reconstruction [BC92].

1.3 Our results

Throughout this article, $\mathbb{F}$ will denote $\mathbb{F}_q$ with $\mathrm{char}(\mathbb{F}) \geq (wdn)^7$, and $\mathbb{L}$ the extension field $\mathbb{F}_{q^w}$. ($\mathbb{L}$ can constructed from a basis of $\mathbb{F}_q$ using a randomized algorithm running in $(w \log q)^{O(1)}$ time [vzGG03].) Also, we will assume $d \geq 5$ for technical reasons. Theorem 1 solves Problem 1 for $n \geq 2w^2$.

Theorem 1 (Average-case matrix factorization)**.** *For $n \geq 2w^2$, there is a randomized algorithm that takes as input blackbox access to w^2, (n, d)-polynomials $\{f_{st}\}_{s,t\in[w]}$ that constitute the entries of a random (w, d, n)-matrix product $F = X_1 \cdot X_2 \ldots X_d$ over $\mathbb{F}$, and with probability $1 - (wdn)^{-\Omega(1)}$ returns $w \times w$ linear matrices $Y_1, Y_2, \ldots, Y_d$ over $\mathbb{L}$ satisfying $F = \prod_{i=1}^{d} Y_i$. The algorithm runs in $(wdn \log q)^{O(1)}$ time and queries the blackbox at points in $\mathbb{L}^n$.*

Remarks:

- The constraint on $\mathrm{char}(\mathbb{F})$ is a bit arbitrary, the results in this paper hold as long as $|\mathbb{F}|$ and $\mathrm{char}(\mathbb{F})$ are sufficiently large polynomial functions in w, d and n.

- *Pure matrix product:* A (w,d,n)-matrix product $X_1 \cdot X_2 \ldots X_d$ over $\mathbb{F}$ is *pure* if it satisfies the following properties:

    1. For every $i \in [d]$, X_i is full-rank i.e., the affine forms in X_i are $\mathbb{F}$-linearly independent.
    2. For every $i,j \in [d]$ and $i \neq j$, $\det(X_i)$ and $\det(X_j)$ are coprime. Here $\det(X_i)$ is the determinant of X_i.
    3. For every $i,j \in [d]$ and $i < j$, the w^2 polynomial entries of the partial product $X_{i+1} \cdots X_j$ are $\mathbb{F}$-linearly independent modulo the affine forms in the first row and column of X_i. [8]

    It can be easily shown (using Claim 2.3 and Claim 2.4) that a random (w,d,n)-matrix product is a pure matrix product (in short, a pure product) with high probability, for $n \geq 2w^2$. Theorem 1 actually gives a polynomial time linear matrix factorization algorithm for a pure product.

- *Uniqueness of factorization:* The proof of the theorem also shows that linear matrix factorization of a pure product is unique in the following sense – there are $C_i, D_i \in \mathsf{GL}(w,\mathbb{L})$ such that $Y_i = C_i \cdot X_i \cdot D_i$, for every $i \in [d]$. Moreover, there are $c_1, \ldots, c_{d-1} \in \mathbb{L}^\times$ satisfying $C_1 = D_d = I_w$, $D_i \cdot C_{i+1} = c_i I_w$ for $i \in [d-1]$, and $\prod_{i=1}^{d-1} c_i = 1$. At a very high level, it is this uniqueness feature that guides the algorithm in finding a factorization. Such a factorization need not be unique if only the first two properties are satisfied. For instance[9],

$$\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \cdot \begin{bmatrix} 2x_3 - x_2 & x_4 \\ x_1 & x_3 \end{bmatrix} = \begin{bmatrix} x_3 & x_1 \\ x_4 & 2x_3 - x_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} = \begin{bmatrix} 2x_1x_3 & x_1x_4 + x_2x_3 \\ 2x_3^2 - x_2x_3 + x_1x_4 & 2x_3x_4 \end{bmatrix}.$$

Using Theorem 1, Theorem 2 addresses Problem 2 for $n \geq 4w^2$.

Theorem 2 (Average-case ABP reconstruction)**.** *For $n \geq 4w^2$, there is a randomized algorithm that takes as input blackbox access to a (n,d)-polynomial f computed by a random (w,d,n)-ABP over $\mathbb{F}$, and with probability $1-(wdn)^{-\Omega(1)}$ returns a (w,d,n)-ABP over $\mathbb{L}$ computing f. The algorithm runs in time $(d^{w^3} n \log q)^{O(1)}$ and queries the blackbox at points in $\mathbb{L}^n$.*

Remarks:

1. *Comparison to* [KNST17]: The differences are already highlighted before. Moreover, the algorithm in [KNST17] works over both $\mathbb{Q}$ and $\mathbb{F}_q$, whereas ours is over $\mathbb{F}_q$. The choice of finite fields comes from Theorem 3 (see the remarks following it).

2. *Time-complexity*: There is one step in the algorithm that finds the affine forms in X_1 and X_d by solving systems of polynomial equations over $\mathbb{F}$, and this takes $d^{O(w^3)}$ field operations. Except this step, every other step runs in $(wdn \log q)^{O(1)}$ time. If the complexity of this step is improved then the overall time complexity of the algorithm will also come down.

[8] The choice of the first row and column are arbitrary. The analysis holds if the entries of $X_{i+1} \cdots X_j$ are $\mathbb{F}$-linearly independent modulo the affine forms in some row and column of X_i. Also, we have not attempted to optimize this third property, in order to keep the analysis relatively simple. It may be possible to weaken the property significantly with a more careful analysis.

[9] We thank Rohit Gurjar for showing us a similar example, but with non-coprime determinants.

3. *Not pseudorandom*: Consider a formal (w,d,n)-ABP where the coefficients of the affine forms are distinct $\mathbf{y}$-variables, and let $h(\mathbf{x},\mathbf{y})$ be the polynomial computed by this ABP. Here, $|\mathbf{y}| = (n+1)\cdot(w^2(d-2)+2w) = m$ (say). If $w = O(\sqrt{n})$, the family $H = \{h(\mathbf{x},\mathbf{b}) : \mathbf{b} \in \mathbb{F}^m\}$ is not pseudorandom under the distribution defined by $\mathbf{b} \in_r \mathbb{F}^m$. This is because, the w affine forms in X_1 are linearly independent with high probability. So, the variety of $f = h(\mathbf{x},\mathbf{b})$ (denoted by $\mathbb{V}(f)$) has a subspace of dimension $n-w$ over $\mathbb{F}$; a random polynomial does not have this property with high probability. Using a randomized algorithm (Theorem 2.6 and 3.9 in [HW99]) we can check if $\mathbb{V}(f)$ has a large subspace in $(d^{w^2} n \log q)^{O(1)}$ time. Observe that $(d^{w^2} n \log q)^{O(1)} = d^{O(n)}$ for $w = O(\sqrt{n})$, and so the algorithm does not take time much larger than the number of monomials in f to distinguish it from a random (n,d)-polynomial thereby implying that H is not a pseudorandom family. However, a family not being pseudorandom under a distribution does not say much a priori about average-case reconstruction under the same distribution for the family. The latter is presumably a much harder problem for arbitrary non-pseudorandom polynomial families.

4. *Non-degenerate ABP*: Similar to pure product, we can state a set of non-degeneracy conditions such that the algorithm in Theorem 2 (with a slight modification) solves the reconstruction problem for ABPs satisfying these conditions. These non-degeneracy conditions are stated in Section 4.3, and the proof of Theorem 2 shows that a random (w,d,n)-ABP satisfies them with high probability, for $n \geq 4w^2$.

The proof of Theorem 1 requires an efficient affine equivalence test for the determinant. An n-variate polynomial $f(\mathbf{x})$ is *affine equivalent* to an m-variate polynomial g, for $n \geq m$, if there is an $A \in \mathbb{F}^{m\times n}$ of rank m and an $\mathbf{a} \in \mathbb{F}^m$ such that $f = g(A\cdot\mathbf{x}+\mathbf{a})$. Further, for $m = n$, f is *equivalent* to g if there is an $A \in \mathsf{GL}(n,\mathbb{F})$ such that $f = g(A\cdot\mathbf{x})$. Given blackbox access to a (n,w)-polynomial f, where $n \geq w^2$, the affine equivalence test problem for the determinant is to check whether f is affine equivalent to Det_w, and if yes then output a $B \in \mathbb{F}^{w^2\times n}$ of rank w^2 and a $\mathbf{b} \in \mathbb{F}^{w^2}$ such that $f = \mathsf{Det}_w(B\cdot\mathbf{x}+\mathbf{b})$. Here Det_w is the $w\times w$ symbolic determinant polynomial. The theorem below solves this problem over finite fields – it returns a $B \in \mathbb{L}^{w^2\times n}$ of rank w^2 and a $\mathbf{b} \in \mathbb{L}^{w^2}$.

Theorem 3 (Determinant equivalence test)**.** *There is a randomized algorithm that takes as input blackbox access to a (n,w)-polynomial $f \in \mathbb{F}[\mathbf{x}]$, where $n \geq w^2$, and does the following with probability $1-\frac{n^{O(1)}}{q}$: If f is affine equivalent to Det_w then it outputs a $B \in \mathbb{L}^{w^2\times n}$ of rank w^2 and a $\mathbf{b} \in \mathbb{L}^{w^2}$ such that $f = \mathsf{Det}_w(B\cdot\mathbf{x}+\mathbf{b})$, else it outputs '$f$ not affine equivalent to Det_w'. The algorithm runs in $(n\log q)^{O(1)}$ time and queries the blackbox at points in $\mathbb{L}^n$.*

Remarks:

1. *Comparison to [Kay12]*: An efficient equivalence test for the determinant over $\mathbb{C}$ was given in [Kay12]. The computation model in [Kay12] assumes that arithmetic over $\mathbb{C}$ and root finding of univariate polynomials over $\mathbb{C}$ can be done efficiently. While we follow the general strategy of analyzing the Lie algebra of the determinant and reduction to PS-equivalence from [Kay12], our algorithm is somewhat *simpler*: Unlike [Kay12], our algorithm does not involve the Cartan subalgebras and is almost the same as the simpler equivalence test for the permanent polynomial in [Kay12]. The simplification is achieved by showing that the characteristic polynomial of a random element of the Lie algebra of Det_w splits completely over $\mathbb{L}$ with high probability (Lemma 5.2) – this is crucial for Theorem 1 as it allows the algorithm to output a matrix factorization over a fixed low extension of $\mathbb{F}$, namely $\mathbb{L}$.

2. *Average-case ABP reconstruction over* $\mathbb{Q}$: In our arguments, Theorem 3 is the only place where we need the underlying field is finite. In other words, the algorithms in Theorems 1 and 2 work over $\mathbb{Q}$ if only there is an efficient equivalence test for Det_w over $\mathbb{Q}$. Also, if there is an affine equivalence test for Det_w that outputs $B, \mathbf{b}$ over the base field ($\mathbb{Q}$ or $\mathbb{F}$) then the algorithm in Theorem 2 would output an ABP over the base field.

1.4 Algorithms and their analysis

The algorithms mentioned in Theorem 1 and 2 are given in Algorithm 1 and 2, respectively. In this section, we briefly discuss their correctness and complexity – for the missing details, we allude to the relevant parts of the subsequent sections.

1.4.1 Analysis of Algorithm 1

Since $F = X_1 \cdot X_2 \ldots X_d$ is a random (w, d, n)-matrix product, with probability $1 - (wdn)^{-\Omega(1)}$, the first two properties of a pure product are satisfied: Every X_i is a *full rank* linear matrix, and $\det(X_1), \det(X_2), \ldots, \det(X_d)$ are coprime irreducible polynomials (see Claim 2.3). Claim 2.4 shows that the third property of a pure product is also satisfied with probability $1 - (wdn)^{-\Omega(1)}$. We analyze Algorithm 1 assuming that F is a pure product over $\mathbb{F}$ (which also implies that F is a pure product over $\mathbb{L}$). The third property of a pure product will be used only in Observation 3.5 in Section 3.2. The algorithm has three main stages:

1. *Computing the irreducible factors of* $\det(F)$ *(Steps 2–6):* From blackbox access to the entries of F, a blackbox access to $\det(F)$ is computed in $(wdn \log q)^{O(1)}$ time using Gaussian elimination. Subsequently, using Kaltofen-Trager's factorization algorithm [KT90], blackbox access to the irreducible factors $g_1, g_2, \ldots, g_d$ of $\det(F)$ are constructed in $(wdn \log q)^{O(1)}$ time (see Lemma 2.1). Since $\det(X_1), \ldots, \det(X_d)$ are coprime irreducible polynomials, there is a permutation σ of $[d]$, and $c_i \in \mathbb{F}^\times$ for all $i \in [d]$, such that $c_i \cdot \det(X_i) = g_{\sigma(i)}$ and $\prod_{i=1}^{d} c_i = 1$. For the next two stages, assume $w > 1$ as the $w = 1$ case gets solved readily at this stage.

2. *Affine equivalence test (Steps 9–16):* Let $j = \sigma(i)$ and X'_i be the matrix X_i with the affine forms in the first row multiplied by c_i. Then, $g_j = \det(X'_i) = c_i \cdot \det(X_i)$, which is affine equivalent to Det_w. At step 11, the algorithm in Theorem 3 (given in Section 5) finds a $B_j \in \mathbb{L}^{w^2 \times n}$ of rank w^2 and $\mathbf{b}_j \in \mathbb{L}^{w^2}$ such that $g_j = \mathrm{Det}_w(B_j \cdot \mathbf{x} + \mathbf{b}_j)$, with probability $1 - (wdn)^{-\Omega(1)}$. Let Z_j be the matrix obtained by appropriately replacing the entries of the $w \times w$ symbolic matrix with the affine forms in $B_j \cdot \mathbf{x} + \mathbf{b}_j$ such that $\det(Z_j) = g_j = \det(X'_i)$. This certifies that there are matrices $C_i, D_i \in \mathsf{SL}(w, \mathbb{L})$ satisfying, $Z_j = C_i \cdot X'_i \cdot D_i$ or $Z_j^T = C_i \cdot X'_i \cdot D_i$ (see Fact 1 in Section 5.1). Multiplying the first column of C_i with c_i, and calling the resulting matrix C_i again, we see that there are matrices $C_i, D_i \in \mathsf{GL}(w, \mathbb{L})$ satisfying, $Z_j = C_i \cdot X_i \cdot D_i$ or $Z_j^T = C_i \cdot X_i \cdot D_i$. Observe that such C_i, D_i are unique up to multiplications by elements in $\mathbb{L}^\times$ i.e., if $C_i \cdot X_i \cdot D_i = C'_i \cdot X_i \cdot D'_i$, where X_i is a full rank matrix, then $C'_i = \alpha C_i$ and $D'_i = \alpha^{-1} D_i$ for some $\alpha \in \mathbb{L}^\times$.

3. *Rearrangement of the retrieved matrices (Steps 19–22):* This stage is the most crucial part of Algorithm 1. At step 19, Algorithm 3 constructs the matrices $Y_1, Y_2, \ldots, Y_d$ by determining the permutation σ and whether $Z_{\sigma(i)} = C_i \cdot X_i \cdot D_i$ or $Z_{\sigma(i)}^T = C_i \cdot X_i \cdot D_i$. Internally, Algorithm

Algorithm 1 Average-case matrix factorization

INPUT: Blackbox access to w^2, (n, d)-polynomials $\{f_{st}\}_{s,t \in [w]}$ that constitute the entries of a random (w, d, n)- matrix product $F = X_1 \cdot X_2 \ldots X_d$.
OUTPUT: Linear matrices $Y_1, Y_2, \ldots, Y_d$ over $\mathbb{L}$ such that $F = Y_1 \cdot Y_2 \ldots Y_d$.

/* Factorization of the determinant */
Compute blackbox access to $\det(F)$.
Compute blackbox access to the irreducible factors of $\det(F)$; call them $g_1, g_2, \ldots, g_d$.
if the number of irreducible factors is not equal to d **then**
  Output 'Failed'.
end if

/* Affine equivalence test for determinant */
Set $j = 1$.
while $j \leq d$ **do**
  Call the algorithm in Theorem 3 with input as blackbox access to g_j; let B_j and $\mathbf{b}_j$ be its output. Construct the $w \times w$ full-rank linear matrix Z_j over $\mathbb{L}$ determined by B_j and $\mathbf{b}_j$.
  if the algorithm outputs 'g_j not affine equivalent to Det_w' **then**
    Output 'Failed'.
  end if
  Set $j = j + 1$.
end while

/* Rearrangement of the matrices */
Call Algorithm 3 on input blackbox access to F and $Z_1, \ldots, Z_d$, and let $Y_1, \ldots, Y_d$ be its output.
if Algorithm 3 outputs 'Rearrangement not possible' **then**
  Output 'Failed'.
end if

Output $Y_1, Y_2, \ldots, Y_d$.

3 uses Algorithm 4, which when given blackbox access to $F_d = F$ and a Z (that is either Z_k or Z_k^T for some $k \in [d]$), does the following with probability $1 - (wdn)^{-\Omega(1)}$: If $Z = C_d \cdot X_d \cdot D_d$ then it outputs a $\tilde{D}_d = a_d D_d$ for some $a_d \in \mathbb{L}^\times$. For all other cases – if $Z = C_i \cdot X_i \cdot D_i$ or $Z^T = C_i \cdot X_i \cdot D_i$ for $i \in [d-1]$, or $Z^T = C_d \cdot X_d \cdot D_d$ – it outputs 'Failed'. Algorithm 4 uses the critical fact that F is a pure product to accomplish the above and locate the unique last matrix. The running time of the algorithm, which is $(wdn \log q)^{O(1)}$, and its proof of correctness (which also gives the uniqueness of factorization mentioned in the remark after Theorem 1) are discussed in Section 3.2. Algorithm 3 calls Algorithm 4 on inputs F, Z_k and F, Z_k^T for all $k \in [d]$. If Algorithm 4 returns a matrix $\tilde{D}_d$ for some $k \in [d]$ on either inputs F, Z_k or F, Z_k^T then it sets $M_d = Z_k$ or $M_d = Z_k^T$ respectively, and $\sigma(d) = k$. Subsequently, Algorithm 3 computes blackbox access to a length $d-1$ matrix product $F_{d-1} = F \cdot \tilde{D}_d \cdot M_d^{-1} = X_1 \cdots X_{d-2} \cdot (X_{d-1} \cdot a_d C_d^{-1})$, and repeats the above process to compute M_{d-1} and $\sigma(d-1)$ with the inputs F_{d-1} and $\{Z_1, \ldots, Z_d\} \setminus Z_{\sigma(d)}$. Thus, using Algorithm 4 repeatedly, Algorithm 3 iteratively determines σ and $M_d, M_{d-1}, \ldots, M_2$: At the $(d-t+1)$-th iteration, for $t \in [d-1, 2]$, it computes a matrix $\tilde{D}_t = a_t(C_{t+1} \cdot D_t)$ for some $a_t \in \mathbb{L}^\times$, sets M_t and $\sigma(t)$ accordingly, creates blackbox access to $F_{t-1} = F_t \cdot \tilde{D}_t \cdot M_t^{-1}$ and prepares the list $\{Z_1, \ldots, Z_d\} \setminus \{Z_{\sigma(d)}, Z_{\sigma(d-1)}, \ldots, Z_{\sigma(t)}\}$ for the next iteration. Finally, setting $Y_1 = F_1$ and $Y_i = M_i \cdot \tilde{D}_i^{-1}$, for all $i \in [2, d]$, we have $F = \prod_{i=1}^{d} Y_i$.

1.4.2 Analysis of Algorithm 2

Let f be the polynomial computed by a (w, d, n)-ABP $X_1 \cdot X_2 \ldots X_d$. We can assume that f is a homogeneous degree-d polynomial and the entries in each X_i are linear forms (i.e., affine forms with constant term zero), owing to the following simple homogenization trick.

Homogenization of ABP: Consider the $(n+1)$-variate homogeneous degree-d polynomial

$$f_{\text{hom}} = x_0^d \cdot f\left(\frac{x_1}{x_0}, \frac{x_2}{x_0}, \ldots, \frac{x_n}{x_0}\right).$$

The polynomial f_{hom} is computable by the (w, d, n)-ABP $X_1' \cdot X_2' \ldots X_d'$, where X_i' is equal to X_i but with the constant term in the affine forms multiplied by x_0. If we construct an ABP for f_{hom} then an ABP for f is obtained by setting $x_0 = 1$.

We give an overview of the three main stages in Algorithm 2. As in Algorithm 1, the matrices $X_1, X_2, \ldots, X_d$ are assumed to be full rank linear matrices and further, for a similar reason, the $2w$ linear forms in X_1 and X_d are assumed to be $\mathbb{F}$-linearly independent. For a field $\mathbb{K} \supseteq \mathbb{F}$, we say f is zero modulo a $\mathbb{K}$-linear space $\mathcal{X} = \text{span}_{\mathbb{K}}\{l_1, \ldots, l_w\}$, where l_i's are linear forms in $\mathbb{K}[\mathbf{x}]$, if f is in the ideal of $\mathbb{K}[\mathbf{x}]$ generated by $\{l_1, \ldots, l_w\}$. This is also denoted by $f = 0 \mod \langle l_1, \ldots, l_w \rangle$.

1. *Computing the corner spaces (Steps 2–6):* Polynomial f is zero modulo each of the two w-dimensional $\mathbb{F}$-linear spaces $\mathcal{X}_1$ and $\mathcal{X}_d$ spanned by the linear forms in X_1 and X_d respectively. We show in Lemma 4.1, if $n \geq 4w^2$ then with probability $1 - (wdn)^{-\Omega(1)}$ the following holds: Let $\mathbb{K} \supseteq \mathbb{F}$ be any field. If $f = 0 \mod \langle l_1, \ldots, l_w \rangle$, where l_i's are linear forms in $\mathbb{K}[\mathbf{x}]$, then the l_i's either belong to the $\mathbb{K}$-span of the linear forms in X_1 or belong to the $\mathbb{K}$-span of the linear forms in X_d. In this sense, the spaces $\mathcal{X}_1$ and $\mathcal{X}_d$ are unique. The algorithm invokes Algorithm 5 which computes bases of $\mathcal{X}_1$ and $\mathcal{X}_d$ by solving $O(n)$ systems

Algorithm 2 Average-case ABP reconstruction

INPUT: Blackbox access to a (n, d)-polynomial f computed by a random (w, d, n)-ABP.
OUTPUT: A (w, d, n)-ABP over $\mathbb{L}$ computing f.

/* Computing the corner spaces */
Call Algorithm 5 on f to compute bases of the two *unique* w-dimensional $\mathbb{F}$-linear spaces $\mathcal{X}_1$ and $\mathcal{X}_d$, spanned by linear forms in $\mathbb{F}[\mathbf{x}]$, such that f is zero modulo each of $\mathcal{X}_1$ and $\mathcal{X}_d$.
if Algorithm 5 outputs 'Failed' **then**
    Output 'Failed to construct an ABP'.
end if
Compute a transformation $A \in \mathsf{GL}(n, \mathbb{F})$ that maps the bases of $\mathcal{X}_1$ and $\mathcal{X}_d$ to distinct variables $\mathbf{y} = \{y_1, y_2, \ldots, y_w\}$ and $\mathbf{z} = \{z_1, z_2, \ldots z_w\}$ respectively, where $\mathbf{y}, \mathbf{z} \subseteq \mathbf{x}$. Let $\mathbf{r} = \mathbf{x} \setminus (\mathbf{y} \uplus \mathbf{z})$, $X_1' = (y_1\ y_2\ \ldots\ y_w)$, $X_d' = (z_1\ \ z_2\ \ \ldots\ z_w)^T$ and $f' = f(A \cdot \mathbf{x})$.

/* Computing the coefficients of the $\mathbf{r}$ variables in the intermediate matrices */
Construct blackbox access to the w^2 polynomials that constitute the entries of the $w \times w$ matrix $F = \left(\frac{\partial f'}{\partial y_s z_t}\ |_{\mathbf{y}=0, \mathbf{z}=0}\right)_{s,t \in [w]}$.
Call Algorithm 1 on input F to compute a factorization of F as $S_2 \cdot S_3 \ldots S_{d-1}$.
if Algorithm 1 outputs 'Failed' **then**
    Output 'Failed to construct an ABP'.
end if

/* Computing the coefficients of the $\mathbf{y}$ and $\mathbf{z}$ variables in the intermediate matrices */
Call Algorithm 6 on inputs f' and $\{S_2, S_3, \ldots, S_{d-1}\}$ to compute matrices $T_2, T_3, \ldots, T_{d-1}$ such that f' is computed by the ABP $X_1' \cdot T_2 \cdots T_{d-1} \cdot X_d'$.
if Algorithm 6 outputs 'Failed' **then**
    Output 'Failed to construct an ABP'.
end if
Apply the transformation A^{-1} on the $\mathbf{x}$ variables in the matrices X_1', X_d', and T_k for $k \in [2, d-1]$. Call the resulting matrices Y_1, Y_d, and Y_k for $k \in [2, d-1]$ respectively.
Output $Y_1 \cdot Y_2 \ldots Y_d$ as the ABP computing f.

of polynomial equations over $\mathbb{F}$. Such a system has $d^{O(w^2)}$ equations in $m = O(w^3)$ variables and the degree of the polynomials in the system is at most d; we intend to find all the solutions in $\mathbb{F}^m$. It turns out that owing to the uniqueness of $\mathcal{X}_1$ and $\mathcal{X}_d$, the variety over $\overline{\mathbb{F}}$ (the algebraic closure of $\mathbb{F}$) defined by such a system has exactly two points and these points lie in $\mathbb{F}^m$. From the two solutions, bases of $\mathcal{X}_1$ and of $\mathcal{X}_d$ can be derived. The two solutions of the system are computed by a randomized algorithm running in $(d^{w^3} \log q)^{O(1)}$ time ([Ier89, HW99], see Lemma 2.2) – the algorithm exploits the fact that the variety over $\overline{\mathbb{F}}$ is zero-dimensional. Thus, at step 2, the two spaces are either equal to $\mathcal{X}_1$ and $\mathcal{X}_d$ or $\mathcal{X}_d$ and $\mathcal{X}_1$ respectively. Without loss of generality, we assume the former. Once bases of the corner spaces $\mathcal{X}_1$ and $\mathcal{X}_d$ are computed, an invertible transformation A maps the linear forms in the bases to distinct variables (as the linear forms in X_1 and X_d are $\mathbb{F}$-linearly independent).

2. *Computing the coefficients of the* $\mathbf{r}$ *variables (Steps 9–13)*: There is an ABP $X_1' \cdot X_2' \dots X_d'$ computing $f' = f(A \cdot \mathbf{x})$, where X_1' and X_d' are equal to $(y_1\ y_2\ \dots\ y_w)$ and $(z_1\ z_2\ \dots\ z_w)^T$ respectively. For $k \in [2, d-1]$, let $R_k = (X_k')_{\mathbf{y}=0,\mathbf{z}=0}$ and $F = R_2 \cdot R_3 \dots R_{d-1}$. As $X_1 \cdot X_2 \dots X_d$ is a random (w, d, n)-ABP, $R_2 \cdot R_3 \dots R_{d-1}$ is a random $(w, d-2, n-2w)$-matrix product over $\mathbb{F}$. The (s,t)-th entry of F is equal to $\left(\frac{\partial f'}{\partial y_s z_t}\right)_{\mathbf{y}=0,\mathbf{z}=0}$, for $s, t \in [w]$. Blackbox access to each of the w^2 entries of F are constructed in $(wdn \log q)^{O(1)}$ time using Claim 2.1. From F, Algorithm 1 computes linear matrices $S_2, \dots, S_{d-1}$ over $\mathbb{L}$ in $\mathbf{r} = \mathbf{x} \setminus (\mathbf{y} \uplus \mathbf{z})$ variables such that $F = S_2 \cdot S_3 \dots S_{d-1}$. Moreover, the uniqueness of factorization implies there are linear matrices $T_2, \dots, T_{d-1}$ over $\mathbb{L}$ in the $\mathbf{x}$-variables, satisfying $(T_k)_{\mathbf{y}=0,\mathbf{z}=0} = S_k$, such that f' is computed by the ABP $X_1' \cdot T_2 \cdots T_{d-1} \cdot X_d'$.

3. *Computing the coefficients of* $\mathbf{y}$ *and* $\mathbf{z}$ *variables in* T_k *(Steps 16–20)*: Algorithm 6 finds the coefficients of the $\mathbf{y}$ and $\mathbf{z}$ variables in the linear forms present in $T_2, \dots, T_{d-1}$ in $(wdn \log q)^{O(1)}$ time. We present the idea here; the detail proof of correctness is given in Section 4.2. In the following discussion, $M(i,j)$ denotes the (i,j)-th entry, $M(i,*)$ the i-th row, and $M(*,j)$ the j-th column of a linear matrix M. Let us focus on finding the coefficients of y_1 in the linear forms present in $T_2(1,*), T_3, \dots, T_{d-2}, T_{d-1}(*,1)$. There are $w^2(d-4) + 2w$ linear forms in these matrices and these would be indexed by $[w^2(d-4) + 2w]$. Let c_e be the coefficient of y_1 in the e-th linear form l_e for $e \in [w^2(d-4) + 2w]$. We associate a polynomial $h_e(\mathbf{r})$ in $\mathbf{r}$ variables with l_e as follows: If l_e is the (i,j)-th entry of T_k then $h_e \stackrel{\text{def}}{=} [S_2(1,*) \cdot S_3 \cdots S_{k-2} \cdot S_{k-1}(*,i)] \cdot [S_{k+1}(j,*) \cdot S_{k+2} \cdots S_{d-2} \cdot S_{d-1}(*,1)]$, by identifying the 1×1 matrix of the R.H.S with the entry of the matrix. Observe that if f' is treated as a polynomial in $\mathbf{y}$ and $\mathbf{z}$ variables with coefficients in $\mathbb{L}(\mathbf{r})$ then the coefficient of $y_1^2 z_1$ is exactly $\sum_{e \in [w^2(d-4)+2w]} c_e \cdot h_e(\mathbf{r})$. On the other hand, this coefficient is $\left(\frac{\partial f'}{\partial y_1^2 z_1}\right)_{\mathbf{y}=0,\mathbf{z}=0}$, for which we can obtain blackbox access using Claim 2.1. This allows us to write the equation,

$$\sum_{e=1}^{w^2(d-4)+2w} c_e \cdot h_e(\mathbf{r}) = \left(\frac{\partial f'}{\partial y_1^2 z_1}\right)_{\mathbf{y}=0,\mathbf{z}=0}. \tag{1}$$

We show in Lemma 4.2 and Corollary 4.1 that the polynomials h_e, for $e \in [w^2(d-4) + 2w]$, are $\mathbb{L}$-linearly independent with probability $1 - (wdn)^{-\Omega(1)}$, over the randomness of the input f. By substituting random values to the $\mathbf{r}$ variables in the above equation, we can set up a system of $w^2(d-4)+2w$ linear equations in the c_e's. The linear independence of the h_e's ensures that we can solve for c_e (by Claim 2.2).

1.4.3 Proof strategy for Theorem 3

The algorithm in Theorem 3 has three stages:

1. *Reduction to equivalence testing*: Applying known techniques – 'variable reduction' (Claim 5.1) and 'translation equivalence' (Claim 5.2) – the affine equivalence testing problem is efficiently reduced to equivalence testing for Det_w with high probability. An equivalence test takes blackbox access to a w^2-variate polynomial $g(\mathbf{y})$ as input and does the following with high probability: If g is equivalent to Det_w then it outputs a $Q \in \mathsf{GL}(w^2, \mathbb{L})$ such that $g = \mathsf{Det}_w(Q \cdot \mathbf{y})$ else it outputs 'g not equivalent to Det_w'.

2. *Reduction to PS-equivalence*: The reduction is given in Algorithm 7. The algorithm proceeds by computing an $\mathbb{F}$-basis of the Lie algebra of the group of symmetries of g (denoted as $\mathfrak{g}_g$, see Claim 5.3). It then picks an element F uniformly at random from $\mathfrak{g}_g$ and computes its characteristic polynomial $h(x)$. Since $F \in \mathfrak{g}_g$, it is similar to a $L \in \mathfrak{g}_{\mathsf{Det}_w}$ (see Fact 3 in Section 5.1), implying that their characteristic polynomials are equal. As F is a random element of $\mathfrak{g}_g$, L is also a random element of $\mathfrak{g}_{\mathsf{Det}_w}$. In Lemma 5.2, we show that the characteristic polynomial h of a $L \in_r \mathfrak{g}_{\mathsf{Det}_w}$ is square-free and splits completely over $\mathbb{L}$, with high probability. (This lemma makes our reduction to PS-equivalence simpler than [Kay12], enabling the equivalence test to work over finite fields.) The roots of h are computed in randomized $(w \log q)^{O(1)}$ time ([CZ81], see also [vzGG03]). From the roots, a $D \in \mathsf{GL}(w^2, \mathbb{L})$ can be computed such that $D^{-1}FD$ is diagonal[10]. Thereafter, the structure of the group of symmetries of Det_w and its Lie algebra helps argue, in Section 5.2, that $f(D \cdot \mathbf{x})$ is PS-equivalent to Det_w.

3. *Doing the PS-equivalence*: This step follows directly from [Kay12] (see Lemma 5.1).

1.5 Few questions

The following questions are immediate from the above discussions:

(a) Can we compute the corner spaces in $(wd \log q)^{O(1)}$ time? If so then the overall complexity of the algorithm would come down to $(wd \log q)^{O(1)}$.

(b) In the equivalence test for the determinant, can we output a linear matrix over the base field $\mathbb{F}$ instead of a matrix over the extension $\mathbb{L}$?

(c) Is it possible to do nontrivial reconstruction in the average-case when w is significantly larger than $\sqrt{n}$, say for $w = \frac{n}{2}$?

(d) For w significantly larger than $\sqrt{n}$, say $w = n^2$, can we show that linear factorization of a random (w, d, n)-matrix product is unique (in the sense as mentioned in the second remark after Theorem 1)?

[10] In [Kay12], a basis of the centralizer of F in $\mathfrak{g}_g$ is computed first and then a $D \in \mathsf{GL}(w^2, \mathbb{C})$ is obtained that simulaneously diagonalizes this basis.

2 Preliminaries

2.1 Notations

$\mathsf{GL}(w,\mathbb{F})$ is the set of $w \times w$ invertible matrices over $\mathbb{F}$, and $\mathsf{SL}(w,\mathbb{F})$ the set of $w \times w$ matrices over $\mathbb{F}$ with determinant one. Bold letters $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}, \mathbf{w}$ are used to represent either column vectors (or sets) of variables or column vectors of field elements, calligraphic letters like $\mathcal{X}$ to represent vector spaces, capital letters like A, B, C, S, T for matrices or sets – the context of a usage of any of these symbols would hopefully make its purpose clear. The derivative of a polynomial f with respect to a monomial μ is denoted as $\frac{\partial f}{\partial \mu}$ or $\partial_\mu f$.

2.2 Algorithmic preliminaries

The following result on blackbox polynomial factorization is proved in [KT90].

Lemma 2.1 ([KT90]). *There is a randomized algorithm that takes as input blackbox access to a (n,d)-polynomial f over $\mathbb{F}$, and constructs blackbox access to the irreducible factors of f over $\mathbb{F}$ in $(nd\log q)^{O(1)}$ time with success probability $1 - \frac{(nd)^{O(1)}}{q}$.*

Let I be an ideal of $\mathbb{F}[\mathbf{x}]$ generated by (n,d)-polynomials $g_1, \ldots, g_m$, and $\mathbb{V}_{\overline{\mathbb{F}}}(I)$ the variety or the algebraic set defined by I over $\overline{\mathbb{F}}$. $\mathbb{V}_{\overline{\mathbb{F}}}(I)$ is zero-dimensional if it has finitely many points. We say a point $\mathbf{a} \in \mathbb{V}_{\overline{\mathbb{F}}}(I)$ is $\mathbb{F}$-rational if $\mathbf{a} \in \mathbb{F}^n$. The proof of the next result follows from [Ier89] (see also [HW99]).

Lemma 2.2 ([Ier89]). *There is a randomized algorithm that takes input m, (n,d)-polynomials $g_1, g_2, \ldots, g_m$ generating an ideal I of $\mathbb{F}[\mathbf{x}]$. If $\mathbb{V}_{\overline{\mathbb{F}}}(I)$ is zero-dimensional and all points in it are $\mathbb{F}$-rational then the algorithm computes all the points in $\mathbb{V}_{\overline{\mathbb{F}}}(I)$ with probability $1 - \exp(-mnd\log q)$. The running time of the algorithm is $(md^n \log q)^{O(1)}$.*

A similar result, but for homogeneous $g_1, \ldots, g_m$, follows from [Laz01].

2.3 A few useful facts

We list down four useful claims here. A proof of the first can be given using interpolation. Proofs of the next two follow from applications of the Schwartz-Zippel lemma [Sch80, Zip79].

Claim 2.1. *There is a deterministic algorithm that given blackbox access to a (n,d)-polynomial $f \in \mathbb{F}[\mathbf{x}]$, and a monomial μ of constant degree in $\mathbf{x}$, computes blackbox access to $\partial_\mu f$ in $(nd\log q)^{O(1)}$ time.*

Claim 2.2. *Let $f_1, f_2, \ldots, f_m$ be $\mathbb{F}$-linearly independent (n,d)-polynomials in $\mathbb{F}[\mathbf{x}]$. If $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ are points in $\mathbb{F}^n$ chosen independently and uniformly at random, then the matrix $(f_t(\mathbf{a}_s))_{s,t\in[m]}$ has rank m over $\mathbb{F}$ with probability at least $1 - \frac{dm}{q}$.*

Claim 2.3. *Let $X_1 \cdot X_2 \ldots X_d$ be a random (w,d,n)-matrix product over $\mathbb{F}$. If $n \geq w^2$ then $X_1, X_2, \ldots, X_d$ are full rank linear matrices and $\det(X_1), \det(X_2), \ldots, \det(X_d)$ are coprime irreducible polynomials with probability $1 - (wdn)^{-\Omega(1)}$.*

The following claim implies that a random matrix product satisfies the third property of a pure product with high probability.

Claim 2.4. *If $E = Q_1 \cdots Q_\ell$ is a random (w, ℓ, m)-matrix product over $\mathbb{F}$, where $w^2 + 1 \leq m \leq n$ and $\ell \leq d$, then the entries of E are $\mathbb{F}$-linearly independent with probability $1 - (wdn)^{-\Omega(1)}$.*

If the entries of E are $\mathbb{F}$-linearly independent then they are also $\mathbb{L}$-linearly independent. A proof of the claim is given in Appendix A.

3 Average-case matrix factorization: Proof of Theorem 1

The algorithm in Theorem 1 is presented in Algorithm 1. To complete the analysis, given in Section 1.4.1, we need to argue the correctness of the key step of rearrangement of the matrices (Algorithm 3) by finding the last matrix (Algorithm 4). As the functioning of Algorithm 3 is already sketched out in Section 1.4.1, the reader may skip to Section 3.2. For completeness, we include an analysis of Algorithm 3 in the following subsection.

3.1 Rearranging the matrices

Recall, we have assumed F is a pure (w, d, n)-matrix product $X_1 \cdot X_2 \ldots X_d$ over $\mathbb{F}$, and hence also a pure product over $\mathbb{L}$. The inputs to Algorithm 3 are d full rank linear matrices $Z_1, Z_2, \ldots, Z_d$ over $\mathbb{L}$ such that there are matrices $C_i, D_i \in \mathsf{GL}(w, \mathbb{L})$ and a permutation σ of $[d]$ satisfying $Z_{\sigma(i)} = C_i \cdot X_i \cdot D_i$ or $Z^T_{\sigma(i)} = C_i \cdot X_i \cdot D_i$ for every $i \in [d]$. Algorithm 3 iteratively determines σ (implicitly) by repeatedly using Algorithm 4. The behavior of Algorithm 4 is summarized in the lemma below. For the lemma statement, assume $n \geq 2w^2$, Z is a full rank linear matrix over $\mathbb{L}$, and F_t is a pure (w, t, n)-matrix product $R_1 \cdot R_2 \ldots R_t$ over $\mathbb{L}$, where $t \leq d$. Further, there are matrices $C, D \in \mathsf{GL}(w, \mathbb{L})$ and $i \in [t]$ such that $Z = C \cdot R_i \cdot D$ or $Z^T = C \cdot R_i \cdot D$.

Lemma 3.1. *Algorithm 4 takes input Z and blackbox access to the w^2 entries of F_t, and with probability $1 - (wdn)^{-\Omega(1)}$ does this: If $Z = C \cdot R_t \cdot D$ then it outputs a $\tilde{D} = aD$ for an $a \in \mathbb{L}^\times$, and for all other cases – $Z = C \cdot R_i \cdot D$ or $Z^T = C \cdot R_i \cdot D$ for $i \in [t-1]$, or $Z^T = C \cdot R_t \cdot D$ – it outputs 'Failed'.*

Algorithm 4 and the proof of Lemma 3.1 are presented in Section 3.2. We analyze Algorithm 3 below by tracing its steps:

Step 2: The algorithm enters an outer loop and iterates from $t = d$ to $t = 2$. For a fixed $t \in [d, 2]$, at the start of the loop the algorithm ensures F_t is a pure (w, t, n)-matrix product $R_1 \cdot R_2 \ldots R_t$ over $\mathbb{L}$. For $t = d$, $R_i = X_i$ for all $i \in [d]$. Further, there is a permutation σ_t of $[t]$, and for every $i \in [t]$ there are matrices $C_i, D_i \in \mathsf{GL}(w, \mathbb{L})$ such that either $Z_{\sigma_t(i)} = C_i \cdot R_i \cdot D_i$ or $Z^T_{\sigma_t(i)} = C_i \cdot R_i \cdot D_i$. In the loop, the algorithm determines $\sigma_t(t)$ and whether $Z_{\sigma_t(t)} = C_t \cdot R_t \cdot D_t$ or $Z^T_{\sigma_t(t)} = C_t \cdot R_t \cdot D_t$.

Steps 4–21: Inside the inner loop, the algorithm calls Algorithm 4 on inputs F_t, Z_k (step 5) and F_t, Z_k^T (step 13) for all $k \in [t]$. By Lemma 3.1, only when $k = \sigma_t(t)$, Algorithm 4 returns a $\tilde{D} = a_t D_t$ for some $a_t \in \mathbb{L}^\times$. The renaming of Z_k and Z_t (in steps 7 and 15) ensures that we have a suitable permutation σ_{t-1} of $[t-1]$ in the next iteration of the outer loop. The setting of M_t (in steps 8 and 16) implies that $M_t = C_t \cdot R_t \cdot D_t$. Hence,

$$F_{t-1} = F_t \cdot \tilde{D}_t \cdot M_t^{-1} = (R_1 \cdot R_2 \ldots R_{t-1}) \cdot (a_t C_t^{-1}).$$

Algorithm 3 Rearrangement of the matrices

INPUT: Blackbox access to F, and $w \times w$ full rank linear matrices $Z_1, Z_2, \ldots, Z_d$ over $\mathbb{L}$.
OUTPUT: Linear matrices $Y_1, Y_2, \ldots, Y_d$ over $\mathbb{L}$ such that $F = Y_1 \cdot Y_2 \cdots Y_d$.

Set $t = d, k = 1$, and $F_d = F$.
while $t > 1$ **do**

  while $k \leq t$ **do**
    Call Algorithm 4 on inputs F_t and Z_k.
    if Algorithm 4 outputs $\tilde{D}$ **then**
      Rename Z_k as Z_t and Z_t as Z_k, and set $\tilde{D}_t = \tilde{D}$. /* σ is determined implicitly. */
      Set $M_t = Z_t$ and $F_{t-1} = F_t \cdot \tilde{D}_t \cdot M_t^{-1}$.
      Set $k = 1$ and $t = t - 1$.
      Exit the inner loop.
    end if

    Call Algorithm 4 on inputs F_t and Z_k^T.
    if Algorithm 4 outputs a $\tilde{D}$ **then**
      Rename Z_k as Z_t and Z_t as Z_k, and set $\tilde{D}_t = \tilde{D}$. /* σ is determined implicitly. */
      Set $M_t = Z_t^T$ and $F_{t-1} = F_t \cdot \tilde{D}_t \cdot M_t^{-1}$.
      Set $k = 1$ and $t = t - 1$.
      Exit the (inner) loop.
    end if
    Set $k = k + 1$.
  end while

  if $k = t + 1$ **then**
    Exit the (outer) loop.
  end if

end while

if $t \geq 2$ **then**
  Output 'Rearrangement not possible'.
else
  Set $Y_1 = F_1$, and $Y_t = M_t \cdot \tilde{D}_t^{-1}$ for all $t \in [2, d]$. Output $Y_1, \ldots, Y_d$.
end if

Note that F_{t-1} is a pure $(w, t-1, n)$-matrix product over $\mathbb{L}$. By reusing symbols and calling $R_{t-1} \cdot (a_t C_t^{-1})$ as R_{t-1}, and $a_t^{-1} C_t \cdot D_{t-1}$ as D_{t-1}, we observe that the setup at step 2 is maintained in the next iteration of the outer loop.

Step 32: As $F_{t-1} = F_t \cdot \tilde{D}_t \cdot M_t^{-1}$ at every iteration of the outer loop, setting $Y_t = M_t \cdot \tilde{D}_t^{-1}$ implies $F_{t-1} = F_t \cdot Y_t^{-1}$ for every $t \in [d, 2]$. Therefore, $F = F_d = Y_1 \cdots Y_d$.

3.2 Determining the last matrix: Proof of Lemma 3.1

We give an overview of the proof by first assuming that Z is the 'last' matrix in the pure product F_t. The correctness of the idea is then made precise by tracing the steps of Algorithm 4.

Overview: Suppose $Z = C \cdot R_t \cdot D$, where $C, D \in \mathsf{GL}(w, \mathbb{L})$. As Z is a full rank linear matrix, we can assume the entries of Z are distinct variables, by applying an invertible linear transformation. For any polynomial $h \in \mathbb{L}[\mathbf{x}]$, $h \mod \det(Z)$ can be identified with an element of $\mathbb{L}(\mathbf{x})$. This is because, $\det(Z)$ is multilinear and so there is an injective ring homomorphism from $\mathbb{L}[\mathbf{x}]/(\det(Z))$ to $\mathbb{L}(\mathbf{x})$ via a simple substitution map taking a variable to a rational function. Let $Z', F_t' \in \mathbb{L}(\mathbf{x})^{w \times w}$ be obtained by reducing the entries of Z and F_t, respectively, modulo $\det(Z)$. The coprimality of the determinants of $R_1, \ldots, R_t$ and their full rank nature imply,

$$D \cdot \text{Kernel}_{\mathbb{L}(\mathbf{x})}(Z') = \text{Kernel}_{\mathbb{L}(\mathbf{x})}(F_t'),$$

and these two kernels have dimensions one. A basis of $\text{Kernel}_{\mathbb{L}(\mathbf{x})}(Z')$ can be easily derived as Z is known explicitly. However, we only have blackbox access to F_t'. To leverage the above relation, we compute bases of $\text{Kernel}_{\mathbb{L}}(F_t'(\mathbf{a}))$ and $\text{Kernel}_{\mathbb{L}}(Z'(\mathbf{a}))$ for several random $\mathbf{a} \in_r \mathbb{F}^n$, and form two matrices $U, V \in \mathsf{GL}(w, \mathbb{L})$ from these bases so that D equals $U \cdot V^{-1}$ (up to scaling by elements in $\mathbb{L}^\times$). Hereafter, $\text{Kernel}_{\mathbb{L}}$ will also be denoted as Ker in Algorithm 4 and its analysis.

Applying an invertible linear map (Step 2): The invertible linear transformation lets us assume that $Z = (z_{lk})_{l,k \in [w]}$, where z_{lk}'s are distinct variables in $\mathbf{x}$.

Reducing Z and F_t modulo $\det(Z)$ (Step 5): The reduction of the entries of Z and the blackbox entries of F_t modulo $\det(Z)$ is achieved by the substitution,

$$z_{11} = -\frac{\sum_{k=2}^{w} z_{1k} \cdot N_{1k}}{N_{11}}.$$

After the substitution, the matrices become Z' and $F_t' = R_1' \cdot R_2' \ldots R_t'$ respectively. As there are $i \in [t]$ and $C, D \in \mathsf{GL}(w, \mathbb{L})$ such that either $Z = C \cdot R_i \cdot D$ or $Z^T = C \cdot R_i \cdot D$, we have either $Z' = C \cdot R_i' \cdot D$ or $(Z')^T = C \cdot R_i' \cdot D$ and hence $\det(Z') = \det(R_i') = \det(F_t') = 0$.

Observation 3.1. *1.* $\text{Kernel}_{\mathbb{L}(\mathbf{x})}(Z') = \text{span}_{\mathbb{L}(\mathbf{x})}\{(N_{11}\ N_{12}\ \ldots N_{1w})^T\}$,

2. $\text{Kernel}_{\mathbb{L}(\mathbf{x})}((Z')^T) = \text{span}_{\mathbb{L}(\mathbf{x})}\{(N_{11}\ N_{21}\ \ldots N_{w1})^T\}$.

Hence, $\text{Kernel}_{\mathbb{L}(\mathbf{x})}(Z')$ has dimension one, and the observation below implies $\text{Kernel}_{\mathbb{L}(\mathbf{x})}(F_t')$ is also one dimensional. The proof follows from the coprimality of $\det(R_1), \det(R_2), \ldots, \det(R_t)$.

Algorithm 4 Determining the last matrix

INPUT: Blackbox access to a (w,t,n)-matrix product F_t and a full rank linear matrix Z over $\mathbb{L}$.
OUTPUT: A matrix $\tilde{D} \in \mathsf{GL}(w,\mathbb{L})$, if Z is the 'last' matrix of the product F_t.

/* Applying an invertible linear map */
Let the first w^2 variables in $\mathbf{x}$ be $\mathbf{z} = \{z_{lk}\}_{l,k\in[w]}$. Compute an invertible linear map A that maps the affine forms in Z to distinct $\mathbf{z}$ variables, and apply A to the w^2 blackbox entries of F_t. Reusing symbols, $Z = (z_{lk})_{l,k\in[w]}$ and F_t is the matrix product after the transformation.

/* Reducing Z and F_t modulo $\det(Z)$ */
Let N_{lk} be the (l,k)-th cofactor of Z, for $l,k \in [w]$. Substitute $z_{11} = \frac{-\sum_{k=2}^{w} z_{1k}N_{1k}}{N_{11}}$ in Z and in the blackbox for F_t. Call the matrices Z' and F_t' respectively after the substitution.

/* Computing the kernels at random points */
for $k = 1$ **to** $w+1$ **do**
  Choose $\mathbf{a}_k, \mathbf{b}_k \in_r \mathbb{F}^n$. Compute bases of $\mathrm{Ker}(F_t'(\mathbf{a}_k))$, $\mathrm{Ker}(Z'(\mathbf{a}_k))$, $\mathrm{Ker}(F_t'(\mathbf{b}_k))$, $\mathrm{Ker}(Z'(\mathbf{b}_k))$. Pick non-zero $\mathbf{u}_k \in \mathrm{Ker}(F_t'(\mathbf{a}_k))$, $\mathbf{v}_k \in \mathrm{Ker}(Z'(\mathbf{a}_k))$, $\mathbf{w}_k \in \mathrm{Ker}(F_t'(\mathbf{b}_k))$, $\mathbf{s}_k \in \mathrm{Ker}(Z'(\mathbf{b}_k))$. If the computation fails (i.e., $N_{11}(\mathbf{a}_k) = 0$ or $N_{11}(\mathbf{b}_k) = 0$), or any of the kernels is not one dimensional, output 'Failed'.
end for

/* Extracting D from the kernels */
Compute $\alpha_k, \beta_k, \gamma_k, \delta_k \in \mathbb{L}$ for $k \in [w]$ such that $\mathbf{u}_{w+1} = \sum_{k=1}^{w} \alpha_k \mathbf{u}_k$, $\mathbf{v}_{w+1} = \sum_{k=1}^{w} \beta_k \mathbf{v}_k$, $\mathbf{w}_{w+1} = \sum_{k=1}^{w} \gamma_k \mathbf{w}_k$ and $\mathbf{s}_{w+1} = \sum_{k=1}^{w} \delta_k \mathbf{s}_k$. If the computation fails, or any of α_k, β_k, γ_k, δ_k is zero for some $k \in [w]$, output 'Failed'.

Set $U, V, W, S \in \mathbb{L}^{w\times w}$ such that the k-th column of U, V, W, S are $\frac{\alpha_k \cdot \mathbf{u}_k}{\beta_k}$, $\mathbf{v}_k$, $\frac{\gamma_k \cdot \mathbf{w}_k}{\delta_k}$, $\mathbf{s}_k$ respectively. If any of $U, V, W, S \notin \mathsf{GL}(w,\mathbb{L})$, output 'Failed'.

if $UV^{-1}SW^{-1}$ is a scalar matrix **then**
  Set $\tilde{D} = U \cdot V^{-1}$ and output $\tilde{D}$.
else
  Output 'Failed'. /* The check fails w.h.p if Z is not the 'last' matrix */
end if

Observation 3.2. *For all $j \in [t]$ and $j \neq i$, $\det(R'_j) \neq 0$, and so the dimension of $\mathrm{Kernel}_{\mathbb{L}(\mathbf{x})}(F'_t)$ is one.*

Computing the kernels at random points (Steps 8–10): The following observation shows that the algorithm does not fail at step 9 with high probability. The proof is immediate from the above two observations and an application of the Schwartz-Zippel lemma.

Observation 3.3. *Let $\mathbf{a}_k, \mathbf{b}_k \in_r \mathbb{F}^n$ for $k \in [w+1]$. Then, for every $k \in [w+1]$, and $\mathbf{a} = \mathbf{a}_k$ or $\mathbf{b}_k$,*

1. $\mathrm{Ker}(Z'(\mathbf{a})) = \mathrm{span}_{\mathbb{L}}\{(N_{11}(\mathbf{a})\ N_{12}(\mathbf{a})\ \ldots N_{1w}(\mathbf{a}))^T\}$,
2. $\mathrm{Ker}((Z'(\mathbf{a}))^T) = \mathrm{span}_{\mathbb{L}}\{(N_{11}(\mathbf{a})\ N_{21}(\mathbf{a})\ \ldots N_{w1}(\mathbf{a}))^T\}$,

and $\mathrm{Ker}(F'_t(\mathbf{a}_k))$, $\mathrm{Ker}(F'_t(\mathbf{b}_k))$ are one dimensional subspaces of $\mathbb{L}^w$, with probability $1-(wdn)^{-\Omega(1)}$.

Extracting D from the kernels (Steps 13 – 21): We analyse these steps for three separate cases. The analysis shows that if Z is the 'last' matrix then the algorithm succeeds with high probability, otherwise the test at step 17 fails with high probability.

Case a [$Z = C \cdot R_t \cdot D$]: From Observation 3.2, $\det(R'_j(\mathbf{a}_k))$ and $\det(R'_j(\mathbf{b}_k))$ are nonzero with high probability, for all $j \in [t-1]$ and $k \in [w+1]$. Assuming this, the following holds for all $k \in [w+1]$:

$$\begin{aligned} D \cdot \mathrm{Ker}(Z'(\mathbf{a}_k)) &= \mathrm{Ker}(F'_t(\mathbf{a}_k)) \ , \\ D \cdot \mathrm{Ker}(Z'(\mathbf{b}_k)) &= \mathrm{Ker}(F'_t(\mathbf{b}_k)) \ . \end{aligned} \tag{2}$$

Hence, at step 9, there are $\lambda_k, \rho_k \in \mathbb{L}^\times$ such that

$$D \cdot \mathbf{v}_k = \lambda_k \mathbf{u}_k, \quad D \cdot \mathbf{s}_k = \rho_k \mathbf{w}_k \qquad \text{for } k \in [w+1].$$

Step 13 also succeeds with high probability due to the following claim (proof in Appendix A).

Claim 3.1. *With probability $1-(wdn)^{-\Omega(1)}$, any subset of w vectors in any of the sets $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{w+1}\}$, $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{w+1}\}$, $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{w+1}\}$, or $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{w+1}\}$ are $\mathbb{L}$-linearly independent.*

At this step, $\mathbf{v}_{w+1} = \sum_{k=1}^{w} \beta_k \mathbf{v}_k$ and $\mathbf{s}_{w+1} = \sum_{k=1}^{w} \delta_k \mathbf{s}_k$, and so by applying D on both sides,

$$\lambda_{w+1}\mathbf{u}_{w+1} = \sum_{k=1}^{w} \beta_k \lambda_k \mathbf{u}_k, \quad \rho_{w+1}\mathbf{w}_{w+1} = \sum_{k=1}^{w} \delta_k \rho_k \mathbf{w}_k \ .$$

Also, $\mathbf{u}_{w+1} = \sum_{k=1}^{w} \alpha_k \mathbf{u}_k$ and $\mathbf{w}_{w+1} = \sum_{k=1}^{w} \gamma_k \mathbf{w}_k$. By Claim 3.1, none of the $\alpha_k, \beta_k, \gamma_k, \delta_k$ is zero and

$$\frac{\lambda_k}{\lambda_{w+1}} = \frac{\alpha_k}{\beta_k} \ , \quad \frac{\rho_k}{\rho_{w+1}} = \frac{\gamma_k}{\delta_k}, \qquad \text{for all } k \in [w].$$

From the construction of matrices U, V, W and S at step 15,

$$D \cdot V = \lambda_{w+1} U \ , \quad D \cdot S = \rho_{w+1} W,$$

and $U, V, W, S \in \mathsf{GL}(w, \mathbb{L})$ (by Claim 3.1). Therefore, $UV^{-1}SW^{-1}$ is a scalar matrix.

Case b [$Z^T = C \cdot R_t \cdot D$]: In this case, the check at step 17 fails with high probability. Suppose the algorithm passes steps 13 and 15, and reaches step 17. We show that $UV^{-1}SW^{-1}$ being a scalar matrix implies an event $\mathcal{E}$ that happens with a low probability. The event $\mathcal{E}$ can be derived as follows:

Let $M \stackrel{\text{def}}{=} U \cdot V^{-1}$, and $c \in \mathbb{L}^\times$ such that $M = cW \cdot S^{-1}$. Assuming the invertibility of $R'_j(\mathbf{a}_k)$ and $R'_j(\mathbf{b}_k)$ for $j \in [t-1]$ (Observation 3.2), and as in Equation 2, the following holds for all $k \in [w+1]$.

$$\begin{aligned} D \cdot \mathrm{Ker}((Z'(\mathbf{a}_k))^T) &= \mathrm{Ker}(F'_t(\mathbf{a}_k)) \,, \\ D \cdot \mathrm{Ker}((Z'(\mathbf{b}_k))^T) &= \mathrm{Ker}(F'_t(\mathbf{b}_k)) \,. \end{aligned}$$

By Observation 3.3, we can assume the above four kernels are one-dimensional. Hence, at step 9 there are $\mathbf{p}_k \in \mathrm{Ker}((Z'(\mathbf{a}_k))^T)$ and $\mathbf{q}_k \in \mathrm{Ker}((Z'(\mathbf{b}_k))^T)$ satisfying $D \cdot \mathbf{p}_k = \mathbf{u}_k$ and $D \cdot \mathbf{q}_k = \mathbf{w}_k$, for every $k \in [w+1]$. Consider the $w \times w$ matrices P and Q such that the k-th column of these matrices are $\frac{\alpha_k}{\beta_k}\mathbf{p}_k$ and $\frac{\gamma_k}{\delta_k}\mathbf{q}_k$ respectively, where $\alpha_k, \beta_k, \gamma_k, \delta_k$ are the constants computed at step 13. Clearly, $D \cdot P = U$ and $D \cdot Q = W$, where U, W are the matrices computed at step 15.

As $M = cW \cdot S^{-1}$ (by assumption), we have $D^{-1}MS = cD^{-1}W = cQ$. Hence, for $k \in [w]$,

$$D^{-1}M \cdot \mathbf{s}_k = \frac{c\gamma_k}{\delta_k}\mathbf{q}_k.$$

At step 13, $\mathbf{w}_{w+1} = \sum_{k=1}^{w} \gamma_k \mathbf{w}_k$ and $\mathbf{s}_{w+1} = \sum_{k=1}^{w} \delta_k \mathbf{s}_k$. Multiplying D^{-1} on both sides and $D^{-1}M$ on both sides of these two equations respectively,

$$\mathbf{q}_{w+1} = \sum_{k=1}^{w} \gamma_k \mathbf{q}_k, \quad \text{and} \quad D^{-1}M \cdot \mathbf{s}_{w+1} = \sum_{k=1}^{w} c\gamma_k \mathbf{q}_k \;.$$

$$\Rightarrow \quad D^{-1}M \cdot \mathbf{s}_{w+1} = c\mathbf{q}_{w+1}. \tag{3}$$

From Observation 3.3, there are $\lambda_1, \lambda_2 \in \mathbb{L}^\times$ such that

$$\begin{aligned} \mathbf{s}_{w+1} &= \lambda_1 \cdot (N_{11}(\mathbf{b}_{w+1}) \;\; N_{12}(\mathbf{b}_{w+1}) \;\; \ldots \;\; N_{1w}(\mathbf{b}_{w+1}))^T \,, \\ \mathbf{q}_{w+1} &= \lambda_2 \cdot (N_{11}(\mathbf{b}_{w+1}) \;\; N_{21}(\mathbf{b}_{w+1}) \;\; \ldots \;\; N_{w1}(\mathbf{b}_{w+1}))^T \,. \end{aligned}$$

Let $D^{-1}M = (m_{lk})_{l,k \in [w]}$. Using the above values of $\mathbf{s}_{w+1}$ and $\mathbf{q}_{w+1}$ in Equation 3 and restricting to the first two entries of the resulting column vectors, we have

$$\lambda_1 \left(\sum_{k=1}^{w} m_{1k} N_{1k}(\mathbf{b}_{w+1}) \right) = c\lambda_2 N_{11}(\mathbf{b}_{w+1}) \;\;, \quad \lambda_1 \left(\sum_{k=1}^{w} m_{2k} N_{1k}(\mathbf{b}_{w+1}) \right) = c\lambda_2 N_{21}(\mathbf{b}_{w+1}) \;\;.$$

Thus we get the following relation,

$$N_{21}(\mathbf{b}_{w+1}) \left(\sum_{k=1}^{w} m_{1k} N_{1k}(\mathbf{b}_{w+1}) \right) = N_{11}(\mathbf{b}_{w+1}) \left(\sum_{k=1}^{w} m_{2k} N_{1k}(\mathbf{b}_{w+1}) \right).$$

Event $\mathcal{E}$ is defined by the above equality, i.e. we say $\mathcal{E}$ has happened whenever the above equality holds. Now observe that $D^{-1}M$ is independent of the random bits used to choose $\mathbf{b}_{w+1}$, one way of seeing this is that $D^{-1}M$ is already fixed before $\mathbf{b}_{w+1}$ is chosen. Hence, it is sufficient to show that the above equality happens with low probability over the randomness of $\mathbf{b}_{w+1}$, for any arbitrarily fixed $m_{11}, \ldots, m_{1w}$ and $m_{21}, \ldots, m_{2w}$ from $\mathbb{L}$. Moreover, as $D^{-1}M$ is invertible, we can assume – not all in $\{m_{11}, \ldots, m_{1w}\}$ or $\{m_{21}, \ldots, m_{2w}\}$ are zero. The following observation and Schwartz-Zippel lemma complete the proof in this case.

Observation 3.4. *$N_{21}(\mathbf{z}) \left(\sum_{k=1}^{w} m_{1k} \cdot N_{1k}(\mathbf{z})\right) \neq N_{11}(\mathbf{z}) \left(\sum_{k=1}^{w} m_{2k} \cdot N_{1k}(\mathbf{z})\right)$ as polynomials in $\mathbb{F}[\mathbf{z}]$.*

Proof. Suppose the two sides are equal. As $N_{21}(\mathbf{z})$ and $N_{11}(\mathbf{z})$ are irreducible and coprime polynomials, $N_{21}(\mathbf{z})$ must divide $\sum_{k=1}^{w} m_{2k} \cdot N_{1k}(\mathbf{z})$. But the two polynomials have the same degree and they are monomial disjoint, thereby giving us a contradiction. □

Case c [$Z = C \cdot R_i \cdot D$ or $Z^T = C \cdot R_i \cdot D$ for some $i \in [t-1]$]: Assume $Z = C \cdot R_i \cdot D$ for some $i \in [t-1]$. The case $Z^T = C \cdot R_i \cdot D$ can be argued similarly. Similar to Case b, we show that if the algorithm passes steps 13 and 15, and reaches step 17 then $UV^{-1}SW^{-1}$ being a scalar matrix implies an event $\mathcal{E}$ that happens with very low probability. Hence, the check at step 17 fails with high probability. The event $\mathcal{E}$ can be derived as follows:

Let $M \stackrel{\text{def}}{=} U \cdot V^{-1}$, and $c \in \mathbb{L}^\times$ be such that $M = c \cdot WS^{-1}$. From the construction of W and S,

$$\frac{c\gamma_k}{\delta_k}\mathbf{w}_k = M \cdot \mathbf{s}_k \,, \quad \text{for all} \quad k \in [w],$$

where γ_k, δ_k are as computed at step 13. Since $\mathbf{w}_{w+1} = \sum_{k=1}^{w} \gamma_k \mathbf{w}_k$ and $\mathbf{s}_{w+1} = \sum_{k=1}^{w} \delta_k \cdot \mathbf{s}_k$,

$$c \cdot \mathbf{w}_{w+1} = M \cdot \mathbf{s}_{w+1}. \tag{4}$$

Let $H \stackrel{\text{def}}{=} D^{-1} \cdot R'_{i+1} \ldots R'_t$. From Observation 3.2, the following holds,

$$H^{-1} \cdot \text{Kernel}_{\mathbb{L}(\mathbf{x})}(Z') = \text{Kernel}_{\mathbb{L}(\mathbf{x})}(F'_t).$$

Let $\mathbf{n} = (N_{11}(\mathbf{b}_{w+1}) \quad N_{12}(\mathbf{b}_{w+1}) \quad \ldots \quad N_{1w}(\mathbf{b}_{w+1}))^T$. From Observation 3.3, and as $H(\mathbf{b}_{w+1})$ is invertible with high probability over the random choice of $\mathbf{b}_{w+1}$, there are $\lambda_1, \lambda_2 \in \mathbb{L}^\times$ such that

$$\begin{aligned} \mathbf{w}_{w+1} &= \lambda_1 H^{-1}(\mathbf{b}_{w+1}) \cdot \mathbf{n} \\ \mathbf{s}_{w+1} &= \lambda_2 \mathbf{n}. \end{aligned}$$

Substituting the above values of $\mathbf{w}_{w+1}$ and $\mathbf{s}_{w+1}$ in Equation 4, we have

$$c\lambda_1 H^{-1}(\mathbf{b}_{w+1}) \cdot \mathbf{n} = \lambda_2 M \cdot \mathbf{n} \,, \quad \Rightarrow \quad c\lambda_1 \mathbf{n} = \lambda_2 H(\mathbf{b}_{w+1}) \cdot M \cdot \mathbf{n}.$$

Let $H \cdot M = (h_{lk})_{l,k \in [w]}$. Restricting to the first two entries of the vectors in the above equality, and observing that M is independent of $\mathbf{b}_{w+1}$, we have

$$\begin{aligned} c\lambda_1 N_{11}(\mathbf{b}_{w+1}) &= \lambda_2 \left(\sum_{k=1}^{w} h_{1k}(\mathbf{b}_{w+1}) \cdot N_{1k}(\mathbf{b}_{w+1}) \right), \\ c\lambda_1 N_{12}(\mathbf{b}_{w+1}) &= \lambda_2 \left(\sum_{k=1}^{w} h_{2k}(\mathbf{b}_{w+1}) \cdot N_{1k}(\mathbf{b}_{w+1}) \right). \end{aligned}$$

Hence, we get the following relation

$$N_{11}(\mathbf{b}_{w+1}) \cdot \left(\sum_{k=1}^{w} h_{2k}(\mathbf{b}_{w+1}) \cdot N_{1k}(\mathbf{b}_{w+1}) \right) = N_{12}(\mathbf{b}_{w+1}) \cdot \left(\sum_{k=1}^{w} h_{1k}(\mathbf{b}_{w+1}) \cdot N_{1k}(\mathbf{b}_{w+1}) \right). \quad (5)$$

Event $\mathcal{E}$ is defined by the above equality, that is $\mathcal{E}$ happens if the above equality is satisfied. Observe that the entries of the matrix product $H \cdot M = (h_{lk})_{l,k \in [w]}$ are rational functions in $\mathbf{x}$ variables and are *independent* of the random bits used to choose $\mathbf{b}_{w+1}$. We show next the probability that the above equality holds is low over the randomness of $\mathbf{b}_{w+1}$.

So far we have used only the first two properties of a pure product F_t, i.e, every R_i is full rank and $\det(R_1), \ldots, \det(R_t)$ are mutually coprime. However, these two properties are not sufficient to ensure the uniqueness of the last matrix in the product (as mentioned in a remark after Theorem 1). In the following observation, we use the third property of a pure product which ensures the desired uniqueness of the last matrix.

Observation 3.5. *Let $n \geq 2w^2$. Then all the entries of $H \cdot M$ are nonzero polynomials after setting the variables in $\mathbf{z}_1 \stackrel{\text{def}}{=} \{z_{11}, z_{21}, z_{31}, \ldots, z_{w1}\}$ to zero.*

Proof. $H \cdot M = D^{-1} \cdot R'_{i+1} \ldots R'_t \cdot M = (h_{lk})_{l,k \in [w]}$. Recalling the substitution $z_{11} = \frac{-\sum_{k=2}^{w} z_{1k} N_{1k}}{N_{11}}$ at step 5, we observe that the rational function h_{lk} becomes a polynomial under the setting $z_{11} = z_{21} = \ldots = z_{w1} = 0$, the variable z_{11} does not even appear in h_{lk}. Let $Q_j = (R_j)_{\mathbf{z}_1=0}$. By observing $(R_j)_{\mathbf{z}_1=0} = (R'_j)_{\mathbf{z}_1=0}$, it follows that $(H \cdot M)_{\mathbf{z}_1=0} = D^{-1} \cdot Q_{i+1} \ldots Q_t \cdot M$. By the third property of a pure product, the entries of $Q_{i+1} \ldots Q_t$ are $\mathbb{L}$-linearly independent. Hence, none of the entries of $D^{-1} \cdot Q_{i+1} \ldots Q_t \cdot M$ is zero, as $M \in \mathsf{GL}(\mathbb{L}, w)$ whenever the algorithm passes step 15. □

Observation 3.6. $N_{11}(\mathbf{x}) \cdot (\sum_{k=1}^{w} h_{2k}(\mathbf{x}) N_{1k}(\mathbf{x})) \neq N_{12}(\mathbf{x}) \cdot (\sum_{k=1}^{w} h_{1k}(\mathbf{x}) N_{1k}(\mathbf{x}))$ *as rational functions in* $\mathbb{L}(\mathbf{x})$.

Proof. Suppose $N_{11}(\mathbf{x}) \cdot (\sum_{k=1}^{w} h_{2k}(\mathbf{x}) N_{1k}(\mathbf{x})) = N_{12}(\mathbf{x}) \cdot (\sum_{k=1}^{w} h_{1k}(\mathbf{x}) N_{1k}(\mathbf{x}))$. By substituting $\mathbf{z}_1 = 0$ in the equation, the R.H.S becomes zero whereas the L.H.S reduces to $N_{11}^2 \cdot (h_{21})_{\mathbf{z}_1=0}$, which is nonzero (by Observation 3.5). □

Noting that the degrees of the numerator and the denominator of h_{lk} are upper bounded by wd, we conclude that the equality in Equation 5 happens with a low probability over the randomness of $\mathbf{b}_{w+1}$.

In case c if $Z^T = C \cdot R_i \cdot D$ to begin with then the argument remains very similar except in Observation 3.5, the variables in the first row and column of Z (instead of just the first column) are substituted to zero.

4 Average-case ABP reconstruction: Proof of Theorem 2

The algorithm for average-case ABP reconstruction is presented in Algorithm 2, Section 1.4.2. The algorithm uses Algorithm 5 and Algorithm 6 during its execution – we present and analyze these two algorithms in the following subsections.

4.1 Computing the corner spaces

Let f be the polynomial computed by a random (w, d, n)-ABP $X_1 \cdot X_2 \dots X_d$ over $\mathbb{F}$, where $n \geq 4w^2$.

Lemma 4.1. *With probability $1 - (wdn)^{-\Omega(1)}$ over the randomness of f, the following holds: Let $\mathbb{K} \supseteq \mathbb{F}$ be any field and $f = 0 \mod \langle l_1, \dots, l_k \rangle$, where l_i's are linear forms in $\mathbb{K}[\mathbf{x}]$. Then $k \geq w$ and for $k = w$, the space $\text{span}_{\mathbb{K}}\{l_1, \dots, l_w\}$ equals the $\mathbb{K}$-span of either the linear forms in X_1 or the linear forms in X_d.*

The above uniqueness of the corner spaces, $\mathcal{X}_1$ and $\mathcal{X}_d$ (defined in Section 1.4.2), helps compute them in Algorithm 5. The proof of the lemma is given at the end of this subsection.

Canonical bases of $\mathcal{X}_1$ and $\mathcal{X}_d$: For a set of variables $\mathbf{y} \subseteq \mathbf{x}$ and a linear form g in $\mathbb{F}[\mathbf{x}]$, define $g(\mathbf{y}) \stackrel{\text{def}}{=} g_{\mathbf{x}\setminus\mathbf{y}=0}$. We say $g(\mathbf{y})$ is the linear form g projected to the $\mathbf{y}$ variables. Let $x_1, \dots, x_w$ and v be a designated set of $w+1$ variables in $\mathbf{x}$, and $\mathbf{u} = \mathbf{x} \setminus \{x_1, \dots, x_w, v\}$. With $n \geq 4w^2$, a random (w, d, n)-ABP $X_1 \cdot X_2 \dots X_d$ satisfies the following condition with probability $1 - (wdn)^{-\Omega(1)}$:

(*a) The linear forms in X_1 (similarly, X_d) projected to $x_1, \dots, x_w$ are $\mathbb{F}$-linearly independent.

If the above condition is satisfied then there is a $C \in \mathsf{GL}(w, \mathbb{F})$ such that the linear forms in $X_1 \cdot C$ are of the kind:

$$x_i - \alpha_i v - g_i(\mathbf{u}), \quad \text{for } i \in [w], \tag{6}$$

where each $\alpha_i \in \mathbb{F}$ and g_i is a linear form in $\mathbb{F}[\mathbf{u}]$. Thus, we can assume without loss of generality, the linear forms in X_1 are of the above kind. Similarly, the linear forms in X_d are also of the kind:

$$x_i - \beta_i v - h_i(\mathbf{u}), \quad \text{for } i \in [w], \tag{7}$$

where each $\beta_i \in \mathbb{F}$ and h_i is a linear form in $\mathbb{F}[\mathbf{u}]$. Moreover, with probability $1 - (wdn)^{-\Omega(1)}$ over the randomness of the ABP, the following condition is satisfied:

(*b) $\alpha_1, \dots, \alpha_w$ and $\beta_1, \dots, \beta_w$ are distinct elements in $\mathbb{F}$.

The task at hand for Algorithm 5 is to solve for α_i, g_i and β_j, h_j, for $i, j \in [w]$, assuming that conditions **(*a)** and **(*b)** are satisfied. The bases defined by Equations 6 and 7 are canonical for $\mathcal{X}_1$ and $\mathcal{X}_d$.

We analyze the three main steps of Algorithm 5 next:

1. *Partitioning the variables (Step 2)*: The only thing to note here is, if $n - (w+1)$ is not divisible by $4w^2 - (w+1)$ then we allow the last two sets $\mathbf{u}_{m-1}$ and $\mathbf{u}_m$ to overlap – the algorithm can be suitably adjusted in this case.

2. *Reduction to solving systems of polynomial equations (Steps 5–13)*: At step 7, the task of computing $(\alpha_1, \dots, \alpha_w, g_1(\mathbf{u}_\ell), \dots, g_w(\mathbf{u}_\ell))$ such that

$$f_\ell = 0 \mod \langle x_1 - \alpha_1 v - g_1(\mathbf{u}_\ell), \dots, x_w - \alpha_w v - g_w(\mathbf{u}_\ell) \rangle,$$

can be reduced to solving for all $\mathbb{F}$-rational points of a system of polynomial equations over $\mathbb{F}$ as follows: Treat $\alpha_1, \dots, \alpha_w$ and the $4w^3 - w(w+1)$ coefficients of $g_1(\mathbf{u}_\ell), \dots, g_w(\mathbf{u}_\ell)$, say

Algorithm 5 Computing the corner spaces

INPUT: Blackbox access to a f computed by a random (w, d, n)-ABP.
OUTPUT: Bases of the two corner spaces $\mathcal{X}_1$ and $\mathcal{X}_d$ modulo which f is zero.

1. /* Partitioning the variables */
2. Choose $w+1$ designated variables $x_1, x_2, \ldots, x_w, v$, and let $\mathbf{u} = \mathbf{x} \setminus \{x_1, \ldots, x_w, v\}$. Partition $\mathbf{u}$ into sets $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$, each of size $4w^2 - (w+1)$. .
3.
4. /* Reduction to solving m systems of polynomial equations */
5. **for** $\ell = 1$ **to** m **do**
6.   Set $f_\ell = f_{\mathbf{u} \setminus \mathbf{u}_\ell = 0}$.
7.   Solve for *all possible* $(\alpha_1, \ldots, \alpha_w, g_1(\mathbf{u}_\ell), \ldots, g_w(\mathbf{u}_\ell))$, where each $\alpha_i \in \mathbb{F}$ and $g_i(\mathbf{u}_\ell)$ is a linear form in $\mathbb{F}[\mathbf{u}_\ell]$ such that
$$f_\ell = 0 \mod \langle x_1 - \alpha_1 v - g_1(\mathbf{u}_\ell), \ldots, x_w - \alpha_w v - g_w(\mathbf{u}_\ell)\rangle.$$
8.   **if** Step 7 does not return *exactly two* solutions for $(\alpha_1, \ldots, \alpha_w, g_1(\mathbf{u}_\ell), \ldots, g_w(\mathbf{u}_\ell))$ **then**
9.   Output 'Failed'.
10.   **else**
11.   The solutions be $(\alpha_{\ell 1}, \ldots, \alpha_{\ell w}, g_1(\mathbf{u}_\ell), \ldots, g_w(\mathbf{u}_\ell))$ and $(\beta_{\ell 1}, \ldots, \beta_{\ell w}, h_1(\mathbf{u}_\ell), \ldots, h_w(\mathbf{u}_\ell))$.
12.   **end if**
13. **end for**
14.
15. /* Combining the solutions */
16. **if** $|\cup_{\ell \in [m]} \{(\alpha_{\ell 1}, \ldots, \alpha_{\ell w}), (\beta_{\ell 1}, \ldots, \beta_{\ell w})\}| \neq 2$ **then**
17.   Output 'Failed'.
18. **else**
19.   Without loss of generality, $(\alpha_{\ell 1}, \ldots, \alpha_{\ell w}) = (\alpha_1, \ldots, \alpha_w)$ and $(\beta_{\ell 1}, \ldots, \beta_{\ell w}) = (\beta_1, \ldots, \beta_w)$ for every $\ell \in [m]$. Set $g_i(\mathbf{u}) = \sum_{\ell \in [w]} g_i(\mathbf{u}_\ell)$ and $h_i(\mathbf{u}) = \sum_{\ell \in [w]} h_i(\mathbf{u}_\ell)$ for every $i \in [w]$.
20.   Return $\{x_i - \alpha_i v - g_i(\mathbf{u})\}_{i \in [w]}$ and $\{x_i - \beta_i v - h_i(\mathbf{u})\}_{i \in [w]}$ as the bases of $\mathcal{X}_1$ and $\mathcal{X}_d$.
21. **end if**

$\mathbf{w}$, as formal variables. Substitute $x_i = \alpha_i v + g_i(\mathbf{u}_\ell)$ for every $i \in [w]$ in the blackbox for f_ℓ, and interpolate the resulting polynomial p in the variables $\alpha_1, \ldots, \alpha_w, \mathbf{w}, v, \mathbf{u}_\ell$ with coefficients in $\mathbb{F}$. The interpolation, which can be done in $(d^{w^3} \log q)^{O(1)}$ time, gives p in dense representation (i.e. as a sum of monomials). As the individual degrees of the variables in p are bounded by d, we only need $|\mathbb{F}| > d$ to carry out this interpolation. Now by treating p as a polynomial in the variables $v, \mathbf{u}_\ell$ with coefficients in $\mathbb{F}(\alpha_1, \ldots, \alpha_w, \mathbf{w})$, and equating these coefficients to zero, we get a system of $d^{O(w^2)}$ polynomial equations in $O(w^3)$ variables with degree of each polynomial equation bounded by d. By Lemma 4.1, such a system has exactly two solutions over $\overline{\mathbb{F}}$ and moreover, these two solution points are $\mathbb{F}$-rational. Hence, by applying Lemma 2.2, we can compute the two solutions for $(\alpha_1, \ldots, \alpha_w, \mathbf{w})$ at step 7, in $(d^{w^3} \log q)^{O(1)}$ time.

3. *Combining the solutions (Steps 16–21):* The correctness of the steps follows from condition (*b).

Uniqueness of the corner spaces: Proof of Lemma 4.1

As $n \geq 4w^2$, a random (w, d, n)-ABP $X_1 \cdots X_d$ satisfies the following condition with probability $1 - (wdn)^{-\Omega(1)}$:

(**) For every choice of three (or less) matrices among $X_2, X_3, \ldots, X_{d-1}$, the linear forms in these matrices and X_1 and X_d are $\mathbb{F}$-linearly independent.

So, it is sufficient to prove the following restatement of the lemma.

Lemma 4.1. *Suppose f is computed by a (w, d, n)-ABP $X_1 \cdot X_2 \cdots X_d$ satisfying the above condition (**). If $f = 0 \mod \langle l_1, \ldots, l_k \rangle$, where l_i's are linear forms over $\mathbb{K} \supseteq \mathbb{F}$, then $k \geq w$ and for $k = w$, the space* $\mathrm{span}_{\mathbb{K}}\{l_1, \ldots, l_w\}$ *equals the $\mathbb{K}$-span of either the linear forms in X_1 or the linear forms in X_d.*

We prove the lemma first for $d = 3$, and then use this case to prove it for $d > 3$.

Case $[d = 3]$: There is an $A \in \mathsf{GL}(n, \mathbb{F})$ such that $f(A \cdot \mathbf{x})$ is computed by $(y_1\ y_2\ \ldots y_w) \cdot (r_{ij})_{i,j \in [w]} \cdot (z_1\ z_2\ \ldots z_w)^T$, where $\mathbf{y} = \{y_i\}_{i \in [w]}$, $\mathbf{r} = \{r_{ij}\}_{i,j \in [w]}$ and $\mathbf{z} = \{z_j\}_{j \in [w]}$ are distinct variables in $\mathbf{x}$. If $f = 0 \mod \langle l_1, \ldots, l_k \rangle$, then $f(A \cdot \mathbf{x}) = 0 \mod \langle l_1(A \cdot \mathbf{x}), \ldots, l_k(A \cdot \mathbf{x}) \rangle$. Next, we show that if $f(A \cdot \mathbf{x}) = 0$ modulo k' linear forms $h_1, \ldots, h_{k'} \in \mathbb{K}[\mathbf{y} \uplus \mathbf{z} \uplus \mathbf{r}]$ then $k' \geq w$, and for $k' = w$, the space $\mathrm{span}_{\mathbb{K}}\{h_1, \ldots, h_w\}$ equals either $\mathrm{span}_{\mathbb{K}}\{y_1, \ldots, y_w\}$ or $\mathrm{span}_{\mathbb{K}}\{z_1, \ldots, z_w\}$. It follows that $k \geq k' \geq w$, and for $k = w$, the linear forms $l_1(A \cdot \mathbf{x}), \ldots, l_w(A \cdot \mathbf{x})$ must belong to $\mathbb{K}[\mathbf{y} \uplus \mathbf{z} \uplus \mathbf{r}]$ (otherwise, we will have $f(A \cdot \mathbf{x}) = 0$ modulo less than w linear forms in $\mathbb{K}[\mathbf{y} \uplus \mathbf{z} \uplus \mathbf{r}]$), and hence $\mathrm{span}_{\mathbb{K}}\{l_1, \ldots, l_w\}$ equals the $\mathbb{K}$-span of either the linear forms in X_1 or the linear forms in X_d.

Reusing symbols, assume that f is computed by $X_1 \cdot X_2 \cdot X_3$, where $X_1 = (y_1\ y_2\ \ldots y_w)$, $X_2 = (r_{ij})_{i,j \in [w]}$ and $X_3 = (z_1\ z_2\ \ldots z_w)^T$, and $f = 0 \mod \langle l_1, \ldots, l_k \rangle$, where l_i's are linear forms in $\mathbb{K}[\mathbf{y} \uplus \mathbf{z} \uplus \mathbf{r}]$. Suppose $k \leq w$; otherwise, we have nothing to prove. Consider the reduced Gröbner basis[11] G of the ideal $\langle l_1, \ldots, l_k \rangle$ with respect to the lexicographic monomial ordering defined by

[11] See [CLO07]. Equivalently, think of the set of linear forms obtained from a reduced row echelon form of the coefficient matrix of $l_1, \ldots, l_k$.

$\mathbf{y} \succ \mathbf{z} \succ \mathbf{r}$. There are sets $S_{\mathbf{y}}, S_{\mathbf{z}} \subseteq [w]$ and $S_{\mathbf{r}} \subseteq [w] \times [w]$, satisfying $|S_{\mathbf{y}}| + |S_{\mathbf{z}}| + |S_{\mathbf{r}}| \leq k$, such that G consists of linear forms of the kind:

$$\begin{aligned} y_i - g_i(\mathbf{y}, \mathbf{z}, \mathbf{r}) & \qquad \text{for } i \in S_{\mathbf{y}}, \\ z_j - h_j(\mathbf{z}, \mathbf{r}) & \qquad \text{for } j \in S_{\mathbf{z}}, \\ r_{\ell e} - p_{\ell e}(\mathbf{r}) & \qquad \text{for } (\ell, e) \in S_{\mathbf{r}}, \end{aligned}$$

where g_i, h_j and $p_{\ell e}$ are linear forms over $\mathbb{K}$ in their respective sets of variables. Let X_1', X_2', X_3' be the linear matrices obtained from X_1, X_2, X_3 respectively, by replacing y_i by $g_i(\mathbf{y}, \mathbf{z}, \mathbf{r})$, $r_{\ell e}$ by $p_{\ell e}(\mathbf{r})$ and z_j by $h_j(\mathbf{z}, \mathbf{r})$, for $i \in S_{\mathbf{y}}$, $(\ell, e) \in S_{\mathbf{r}}$ and $j \in S_{\mathbf{z}}$. Then,

$$X_1' \cdot X_2' \cdot X_3' = 0. \tag{8}$$

The dimension of the $\mathbb{K}$-span of the linear forms of X_1' is at least $(w - |S_{\mathbf{y}}|)$, that of X_2' is at least $(w^2 - |S_{\mathbf{r}}|)$, and of X_3' is at least $(w - |S_{\mathbf{z}}|)$. Also, there are $C, D \in \mathsf{GL}(w, \mathbb{K})$ such that $X_1' \cdot C, D \cdot X_3'$ are obtained (via row and column operations on X_1' and X_3', respectively) from X_1, X_3 respectively, by replacing y_i by $g_i(0, \mathbf{z}, \mathbf{r})$ and z_j by $h_j(0, \mathbf{r})$, for $i \in S_{\mathbf{y}}$ and $j \in S_{\mathbf{z}}$. Consider the following equation,

$$(X_1'C) \cdot (C^{-1} X_2' D^{-1}) \cdot (D X_3') = 0. \tag{9}$$

By examining the L.H.S, we can conclude that for $s \in [w] \setminus S_{\mathbf{y}}$ and $t \in [w] \setminus S_{\mathbf{z}}$, the coefficient of the monomial $y_s z_t$ over $\mathbb{K}(\mathbf{r})$ is the (s,t)-th entry of $C^{-1} X_2' D^{-1}$ which must be zero. Hence, the dimension of the $\mathbb{K}$-span of the linear forms in $C^{-1} X_2' D^{-1}$ is at most $w^2 - (w - |S_{\mathbf{y}}|)(w - |S_{\mathbf{z}}|)$. As the dimension of the $\mathbb{K}$-span of the linear forms in X_2' remains unaltered under left and right multiplications by elements in $\mathsf{GL}(w, \mathbb{K})$, we get the relation

$$\begin{aligned} w^2 - |S_{\mathbf{r}}| &\leq w^2 - (w - |S_{\mathbf{y}}|)(w - |S_{\mathbf{z}}|) \\ \Rightarrow (w - |S_{\mathbf{y}}|)(w - |S_{\mathbf{z}}|) &\leq |S_{\mathbf{r}}| \\ \Rightarrow w^2 - (|S_{\mathbf{y}}| + |S_{\mathbf{z}}|)w + |S_{\mathbf{y}}| \cdot |S_{\mathbf{z}}| &\leq |S_{\mathbf{r}}| \\ \Rightarrow w^2 - (w - |S_{\mathbf{r}}|)w + |S_{\mathbf{y}}| \cdot |S_{\mathbf{z}}| &\leq |S_{\mathbf{r}}|, \quad \text{as } |S_{\mathbf{y}}| + |S_{\mathbf{z}}| + |S_{\mathbf{r}}| \leq k \leq w \\ \Rightarrow |S_{\mathbf{r}}| w + |S_{\mathbf{y}}| \cdot |S_{\mathbf{z}}| &\leq |S_{\mathbf{r}}|. \end{aligned}$$

As $|S_{\mathbf{y}}|, |S_{\mathbf{z}}|, |S_{\mathbf{r}}| \geq 0$, we must have $|S_{\mathbf{r}}| = 0$, and either $|S_{\mathbf{y}}| = 0$ or $|S_{\mathbf{z}}| = 0$.

Suppose $|S_{\mathbf{r}}| = |S_{\mathbf{z}}| = 0$ (the case for $|S_{\mathbf{r}}| = |S_{\mathbf{y}}| = 0$ is similar). Then, Equation 9 simplifies to

$$(X_1'C) \cdot (C^{-1} X_2) \cdot X_3 = 0.$$

If $k < w$ then there is a y_s in X_1 that is not replaced while forming $X_1'C$ from X_1. By examining the coefficient of y_s over $\mathbb{K}(\mathbf{r}, \mathbf{z})$ in the L.H.S of the above equation, we arrive at a contradiction. Hence, $k = w$, in which case Equation 8 simplifies to

$$X_1' \cdot X_2 \cdot X_3 = 0.$$

The entries of X_1' are linear forms in $\mathbf{z}$ and $\mathbf{r}$, and so $X_1' = X_1'(\mathbf{z}) + X_1'(\mathbf{r})$ where the entries of $X_1'(\mathbf{z})$ (similarly, $X_1'(\mathbf{r})$) are linear forms in $\mathbf{z}$ (respectively, $\mathbf{r}$). The above equation implies

$$X_1'(\mathbf{z}) \cdot X_2 \cdot X_3 = 0 \quad \text{and} \quad X_1'(\mathbf{r}) \cdot X_2 \cdot X_3 = 0,$$

as the two L.H.S above are monomial disjoint. It is now easy to argue that $X_1'(\mathbf{z}) = X_1'(\mathbf{r}) = 0$, implying $X_1' = 0$ and hence the reduced Gröbner basis G is in fact $\{y_1, \ldots, y_w\}$.

Case [$d > 3$]: As before, by applying an invertible transformation, we can assume that $X_1 = (y_1\ y_2\ \ldots\ y_w)$, $X_2 = (r_{ij})_{i,j \in [w]}$ and $X_d = (z_1\ z_2\ \ldots\ z_w)^T$. Let $\mathbf{u} = \mathbf{x} \setminus (\mathbf{y} \uplus \mathbf{z} \uplus \mathbf{r})$ and $k \leq w$. Consider the reduced Gröbner basis G of the ideal $\langle l_1, l_2, \ldots, l_k \rangle$ with respect to the lexicographic monomial ordering defined by $\mathbf{u} \succ \mathbf{y} \succ \mathbf{z} \succ \mathbf{r}$. There are sets $S_\mathbf{u} \subseteq [n - w^2 - 2w]$, $S_\mathbf{y}, S_\mathbf{z} \subseteq [w]$ and $S_\mathbf{r} \subseteq [w^2]$, satisfying $|S_\mathbf{u}| + |S_\mathbf{y}| + |S_\mathbf{z}| + |S_\mathbf{r}| \leq k$, such that G consists of linear forms of the kind:

$$\begin{aligned} u_m - t_m(\mathbf{u}, \mathbf{y}, \mathbf{z}, \mathbf{r}) &\qquad \text{for } m \in S_\mathbf{u}, \\ y_i - g_i(\mathbf{y}, \mathbf{z}, \mathbf{r}) &\qquad \text{for } i \in S_\mathbf{y}, \\ z_j - h_j(\mathbf{z}, \mathbf{r}) &\qquad \text{for } j \in S_\mathbf{z}, \\ r_{\ell e} - p_{\ell e}(\mathbf{r}) &\qquad \text{for } (\ell, e) \in S_\mathbf{r}, \end{aligned}$$

where t_m, g_i, h_j and $p_{\ell e}$ are linear forms over $\mathbb{K}$ in their respective sets of variables. Let X' be the matrix obtained from X by replacing u_m by $t_m(\mathbf{u}, \mathbf{y}, \mathbf{z}, \mathbf{r})$, y_i by $g_i(\mathbf{y}, \mathbf{z}, \mathbf{r})$, z_j by $h_j(\mathbf{z}, \mathbf{r})$, and $r_{\ell e}$ by $p_{\ell e}(\mathbf{r})$, for $m \in S_\mathbf{u}$, $i \in S_\mathbf{y}$, $j \in S_\mathbf{z}$, and $(\ell, e) \in S_\mathbf{r}$. Then,

$$X_1' \cdot X_2' \cdot X_3' \ldots X_d' = 0.$$

Let $X(\mathbf{u}) \stackrel{\text{def}}{=} (X)_{\mathbf{y}=\mathbf{z}=\mathbf{r}=0}$. By treating the L.H.S of the above equation as a polynomial in $\mathbf{u}$-variables with coefficients from $\mathbb{K}(\mathbf{y}, \mathbf{z}, \mathbf{r})$ and focusing on the degree-$(d-3)$ homogeneous component of this polynomial, we have

$$X_1' \cdot X_2' \cdot X_3'(\mathbf{u}) \ldots X_{d-1}'(\mathbf{u}) \cdot X_d' = 0. \tag{10}$$

If $X_3'(\mathbf{u}) \cdots X_{d-1}'(\mathbf{u}) \in \mathsf{GL}(w, \mathbb{K}(\mathbf{u}))$ then there is a $\mathbf{c} \in \mathbb{F}^{|\mathbf{u}|}$ such that $C = X_3'(\mathbf{c}) \cdots X_{d-1}'(\mathbf{c}) \in \mathsf{GL}(w, \mathbb{K})$. Define

$$f_1 = X_1 \cdot X_2 \cdot C \cdot X_d,$$

and observe that Equation 10 implies f_1 is zero modulo the linear forms,

$$\begin{aligned} y_i - g_i(\mathbf{y}, \mathbf{z}, \mathbf{r}) &\qquad \text{for } i \in S_\mathbf{y}, \\ z_j - h_j(\mathbf{z}, \mathbf{r}) &\qquad \text{for } j \in S_\mathbf{z}, \\ r_{\ell e} - p_{\ell e}(\mathbf{r}) &\qquad \text{for } (\ell, e) \in S_\mathbf{r}. \end{aligned}$$

By applying Case [d=3] on f_1, we get the desired conclusion, i.e. $k = w$ and the $\mathbb{K}$-span of the above linear forms (hence also that of $\{l_1, \ldots, l_k\}$) is either $\text{span}_\mathbb{K}\{y_1, \ldots, y_w\}$ or $\text{span}_\mathbb{K}\{z_1, \ldots, z_w\}$. So, suppose $X_3'(\mathbf{u}) \cdots X_{d-1}'(\mathbf{u}) \notin \mathsf{GL}(w, \mathbb{K}(\mathbf{u}))$ in Equation 10. Then, there is a $j \in [3, d-1]$ such that $\det(X_j'(\mathbf{u})) = 0$. Observe that $X_i'(\mathbf{u})$ can be obtained from $X_i(\mathbf{u})$ by replacing u_m by $t_m(\mathbf{u}, 0, 0, 0)$ for $m \in S_\mathbf{u}$. That is,

$$X_i'(\mathbf{u}) = X_i(\mathbf{u}) \mod \langle \{u_m - t_m(\mathbf{u}, 0, 0, 0)\}_{m \in S_\mathbf{u}} \rangle, \qquad \text{for every } i \in [3, d-1].$$

As $X_j(\mathbf{u})$ is full rank (which follows from condition (**)) and $\det(X_j'(\mathbf{u})) = 0$, the fact below implies $|S_\mathbf{u}| = w$, $|S_\mathbf{y}| = |S_\mathbf{z}| = |S_\mathbf{r}| = 0$.

Observation 4.1. *If the symbolic determinant* Det_w *is zero modulo* s *linear forms then* $s \geq w$.

Hence, Equation 10 simplifies to

$$
\begin{aligned}
X_1 \cdot X_2 \cdot X_3'(\mathbf{u}) \ldots X_{d-1}'(\mathbf{u}) \cdot X_d &= 0, \\
\Rightarrow X_3'(\mathbf{u}) \cdots X_{d-1}'(\mathbf{u}) &= 0. \qquad (11)
\end{aligned}
$$

The above equality can not happen and this can be argued by applying induction on the number of matrices in the L.H.S of Equation 11:

Base case: ($d = 4$) The L.H.S of Equation 11 has one matrix $X_3'(\mathbf{u})$. As $X_3(\mathbf{u})$ is full rank (by condition (**)), it cannot vanish modulo w linear forms.

Induction hypothesis: Equation 11 does not hold if the L.H.S has at most $d - 4$ matrices.

Inductive step: ($d > 4$) Suppose Equation 11 is true. As the $2w^2$ linear forms in $X_3(\mathbf{u})$ and $X_{d-1}(\mathbf{u})$ are linearly independent (condition (**) again), by Observation 4.1, at least one of $X_3'(\mathbf{u})$ and $X_{d-1}'(\mathbf{u})$ is invertible. This gives a shorter product where we can apply the induction hypothesis to get a contradiction.

4.2 Finding the coefficients in the intermediate matrices

Following the notations in Section 1.4.2, $\mathbf{y} = \{y_1, \ldots, y_w\}$ and $\mathbf{z} = \{z_1, \ldots, z_w\}$ are subsets of $\mathbf{x}$, $\mathbf{r} = \mathbf{x} \setminus (\mathbf{y} \uplus \mathbf{z})$, $X_1' = (y_1 \; y_2 \; \ldots \; y_w)$ and $X_d' = (z_1 \; z_2 \; \ldots \; z_w)^T$. When Algorithm 2 reaches the third and final stage, it has blackbox access to a $f' \in \mathbb{F}[\mathbf{x}]$ and linear matrices $S_2, \ldots, S_{d-1} \in \mathbb{L}[\mathbf{r}]^{w \times w}$ returned by Algorithm 1, such that $S_2 \cdot S_3 \ldots S_{d-1}$ is the linear matrix factorization of a random $(w, d-2, n-2w)$-matrix product $R_2 \cdot R_3 \ldots R_{d-1}$ over $\mathbb{F}$. Further, there exist linear matrices $T_2, \ldots, T_{d-1} \in \mathbb{L}[\mathbf{x}]^{w \times w}$ satisfying $(T_k)_{\mathbf{y}=0,\mathbf{z}=0} = S_k$ for every $k \in [2, d-1]$, such that f' is computed by the ABP $X_1' \cdot T_2 \ldots T_{d-1} \cdot X_{d-1}'$. The task for Algorithm 6 is to efficiently compute the coefficients of the $\mathbf{y}$ and $\mathbf{z}$ variables in T_k. At a high level, this is made possible because of the uniqueness of such T_k matrices: Indeed the analysis of Algorithm 6 shows that with high probability the coefficients of $\mathbf{y}$ and $\mathbf{z}$ in $T_3, \ldots, T_{d-2}$ are uniquely determined, and (if a certain canonical form is assumed then) the same is true for matrices T_2 and T_{d-1}.

Canonical form for T_2 and T_{d-1}: Matrix T_2 is said to be in canonical form if for every $l \in [w]$ the coefficient of y_l is zero in the linear form at the (i, j)-th entry of T_2, whenever $i > l$. Similarly, T_{d-1} is in canonical form if for every $l \in [w]$ the coefficient of z_l is zero in the linear form at the (i, j)-th entry of T_{d-1} whenever $j > l$. It can be verified (see [KNST17]), if f' is computed by an ABP $X_1' \cdot T_2 \ldots T_{d-1} \cdot X_{d-1}'$ then it is computed by another ABP where the corresponding T_2 and T_{d-1} are in canonical form, and the other matrices remain unchanged.

Linear independence of minors of a random ABP: The lemma given below is the reason Algorithm 6 is able to reduce the task of finding the coefficients of the $\mathbf{y}$ and $\mathbf{z}$ variables to solving linear equations. In the following discussion, the i-th row and j-th column of a matrix M will be denoted by $M(i, *)$ and $M(*, j)$ respectively.

Let $R_2 \cdot R_3 \ldots R_{d-1}$ be a random $(w, d-2, n-2w)$-matrix product in $\mathbf{r}$-variables over $\mathbb{F}$. For every $s, t \in [w]$, $R_2(s, *) \cdot R_3 \ldots R_{d-2} \cdot R_{d-1}(*, t)$ is a random $(w, d-2, n-2w)$-ABP having a total of $w^2(d-4) + 2w$ linear forms in all the R_k matrices. Let us index the linear forms arbitrarily by $[w^2(d-4) + 2w]$. We associate a polynomial $g_e^{(s,t)}$ with the e-th linear form, for every $e \in [w^2(d -$

$4) + 2w]$, as follows: If the e-th linear form is the (ℓ, m)-th entry of R_k then

$$g_e^{(s,t)}(\mathbf{r}) \stackrel{\text{def}}{=} [R_2(s,*) \cdot R_3 \ldots R_{k-2} \cdot R_{k-1}(*,\ell)] \cdot [R_{k+1}(m,*) \cdot R_{k+2} \ldots R_{d-2} \cdot R_{d-1}(*,t)],$$

by identifying the 1×1 matrix of the R.H.S with the entry of the matrix. The polynomials $\{g_e^{(s,t)} : e \in [w^2(d-4)+2w]\}$, will be called the *minors* of the ABP $R_2(s,*) \cdot R_3 \ldots R_{d-2} \cdot R_{d-1}(*,t)$.

Lemma 4.2. *With probability $1 - (wdn)^{-\Omega(1)}$ over the randomness of $R_2 \cdots R_{d-1}$ the following holds: For every $s, t \in [w]$, the minors $\{g_e^{(s,t)} : e \in [w^2(d-4)+2w]\}$, are $\mathbb{F}$-linearly independent.*

The proof of the lemma is given at the end of this section. Due to the uniqueness of factorization, the matrices $S_2, \ldots, S_{d-1}$ in Algorithm 2 are related to $R_2, \ldots, R_{d-1}$ as follows: There are $C_i, D_i \in \mathsf{GL}(w, \mathbb{L})$ such that $S_i = C_i \cdot R_i \cdot D_i$, for every $i \in [2, d-1]$; moreover, there are $c_2, \ldots, c_{d-2} \in \mathbb{L}^\times$ satisfying $C_2 = D_{d-1} = I_w$, $D_i \cdot C_{i+1} = c_i I_w$ for $i \in [2, d-2]$, and $\prod_{i=2}^{d-2} c_i = 1$. Define minors of the ABP $S_2(s,*) \cdot S_3 \ldots S_{d-2} \cdot S_{d-1}(*,t)$, for every $s, t \in [w]$, like above. The edges of the ABP are indexed by $[w^2(d-4)+2w]$ and a polynomial $h_e^{(s,t)}$ is associated with the e-th linear form as follows: If the e-th linear form is the (ℓ, m)-th entry of S_k then

$$h_e^{(s,t)}(\mathbf{r}) \stackrel{\text{def}}{=} [S_2(s,*) \cdot S_3 \ldots S_{k-2} \cdot S_{k-1}(*,\ell)] \cdot [S_{k+1}(m,*) \cdot S_{k+2} \ldots S_{d-2} \cdot S_{d-1}(*,t)]. \tag{12}$$

It is a simple exercise to derive the following corollary from the lemma above.

Corollary 4.1. *With probability $1 - (wdn)^{-\Omega(1)}$ the following holds: For every $s, t \in [w]$, the minors $\{h_e^{(s,t)} : e \in [w^2(d-4)+2w]\}$ are $\mathbb{L}$-linearly independent.*

We are now ready to argue the correctness of Algorithm 6 by tracing its steps.

1. *Computing the partial derivatives (Step 2)*: In this step, we compute all the third order partial derivatives of f' using Claim 2.1.

2. *Computing almost all the coefficients of the* $\mathbf{y}$ *and* $\mathbf{z}$ *variables (Steps 6–13)*: Equations 13 and 14 are justified by treating f' as a polynomial in the $\mathbf{y}$ and $\mathbf{z}$ variables with coefficients from $\mathbb{L}(\mathbf{r})$, and examining the coefficients of $y_s^2 z_t$ and $y_s z_t^2$ respectively. A linear system obtained at step 9 or step 11 has $w^2(d-4)+2w$ variables and the same number of linear equations. Corollary 4.1, together with Claim 2.2, ensure that the square coefficient matrix of the linear system is invertible (with high probability), and hence the solution computed is unique. The uniqueness implies that the solutions obtained across multiple iterations of the loop do not conflict with each other. For instance, the coefficients of y_s in the linear forms in $T_2(s,*), T_3, \ldots, T_{d-2}$ get computed repeatedly at step 9 for every value of $t \in [w]$ – uniqueness ensures that we always get the same values for these coefficients. This also shows that the matrices $T_3, \ldots, T_4$ are unique. By the end of this stage, the coefficients of $\mathbf{y}$ and $\mathbf{z}$ variables are computed for all the linear forms, except for the coefficients of y_l in $T_2(s,*)$ for $l > s$, and the coefficients of z_l in $T_{d-1}(*,t)$ for $l > t$. These coefficients are retrieved in the next stage.

3. *Computing the remaining* $\mathbf{y}$ *and* $\mathbf{z}$ *coefficients in* T_2 *and* T_{d-1} *(Steps 16–19)*: For an $s \in [w]$, consider the following minors of $S_2(s,*) \cdot S_3 \ldots S_{d-2} \cdot S_{d-1}(*,1)$:

$$S_3(m,*) \cdot S_4 \ldots S_{d-2} \cdot S_{d-1}(*,1) \qquad \text{for all } m \in [w].$$

Algorithm 6 Computing the coefficients of $\mathbf{y}$ and $\mathbf{z}$ variables in T_k

INPUT: Blackbox access to f' and linear matrices $S_2, \ldots, S_{d-1} \in \mathbb{L}[\mathbf{r}]^{w \times w}$.

OUTPUT: Linear matrices $T_2, T_3, \ldots, T_{d-1} \in \mathbb{L}[\mathbf{x}]^{w \times w}$ such that f' is computed by $\mathbf{y} \cdot T_2 \cdot T_3 \ldots T_{d-1} \cdot \mathbf{z}^T$, satisfying $(T_k)_{\mathbf{y}=0,\mathbf{z}=0} = S_k$ for every $k \in [2, d-1]$.

/* Computing the partial derivatives */

Compute blackbox access to $(\frac{\partial f'}{\partial y_s y_l z_t})_{\mathbf{y}=0,\mathbf{z}=0}$ and $(\frac{\partial f'}{\partial y_s z_l z_t})_{\mathbf{y}=0,\mathbf{z}=0}$ for all $s, l, t \in [w]$.

For every $s, t \in [w]$, let $\{h_e^{(s,t)} : e \in [w^2(d-4) + 2w]\}$ be the minors of the ABP $S_2(s, *) \cdot S_3 \ldots S_{d-2} \cdot S_{d-1}(*, t)$, as defined in Equation 12.

/* Computing *almost all* the coefficients of the $\mathbf{y}$ and $\mathbf{z}$ variables in T_k*/

Set $E = w^2(d-4) + 2w$.

for every $s, t \in [w]$ **do**

Pick $\mathbf{a}_1, \ldots, \mathbf{a}_E \in_r \mathbb{F}^{|\mathbf{r}|}$ independently.

Solve the linear system over $\mathbb{L}$ defined by

$$\sum_{e \in [E]} c_e \cdot h_e^{(s,t)}(\mathbf{a}_i) = \left(\frac{\partial f'}{\partial y_s^2 z_t}\right)_{\mathbf{y}=0,\mathbf{z}=0} (\mathbf{a}_i), \qquad \text{for } i \in [E], \tag{13}$$

for a *unique* solution of $\{c_e\}_{e \in [E]}$. If the coefficient matrix is not invertible, output 'Failed'.

For every $e \in [E]$, set the solution value of c_e as the coefficient of y_s in the e-th linear form of the ABP $T_2(s, *) \cdot T_3 \ldots T_{d-2} \cdot T_{d-1}(*, t)$.

Solve the linear system over $\mathbb{L}$ defined by

$$\sum_{e \in [E]} d_e \cdot h_e^{(s,t)}(\mathbf{a}_i) = \left(\frac{\partial f'}{\partial y_s z_t^2}\right)_{\mathbf{y}=0,\mathbf{z}=0} (\mathbf{a}_i), \qquad \text{for } i \in [E], \tag{14}$$

for a *unique* solution of $\{d_e\}_{e \in [E]}$.

For every $e \in [E]$, set the solution value of d_e as the coefficient of z_t in the e-th linear form of the ABP $T_2(s, *) \cdot T_3 \ldots T_{d-2} \cdot T_{d-1}(*, t)$.

end for

/* Computing the remaining $\mathbf{y}$ and $\mathbf{z}$ coefficients in T_2 and T_{d-1} */

for every $s, t \in [w]$ **do**

For every $l > s$, compute the coefficients of y_l in the linear forms in $T_2(s, *)$ by setting up a linear system similar to Equation 13, but with the R.H.S replaced by $\frac{\partial f'}{\partial y_s y_l z_1}$.

For every $l > t$, compute the coefficients of z_l in the linear forms in $T_{d-1}(*, t)$ by setting up a linear system similar to Equation 14, but with the R.H.S replaced by $\frac{\partial f'}{\partial y_1 z_l z_t}$.

end for

The coefficients of the $\mathbf{r}$ variables in the linear forms in T_k remain the same as that in S_k, for all $k \in [2, d-1]$. Output $T_2, T_3, \ldots T_{d-1}$.

Without loss of generality, let these minors be $h_1^{(s,1)}, \ldots, h_w^{(s,1)}$. Let $l > s$. By treating f' as a polynomial in the $\mathbf{y}, \mathbf{z}$ variables, with coefficients from $\mathbb{L}(\mathbf{r})$, and examining the coefficient of $y_s y_l z_1$ in f', we arrive at the equation,

$$\sum_{e=1}^{w} c_e \cdot h_e^{(s,1)} + K(\mathbf{r}) = \left(\frac{\partial f'}{\partial y_s y_l z_1} \right)_{\mathbf{y}=0,\mathbf{z}=0},$$

where $c_1, \ldots, c_w$ are the unknown coefficients of y_l in the linear forms of $T_2(s,*)$, and $K(\mathbf{r})$ is a *known* linear combination of some other minors. The fact that $K(\mathbf{r})$ is known at step 17 follows from this observation – while forming a monomial $y_s y_l z_1$, we either choose y_s from X_1' and y_l from $T_2(s,*)$ or $T_3, \ldots, T_{d-1}(*,1)$, or y_l from X_1' and y_s from $T_3, \ldots, T_{d-1}(*,1)$. In the latter case, we are using the fact that T_2 is in canonical form, and so y_s does not appear in $T_2(l,*)$. As the coefficients of y_s, y_l in $T_3, \ldots, T_{d-1}(*,1)$ are known from the computation in steps 6–13, we conclude that $K(\mathbf{r})$ in known. Thus, we can solve for $c_1, \ldots, c_w$ by plugging in w random points in place of the $\mathbf{r}$ variables and setting up a linear system in w variables. Corollary 4.1 and Claim 2.2 imply the $w \times w$ coefficient matrix of the system is invertible, and hence the solution for $c_1, \ldots, c_w$ is unique. The correctness of step 18 can be argued similarly, and this finally implies that T_2 and T_{d-1} (in canonical form) are unique.

Linear independence of minors: Proof of Lemma 4.2

We have to show that the minors of $R_2(s,*) \cdot R_3 \ldots R_{d-2} \cdot R_{d-1}(*,t)$ are $\mathbb{F}$-linearly independent with high probability, for every $s,t \in [w]$, where $R_2 \cdot R_3 \ldots R_{d-1}$ is a random $(w, d-2, n-2w)$-matrix product. We will prove it for a fixed $s,t \in [w]$, and then by union bound the result will follow for every $s,t \in [w]$. As $n \geq 4w^2$, we have $n - 2w \geq 3w^2$. So, it is sufficient to show the linear independence of the minors of a random (w,d,n)-ABP $X_1 \cdot X_2 \ldots X_d$ in $\mathbf{x}$-variables, for $n \geq 3w^2$.

Treat the coefficients of the linear forms in $X_1, \ldots, X_d$ as formal variables. In particular,

$$X_1 = \sum_{i=1}^{n} U_i^{(1)} x_i, \quad X_k = \sum_{i=1}^{n} U_i^{(k)} x_i \ \text{ for } k \in [2, d-1], \quad X_d = \sum_{i=1}^{n} U_i^{(d)} x_i, \tag{15}$$

where $U_i^{(1)}$ and $U_i^{(d)}$ are row and column vectors of length w respectively, $U_i^{(k)}$ is a $w \times w$ matrix, and the entries of these matrices are distinct $\mathbf{u}$-variables. We will denote the (ℓ, m)-th entry of $U_i^{(k)}$ by $U_i^{(k)}(\ell, m)$, and the m-th entry of $U_i^{(d)}$ by $U_i^{(d)}(m)$. From the above equations, $X_1 \cdot X_2 \ldots X_d$ is a (w,d,n)-ABP over $\mathbb{F}(\mathbf{u})$. We will show in the following claim that the minors of this ABP are $\mathbb{F}(\mathbf{u})$-linearly independent. As the coefficients of the $\mathbf{x}$-monomials of these minors are polynomials (in fact, multilinear polynomials) of degree $d-1$ in the $\mathbf{u}$-variables, an application of the Schwartz-Zippel lemma implies $\mathbb{F}$-linear independence of the minors (with high probability) when the $\mathbf{u}$-variables are set randomly to elements in $\mathbb{F}$ (as is done in a random ABP over $\mathbb{F}$).

Claim 4.1. *The minors of $X_1 \cdot X_2 \ldots X_d$ are $\mathbb{F}(\mathbf{u})$-linearly independent.*

Proof. We will prove by induction on d.

Base case (d=3): Clearly, if the minors are $\mathbb{F}$-linearly independent after setting the $\mathbf{u}$-variables to some $\mathbb{F}$-elements then the minors are also $\mathbb{F}(\mathbf{u})$-linearly independent before the setting. As

$n \geq w^2 + 2w$, it is possible to set the **u**-variables in X_1, X_2, X_3 such that the entries of these matrices (after the setting) become distinct **x**-variables. The minors of this **u**-evaluated ABP $X_1 \cdot X_2 \cdot X_3$ are monomial disjoint and so $\mathbb{F}$-linearly independent.

Inductive step: Split the $w^2(d-2) + 2w$ minors of $X_1 \cdot X_2 \ldots X_d$ into two sets: The first set G_1 consists of minors g_e, for $e \in [w^2(d-3) + 2w]$, such that the e-th linear form is the (ℓ, m)-th entry of some matrix X_k satisfying $k \neq d$ and if $k = d-1$ then $m = w$. The second set G_2 consists of minors g_e, for $e \in [w^2(d-3) + 2w + 1, w^2(d-2) + 2w]$, such that the e-th linear form is either the (ℓ, m)-th entry of X_{d-1} for $m \neq w$, or the ℓ-th entry of X_d. Set G_1 has $p = w^2(d-3) + 2w$ minors and G_2 has w^2 minors.

Suppose $\mu_1, \ldots, \mu_p$ are monomials in **x**-variables of degree $d-2$. Imagine a $(w^2(d-2) + 2w) \times (w^2(d-2) + 2w)$ matrix M whose rows are indexed by the minors in G_1 and G_2, and columns by monomials $\mu_1 x_1, \mu_2 x_1, \ldots, \mu_p x_1$ and $x_2^{d-1}, x_3^{d-1}, \ldots, x_{w^2+1}^{d-1}$. The (g, σ)-th entry of M contains the coefficient of the monomial σ in g, this coefficient is a multilinear polynomial in the **u**-variables. In a sequence of observations, we show that there exist $\mu_1, \ldots, \mu_p$ such that $\det(M) \neq 0$.

Consider the variable $u \stackrel{\text{def}}{=} U_1^{(d)}(w)$. The following observations are easy to verify.

Observation 4.2. *1. Variable u does not appear in any of the monomials of the (g, σ)-th entry of M if $g \in G_2$ or $\sigma \in \{x_2^{d-1}, \ldots, x_{w^2+1}^{d-1}\}$.*

2. Variable u appears in some monomials of the (g, σ)-th entry of M if $g \in G_1$ and $\sigma \in \{\mu_1 x_1, \ldots, \mu_p x_1\}$, irrespective of $\mu_1, \ldots, \mu_p$.

Observation 4.3. *Let $g \in G_1$ and $\sigma \in \{\mu_1 x_1, \ldots, \mu_p x_1\}$. If we treat the (g, σ)-th entry of M as a polynomial in u with coefficients from $\mathbb{F}[\mathbf{u} \setminus u]$ then the coefficient of u does not depend on the variables:*

(a) $U_i^{(d)}(j)$ for $j \neq w$ and $i \in [n]$,

(b) $U_i^{(d)}(w)$ for $i \in [2, n]$,

(c) $U_i^{(d-1)}(\ell, m)$ for $\ell, m \in [w]$ with $m \neq w$, and $i \in [n]$.

Denote the union of the **u**-variables specified in *(a)*, *(b)* and *(c)* of the above observation by **v**.

Observation 4.4. *The set $\{g_{\mathbf{v}=0} : g \in G_1\}$ equals the set $\{h \cdot u x_1 : h$ is a minor of $X_1 \cdot X_2 \ldots X_{d-1}(*, w)\}$.*

By the induction hypothesis, the minors of $X_1 \cdot X_2 \ldots X_{d-1}(*, w)$, say $h_1, \ldots, h_p$, are $\mathbb{F}(\mathbf{u})$-linearly independent. Hence there are p monomials in **x**-variables of degree $d-2$ such that $h_1, \ldots, h_p$, when restricted to these monomials, are $\mathbb{F}(\mathbf{u})$-linearly independent. These p monomials are our choices for $\mu_1, \ldots, \mu_p$. Let N be the $p \times p$ matrix with rows indexed by $h_1, \ldots, h_p$ and columns by $\mu_1, \ldots, \mu_p$, and $N(h, \mu)$ contains the coefficient of the monomial μ in h. Then, $\det(N) \neq 0$. Under these settings, we have the following observation (which can be derived easily from the above).

Observation 4.5. *The coefficient of u^p in $\det(M)$, when treated as a polynomial in u with coefficients from $\mathbb{F}[\mathbf{u} \setminus u]$, is $\det(N) \cdot \det(M_0)$, where M_0 is the submatrix of M defined by rows indexed by $\{g : g \in G_2\}$ and columns by $x_2^{d-1}, \ldots, x_{w^2+1}^{d-1}$.*

The next observation completes the proof of the claim by showing $\det(M) \neq 0$.

Observation 4.6. $\det(M_0) \neq 0$.

The proof of the above follows by noticing that M_0 looks like $(f_i(\mathbf{u}_j))_{i,j\in[w^2]}$, where $\mathbf{u}_1, \ldots, \mathbf{u}_{w^2}$ are some disjoint subsets of the $\mathbf{u}$-variables and $f_1, \ldots, f_{w^2}$ are $\mathbb{F}$-linearly independent polynomials. The observation then follows from Claim 2.2. □

4.3 Non-degenerate ABP

From the analysis, it can be easily shown that Theorem 2 gives a reconstruction algorithm for a (w, d, n)-ABP $X_1 \cdot X_2 \ldots X_d$, where $4w^2 \leq n \leq d^{w^2}$, and the following conditions are satisfied:

1. There are $w+1$ variables $\{x_1, x_2, \ldots, x_w, v\} \subset \mathbf{x}$ such that the linear forms in X_1 (similarly X_d) projected to $x_1, x_2, \ldots, x_w$ (i.e. after setting the variables other than $x_1, x_2, \ldots, x_w$ to zero) are $\mathbb{F}$-linearly independent. Further, if $\mathbf{u} = \mathbf{x} \setminus \{x_1, x_2, \ldots, x_w, v\}$ then in the bases $\{x_i - \alpha_i v - g_i(\mathbf{u}) \mid i \in [w]\}$ and $\{x_i - \beta_i v - h_i(\mathbf{u}) \mid i \in [w]\}$ of the spaces $\mathcal{X}_1$ and $\mathcal{X}_d$ (defined in Section 1.4.2) respectively, where $\alpha_i, \beta_i \in \mathbb{F}$ and g_i, h_i are linear forms in the $\mathbf{u}$-variables, $\alpha_1, \alpha_2, \ldots, \alpha_w, \beta_1, \beta_2, \ldots \beta_w$ are distinct elements of $\mathbb{F}$.

2. For every set $\mathbf{r} \subseteq \mathbf{x}$ of size $4w^2$ the following holds: The linear forms in X_1, X_d and every choice of three matrices among $X_2, \ldots, X_{d-1}$, projected to the $\mathbf{r}$-variables, are $\mathbb{F}$-linearly independent.

3. The matrix product $X_2 \cdot X_3 \ldots X_{d-1}$ modulo the $\mathbb{F}$-linear space spanned by the linear forms in X_1 and X_d is a pure product.

4. The minors of the ABP $X_2(s, *) \cdot X_3 \ldots X_{d-1}(*, t)$ (where $X_2(s, *)$ denotes the s-th row of X_2 and $X_{d-1}(*, t)$ the t-th column of X_{d-1}) modulo the $\mathbb{F}$-linear space spanned by the linear forms in X_1 and X_d are $\mathbb{F}$-linearly independent , for all $s, t \in [w]$.

Given a (w, d, n)-ABP, it can be checked whether the ABP satisfies condition 1 in deterministic $\binom{n}{w+1}(wn \log q)^{O(1)}$ time, condition 2 in deterministic $\binom{n}{4w^2}(wdn \log q)^{O(1)}$ time, and conditions 3 and 4 in randomized $(wdn \log q)^{O(1)}$ time. The one thing to note here is that, to reconstruct an ABP satisfying the above conditions, Algorithm 5 needs to be slightly modified as follows: At step 2, instead of working with a designated set of $w+1$ variables, the algorithm checks condition 1 for every choice of $w+1$ variables till it finds a correct choice. Then the running time of the algorithm is $\binom{n}{w+1}(wn \log q)^{O(1)} + (d^{w^3} n \log q)^{O(1)}$, which equals $(d^{w^3} n \log q)^{O(1)}$ for $n \leq d^{w^2}$.

5 Equivalence test for determinant over finite fields

We prove Theorem 3 in this section. It is known that the affine equivalence test can be reduced to equivalence test [Kay12], as briefly explained below.

Reduction to equivalence test: Suppose f is a (n, w)-polynomial that is affine equivalent to Det_w, where $n \geq w^2$. The following claim reduces the number of variables from n to w^2. A proof can be found in [Kay12] (see also Algorithm 8 and Claim 2.3 in [KNST17]).

Claim 5.1. *There is a randomized algorithm that takes input blackbox access to $f(\mathbf{x})$ and with probability $1 - \frac{n^{O(1)}}{q}$ outputs a matrix $C \in \mathsf{GL}(n, \mathbb{F})$ such that $f(C \cdot \mathbf{x})$ is a (w^2, w)-polynomial. The algorithm runs in $(n \log q)^{O(1)}$ time.*

Suppose $\mathbf{y} \subseteq \mathbf{x}$ is the set of w^2 variables appearing in $f(C \cdot \mathbf{x})$, and let $g(\mathbf{y})$ be the degree-w homogeneous component of $f(C \cdot \mathbf{x})$ which must be equivalent to Det_w. By using an equivalence test for Det_w, we can compute a $Q \in \mathsf{GL}(w^2, \mathbb{L})$ such that $g(\mathbf{y}) = \mathsf{Det}_w(Q \cdot \mathbf{y})$, implying $g(\mathbf{x}) = \mathsf{Det}_w(Q' \cdot \mathbf{x})$ where $Q' \in \mathbb{L}^{w^2 \times n}$ is obtained by padding Q with $(n - w^2)$ all-zero columns. Now observe that there is an $\mathbf{a} \in \mathbb{F}^n$ such that $f(C \cdot \mathbf{x}) = g(\mathbf{x} + \mathbf{a})$; the translation equivalence test in the claim below returns a $\mathbf{c} \in \mathbb{F}^n$ such that $f(C \cdot \mathbf{x}) = g(\mathbf{x} + \mathbf{c})$. Hence, $f(C \cdot \mathbf{x}) = \mathsf{Det}_w(Q'\mathbf{x} + Q' \cdot \mathbf{c})$ implying $f(\mathbf{x}) = \mathsf{Det}_w(Q'C^{-1}\mathbf{x} + Q' \cdot \mathbf{c})$. The algorithm in Theorem 3 returns $B = Q'C^{-1}$ and $\mathbf{b} = Q' \cdot \mathbf{c}$.

Claim 5.2. *Let $f(\mathbf{x}) = g(\mathbf{x} + \mathbf{a})$, where f, g are (n, d)-polynomials and $\mathbf{a} \in \mathbb{F}^n$. There is randomized algorithm that takes blackbox access to f and g and with probability $1 - \frac{(nd)^{O(1)}}{q}$ computes a $\mathbf{c} \in \mathbb{F}^n$ such that $f(\mathbf{x}) = g(\mathbf{x} + \mathbf{c})$.*

See [Kay12, DdOS14] (also Algorithm 9 and Lemma 2.1 in [KNST17]) for proofs of the claim.

For the rest of this section, set $n = w^2$. The equivalence test for Det_w is done in two steps: In the first step, the problem is reduced to the simpler problem of PS-equivalence testing. The second step then solves the PS-equivalence test. A (w^2, w)-polynomial $f \in \mathbb{L}[\mathbf{x}]$ is PS-equivalent to Det_w if there is a permutation matrix P and a diagonal matrix $S \in \mathsf{GL}(w^2, \mathbb{L})$ such that $f = \mathsf{Det}_w(PS \cdot \mathbf{x})$.

Lemma 5.1 ([Kay12]). *There is a randomized algorithm that takes input blackbox access to f, which is PS-equivalent to Det_w, and with probability $1 - \frac{w^{O(1)}}{q}$ outputs a permutation matrix P and a diagonal matrix $S \in \mathsf{GL}(w^2, \mathbb{L})$ such that $f = \mathsf{Det}_w(PS \cdot \mathbf{x})$. The algorithm runs in $(w \log q)^{O(1)}$ time.*

It is in the first step where our algorithm differs from (and slightly simplifies) [Kay12]. This reduction to PS-equivalence testing is given in Section 5.2. As in [Kay12], the algorithm uses the structure of the group of symmetries and the Lie algebra of Det_w. An estimate of the probability that a random element of the Lie algebra of g_{Det_w} has all its eigenvalues in $\mathbb{L}$ (Lemma 5.4) is key to the simplification in the first step.

5.1 Group of symmetries and Lie algebra of determinant

We state a few well known facts and claims about the Lie algebra and the group of symmetries of Det_w. Proofs of these can be found in [Kay12, KNST17] and the references therein.

Definition 5.1. The *group of symmetries* of an n-variate polynomial f, denoted as $\mathscr{G}_f$, consists of matrices $A \in \mathsf{GL}(n, \mathbb{F})$ such that $f(\mathbf{x}) = f(A \cdot \mathbf{x})$.

$\mathsf{Det}_w(\mathbf{x})$ is the determinant of the symbolic matrix $X = (x_{ij})_{i,j \in [w]}$, where $\mathbf{x} = \{x_{ij}\}_{i,j \in [w]}$. Let $A(X)$ denote the $w \times w$ linear matrix obtained by applying a transformation $A \in \mathbb{F}^{w^2 \times w^2}$ on $\mathbf{x}$.

Fact 1. *An $A \in \mathsf{GL}(w^2, \mathbb{F})$ is in $\mathscr{G}_{\mathsf{Det}_w}$ if and only if there are two matrices $S, T \in \mathsf{SL}(w, \mathbb{F})$ such that either $A(X) = S \cdot X \cdot T$ or $A(X) = S \cdot X^T \cdot T$.*

Definition 5.2. The *Lie algebra* of a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, denoted as $\mathfrak{g}_f$, is the set of all $n \times n$ matrices $E = (e_{ij})_{i,j\in[n]}$ in $\mathbb{F}^{n\times n}$ satisfying

$$\sum_{i,j\in[n]} e_{ij}x_j \cdot \frac{\partial f}{\partial x_i} = 0.$$

To express the Lie algebra of Det_w, order the variables of $\mathbf{x}$ in row major fashion and call them $x_1, \ldots, x_n$. Let $\mathcal{Z}_w$ be the $\mathbb{F}$-linear space of all $w \times w$ traceless matrices over $\mathbb{F}$, $\mathcal{L}_{\text{row}}$ be the space $\mathcal{Z}_w \otimes I_w = \{Z \otimes I_w : Z \in \mathcal{Z}_w\}$, and $\mathcal{L}_{\text{col}}$ the space $I_w \otimes \mathcal{Z}_w = \{I_w \otimes Z : Z \in \mathcal{Z}_w\}$.

Fact 2. $\mathfrak{g}_{\mathsf{Det}_w} = \mathcal{L}_{\text{row}} \oplus \mathcal{L}_{\text{col}}$.

It follows that the dimension of $\mathfrak{g}_{\mathsf{Det}_w}$ over $\mathbb{F}$ is $2w^2 - 2$.

Fact 3. *Let f, g be n-variate polynomials such that there is an $A \in \mathsf{GL}(n, \mathbb{F})$ satisfying $f = g(A \cdot \mathbf{x})$. Then $\mathfrak{g}_f = A^{-1} \cdot \mathfrak{g}_g \cdot A = \{A^{-1} \cdot L \cdot A \mid L \in \mathfrak{g}_g\}$.*

Claim 5.3. *There is a randomized algorithm that given blackbox access to a (n, d)-polynomial f over $\mathbb{F}$, computes an $\mathbb{F}$-basis of $\mathfrak{g}_f$ with probability $1 - \frac{(nd)^{O(1)}}{q}$. The algorithm runs in $(nd \log q)^{O(1)}$ time.*

From Fact 2, it is easy to observe that $\mathfrak{g}_{\mathsf{Det}_w}$ contains a diagonal matrix with distinct elements on the diagonal. The next claim can be proved using this observation.

Claim 5.4. *Let $L_1, \ldots, L_{2w^2-2}$ be an $\mathbb{F}$-basis of $\mathfrak{g}_{\mathsf{Det}_w}$, and $L = \sum_{i=1}^{2w^2-2} \alpha_i \cdot L_i$, where $\alpha_1, \ldots, \alpha_{2w^2-2} \in_r \mathbb{F}$ are picked independently. Then, the characteristic polynomial of L is square-free with probability $1 - \frac{w^{O(1)}}{q}$.*

The following lemma is the main technical contribution of this section.

Lemma 5.2. *Let $L_1, \ldots, L_{2w^2-2}$ be an $\mathbb{F}$-basis of $\mathfrak{g}_{\mathsf{Det}_w}$, and $L = \sum_{i=1}^{2w^2-2} \alpha_i \cdot L_i$, where $\alpha_1, \ldots, \alpha_{2w^2-2} \in_r \mathbb{F}$ are picked independently. Then, the characteristic polynomial of L is square-free and splits completely over $\mathbb{L}$ with probability at least $\frac{1}{2w^2}$.*

Proof. Let $h(y)$ be the characteristic polynomial of L. From Claim 5.4, h is square-free with probability $1 - \frac{w^{O(1)}}{q}$. From Fact 2, $L = L_1 + L_2$ where $L_1 \in \mathcal{L}_{\text{row}}$ and $L_2 \in \mathcal{L}_{\text{col}}$. As L is uniformly distributed over $\mathfrak{g}_{\mathsf{Det}}$, so is L_1 over $\mathcal{L}_{\text{row}}$ and L_2 over $\mathcal{L}_{\text{col}}$. In other words, if $L_1 = Z_1 \otimes I_w$ and $L_2 = I_w \otimes Z_2$ then Z_1, Z_2 are both uniformly (and independently) distributed over $\mathcal{Z}_w$. If the characteristic polynomial of Z_1 (similarly Z_2) is irreducible over $\mathbb{F}$ then the eigenvalues of Z_1 (respectively, Z_2) lie in $\mathbb{L}$ and are distinct. If this happens for both Z_1 and Z_2 then there are $D_1, D_2 \in \mathsf{GL}(w, \mathbb{L})$ such that $D_1^{-1}Z_1D_1$ and $D_2^{-1}Z_2D_2$ are diagonal matrices. This further implies,

$$(D_1^{-1} \otimes I_w) \cdot (I_w \otimes D_2^{-1}) \cdot L \cdot (I_w \otimes D_2) \cdot (D_1 \otimes I_w)$$

is a diagonal matrix, due to the observation below.

Observation 5.1. *For any $M, N \in \overline{\mathbb{F}}^{w\times w}$, $(M \otimes I_w)$ and $(I_w \otimes N)$ commutes. Also, if $M, N \in \mathsf{GL}(w, \overline{\mathbb{F}})$ then $(M \otimes I_w)^{-1} = (M^{-1} \otimes I_w)$ and $(I_w \otimes N)^{-1} = (I_w \otimes N^{-1})$.*

Thus, if we show that the characteristic polynomial of $Z \in_r \mathcal{Z}_w$ is irreducible with probability δ then with probability at least δ^2 the characteristic polynomial of L splits completely over $\mathbb{L}$. Much like the proof of Claim 5.4, it can be shown that the characteristic polynomial of $Z \in_r \mathcal{Z}_w$ is square-free with probability $1 - \frac{w^{O(1)}}{q}$. Hence, if the characteristic polynomial of $Z \in_r \mathcal{Z}'_w$, where $\mathcal{Z}'_w \subset \mathcal{Z}_w$ consists of matrices with distinct eigenvalues in $\overline{\mathbb{F}}$, is irreducible with probability ρ then $\delta \geq \rho \cdot (1 - \frac{w^{O(1)}}{q})$. Next, we lower bound ρ.

Let $\mathcal{P}$ be the set of monic, degree-w, square-free polynomials in $\mathbb{F}[y]$ with the coefficient of y^{w-1} equal to zero. Define a map ϕ from $\mathcal{Z}'_w$ to $\mathcal{P}$,

$$\phi : \quad Z \mapsto \text{characteristic polynomial of } Z.$$

The map ϕ is onto as the companion matrix of $p(y) \in \mathcal{P}$ belongs to its pre-image under ϕ. Let $\phi^{-1}(p(y))$ be the set of matrices in $\mathcal{Z}'_w$ that map to p.

Claim 5.5. *Let $p(y) \in \mathcal{P}$. Then*

$$\frac{(q^w - 1) \cdot (q^w - q) \dots (q^w - q^{w-1})}{q^w} \leq |\phi^{-1}(p(y))| \leq \frac{(q^w - 1) \cdot (q^w - q) \dots (q^w - q^{w-1})}{q^w(1 - \frac{w}{q})}.$$

Proof. Let C_p be the companion matrix of $p(y)$. If the characteristic polynomial of a $Z \in \mathcal{Z}'_w$ equals $p(y)$ then there is an $E \in \mathsf{GL}(w, \mathbb{F})$ such that $Z = E \cdot C_p \cdot E^{-1}$, as the eigenvalues of C_p are distinct in $\overline{\mathbb{F}}$. Moreover, for any $E \in \mathsf{GL}(w, \mathbb{F})$, $E \cdot C_p \cdot E^{-1} \in \mathcal{Z}'_w$ has characteristic polynomial $p(y)$. Hence, $\phi^{-1}(p(y)) = \{E \cdot C_p \cdot E^{-1} \mid E \in \mathsf{GL}(w, \mathbb{F})\}$. Suppose $E, F \in \mathsf{GL}(w, \mathbb{F})$ such that $F \cdot C_p \cdot F^{-1} = E \cdot C_p \cdot E^{-1}$. Then $E^{-1}F$ commutes with C_p. Since C_p has distinct eigenvalues in $\overline{\mathbb{F}}$, $E^{-1}F$ can be expressed as a polynomial in C_p, say $h(C_p)$, of degree at most $(w-1)$ with coefficients from $\mathbb{F}$. Let $\mathbb{F}[y]^{\leq (w-1)}$ denote the set of polynomials in $\mathbb{F}[y]$ of degree at most $w-1$. Conversely, if $h \in \mathbb{F}[y]^{\leq (w-1)}$ and $h(C_p)$ is invertible then $F = E \cdot h(C_p)$ is such that $F \cdot C_p \cdot F^{-1} = E \cdot C_p \cdot E^{-1}$. As $h_1(C_p) \neq h_2(C_p)$ for distinct $h_1, h_2 \in \mathbb{F}[y]^{\leq (w-1)}$, we have

$$|\phi^{-1}(p(y))| = \frac{|\mathsf{GL}(w, \mathbb{F})|}{|\{h \in \mathbb{F}[y] : \deg(h) \leq (w-1) \text{ and } h(C_p) \in \mathsf{GL}(w, \mathbb{F})\}|}.$$

The numerator is exactly $(q^w - 1) \cdot (q^w - q) \dots (q^w - q^{w-1})$, and the denominator is trivially upper bounded by q^w. A lower bound on the denominator can be worked out as follows: Let $\lambda_1, \dots, \lambda_w \in \overline{\mathbb{F}}$ be the distinct eigenvalues of C_p. If $h(y) = a_{w-1}y^{w-1} + a_{w-2}y^{w-2} + \dots + a_0 \in \mathbb{F}[y]$, then $h(\lambda_1), \dots, h(\lambda_w)$ are the eigenvalues of $h(C_p)$. Observe that

$$\begin{aligned} & \Pr_{h \in_r \mathbb{F}[y]^{\leq (w-1)}} \quad \{h(\lambda_i) = 0, \text{ for some fixed } i \in [w]\} \leq \frac{1}{q}, \\ \Rightarrow \quad & \Pr_{h \in_r \mathbb{F}[y]^{\leq (w-1)}} \quad \{h(\lambda_i) = 0, \text{ for any } i \in [w]\} \leq \frac{w}{q}, \\ \Rightarrow \quad & \Pr_{h \in_r \mathbb{F}[y]^{\leq (w-1)}} \quad \{h(C_p) \in \mathsf{GL}(w, \mathbb{F})\} \geq 1 - \frac{w}{q}. \end{aligned}$$

Hence, the denominator is lower bounded by $q^w(1 - \frac{w}{q})$.

$\square$

Let $\rho_p = \frac{|\phi^{-1}(p(y))|}{|\mathcal{Z}'_w|}$, the probability that $p(y)$ is the characteristic polynomial of $Z \in_r \mathcal{Z}'_w$. From Claim 5.5, it follows that

$$|\mathcal{Z}'_w| \leq \frac{(q^w - 1)\cdot(q^w - q)\dots(q^w - q^{w-1})}{q^w(1-\frac{w}{q})} \cdot |\mathcal{P}| \quad \Rightarrow \quad 1 - \frac{w}{q} \leq \rho_p \cdot |\mathcal{P}| \, .$$

We show in the next claim that a $p \in_r \mathcal{P}$ is irreducible over $\mathbb{F}$ with probability at least $\frac{1}{w}(1 - \frac{2}{q^{w/2}})$, implying the characteristic polynomial of $Z \in_r \mathcal{Z}'_w$ is irreducible over $\mathbb{F}$ with probability $\rho \geq \frac{1}{w}(1 - \frac{2}{q^{w/2}})(1 - \frac{w}{q})$. Therefore, the probability that the characteristic polynomial of $Z \in_r \mathcal{Z}_w$ is irreducible over $\mathbb{F}$ is $\delta \geq \frac{1}{w}(1 - \frac{2}{q^{w/2}})(1 - \frac{w}{q})(1 - \frac{w^{O(1)}}{q})$. As $q \geq w^7$, the probability that the characteristic polynomial of $L \in_r \mathfrak{g}_{\mathsf{Det}_w}$ splits completely over $\mathbb{L}$ is at least $\delta^2 \geq \frac{1}{2w^2}$.

Claim 5.6. *A polynomial $p \in_r \mathcal{P}$ is irreducible over $\mathbb{F}$ with probability at least $\frac{1}{w}(1 - \frac{2}{q^{w/2}})$.*

Proof. Let $\mathcal{F}$ be the set of monic, degree-w, square-free polynomials in $\mathbb{F}[y]$. The difference between $\mathcal{F}$ and $\mathcal{P}$ is that a polynomial in $\mathcal{P}$ additionally has coefficient of y^{w-1} equal to zero. We argue in the next paragraph that the fraction of $\mathbb{F}$-irreducible polynomials in $\mathcal{F}$ and in $\mathcal{P}$ are the same. As irreducible polynomials are square-free, the number of irreducible polynomials in $\mathcal{F}$ is at least $\frac{q^w - 2q^{w/2}}{w}$ [vzGG03]. Hence, the fraction of irreducible polynomials in $\mathcal{F}$ is at least $\frac{1}{w}(1 - \frac{2}{q^{w/2}})$.

Define a map Ψ from $\mathcal{F}$ to $\mathcal{P}$ as follows: For a $u(y) = y^w + a_{w-1}y^{w-1} + \ldots + a_0 \in \mathcal{F}$, define $\Psi(u) = u(y - \frac{a_{w-1}}{w})$. Observe that the coefficient of y^{w-1} in $\Psi(u)$ is zero. It is also an easy exercise to show that $\Psi(u_1) = \Psi(u_2)$ if and only if there exists an $a \in \mathbb{F}$ such that $u_1(y) = u_2(y+a)$. As $u(y)$ is irreducible over $\mathbb{F}$ if and only if $u(y+a)$ is irreducible over $\mathbb{F}$, for $a \in \mathbb{F}$, the fraction of $\mathbb{F}$-irreducible polynomials in $\mathcal{F}$ is the same as that in $\mathcal{P}$. $\square$

This completes the proof of Lemma 5.2. $\square$

5.2 Reduction to PS-equivalence testing

Algorithm 7 gives a reduction to PS-equivalence testing for Det_w. Suppose the input to the algorithm is a blackbox access to $f = \mathsf{Det}_w(A \cdot \mathbf{x})$, where $A \in \mathsf{GL}(w^2, \mathbb{F})$. We argue the correctness of the algorithm by tracing its steps:

Step 1: An $\mathbb{F}$-basis of $\mathfrak{g}_f$ can be computed efficiently using Claim 5.3.

Step 3–12: At step 4 an element F of $\mathfrak{g}_f$ is chosen uniformly at random. By Fact 3, $F = A^{-1} \cdot L \cdot A$, where L is a random element of $\mathfrak{g}_{\mathsf{Det}_w}$. Lemma 5.2 implies, in every iteration of the loop, h (at step 5) is square-free and splits completely over $\mathbb{L}$ with probability at least $\frac{1}{2w^2}$. Since the loop has $w^3 \log q$ iterations, the algorithm finds an h that is square-free and splits completely over $\mathbb{L}$, with probability at least $1 - \frac{1}{q}$. Assume that the algorithm succeeds in finding such an h, and suppose $\lambda_1, \ldots, \lambda_{w^2} \in \mathbb{L}$ are the distinct roots of h. The algorithm finds a D in step 7 by picking a random solution of the linear system obtained from the relation $F \cdot D = D \cdot \mathrm{diag}(\lambda_1, \ldots, \lambda_{w^2})$ treating the entries of D as formal variables. We argue next that $f(D \cdot \mathbf{x})$ is PS-equivalent to Det_w over $\mathbb{L}$.

Algorithm 7 Reduction to PS-equivalence

INPUT: Blackbox access to a (w^2, w)-polynomial $f \in \mathbb{F}[\mathbf{x}]$ that is equivalent to Det_w over $\mathbb{F}$.
OUTPUT: A $D \in \mathsf{GL}(w^2, \mathbb{L})$ such that $f(D \cdot \mathbf{x})$ is PS-equivalent to Det_w over $\mathbb{L}$.

1. Compute an $\mathbb{F}$-basis of $\mathfrak{g}_f$. Let $\{F_1, F_2, \ldots F_{2w^2-2}\}$ be the basis. Set $j = 1$.
2.
3. **for** $j = 1$ **to** $w^3 \log q$ **do**
4.   Pick $\alpha_1, \ldots, \alpha_{2w^2-2} \in_r \mathbb{F}$ independently. Set $F = \sum_{i \in [2w^2-2]} \alpha_i \cdot F_i$.
5.   Compute the characteristic polynomial h of F. Factorize h into irreducible factors over $\mathbb{L}$.
6.   **if** h is square-free and splits completely over $\mathbb{L}$ **then**
7.   Use the roots of h to compute a $D \in \mathsf{GL}(w^2, \mathbb{L})$ such that $D^{-1} \cdot F \cdot D$ is diagonal.
8.   Exit loop.
9.   **else**
10.   Set $j = j + 1$.
11.   **end if**
12. **end for**
13.
14. **if** No D found at step 7 in the loop **then**
15.   Output 'Failed'.
16. **else**
17.   Output D.
18. **end if**

By Fact 2, $L = L_1 + L_2$ where $L_1 \in \mathcal{L}_{\text{row}}$ and $L_2 \in \mathcal{L}_{\text{col}}$. In other words, there are $Z_1, Z_2 \in \mathcal{Z}_w$ such that $L_1 = Z_1 \otimes I_w$ and $L_2 = I_w \otimes Z_2$. It is easy to verify, if L has distinct eigenvalues then so do Z_1 and Z_2. Hence, there are $D_1, D_2 \in \mathsf{GL}(w, \overline{\mathbb{F}})$ such that $D_1 Z_1 D_1^{-1}$ and $D_2 Z_2 D_2^{-1}$ are both diagonal, implying

$$M \stackrel{\text{def}}{=} (D_1 \otimes I_w) \cdot (I_w \otimes D_2) \cdot L \cdot (D_1^{-1} \otimes I_w) \cdot (I_w \otimes D_2^{-1})$$

is diagonal (by Observation 5.1) with distinct diagonal entries. Also,

$$\begin{aligned} D^{-1} \cdot F \cdot D &= (AD)^{-1} \cdot L \cdot (AD) \\ &= ((D_1 \otimes I_w) \cdot (I_w \otimes D_2) \cdot AD)^{-1} \cdot M \cdot ((D_1 \otimes I_w) \cdot (I_w \otimes D_2) \cdot AD) \end{aligned}$$

As both $D^{-1} \cdot F \cdot D$ and M are diagonal matrices with distinct diagonal entries, it must be that

$$(D_1 \otimes I_w) \cdot (I_w \otimes D_2) \cdot AD = P \cdot S,$$

where P is a permutation matrix and $S \in \mathsf{GL}(w^2, \overline{\mathbb{F}})$ is a diagonal matrix. Now observe that $\mathsf{Det}_w((D_1 \otimes I_w) \cdot \mathbf{x}) = \beta \cdot \mathsf{Det}_w(\mathbf{x})$ and $\mathsf{Det}_w((I_w \otimes D_2) \cdot \mathbf{x}) = \gamma \cdot \mathsf{Det}_w(\mathbf{x})$, for $\beta, \gamma \in \overline{\mathbb{F}} \setminus \{0\}$. Hence,

$$\begin{aligned} \mathsf{Det}_w(P \cdot S \cdot \mathbf{x}) &= \mathsf{Det}_w((D_1 \otimes I_w) \cdot (I_w \otimes D_2) \cdot AD \cdot \mathbf{x}) \\ &= \beta\gamma \cdot \mathsf{Det}_w(AD \cdot \mathbf{x}) \\ &= \beta\gamma \cdot f(D \cdot \mathbf{x}) \\ \Rightarrow f(D \cdot \mathbf{x}) &= \mathsf{Det}_w(P \cdot S' \cdot \mathbf{x}), \end{aligned}$$

where $S' \in \mathsf{GL}(w^2, \overline{\mathbb{F}})$ is also diagonal. Therefore, $f(D \cdot \mathbf{x})$ is PS-equivalent to Det_w over $\overline{\mathbb{F}}$. As $f(D \cdot \mathbf{x}) \in \mathbb{L}[\mathbf{x}]$, it is a simple exercise to show that $f(D \cdot \mathbf{x})$ must be PS-equivalent to Det_w over $\mathbb{L}$.

Acknowledgment

We thank Sébastien Tavenas for a few initial discussions on this work. Thanks also to the anonymous reviewers for their helpful comments.

References

[AH17] Eric Allender and Shuichi Hirahara. New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, pages 54:1–54:14, 2017.

[AV08] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 67–75, 2008.

[BC92] Michael Ben-Or and Richard Cleve. Computing Algebraic Formulas Using a Constant Number of Registers. *SIAM J. Comput.*, 21(1):54–58, 1992.

[BIJL18] Markus Bläser, Christian Ikenmeyer, Gorav Jindal, and Vladimir Lysikov. Generalized Matrix Completion and Algebraic Natural Proofs. In *50th ACM Symposium on Theory of Computing (STOC 2018), Los Angeles, California, USA*, 2018.

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning Algorithms from Natural Proofs. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 10:1–10:24, 2016.

[CKK+15] Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining Circuit Lower Bound Proofs for Meta-Algorithms. *Computational Complexity*, 24(2):333–392, 2015.

[CLO07] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms (3. ed.)*. Springer, 2007.

[CZ81] David G. Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Math. Comp.*, 36(154):587–592, 1981.

[DdOS14] Zeev Dvir, Rafael Mendes de Oliveira, and Amir Shpilka. Testing Equivalence of Polynomials under Shifts. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 417–428, 2014.

[DLM+08] Dana Dachman-Soled, Homin K. Lee, Tal Malkin, Rocco A. Servedio, Andrew Wan, and Hoeteck Wee. Optimal Cryptographic Hardness of Learning Monotone Functions. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, pages 36–47, 2008.

[EGdOW18] Klim Efremenko, Ankit Garg, Rafael Mendes de Oliveira, and Avi Wigderson. Barriers for Rank Methods in Arithmetic Complexity. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 1:1–1:19, 2018.

[FK09] Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. *J. Comput. Syst. Sci.*, 75(1):27–36, 2009.

[FS13] Michael A. Forbes and Amir Shpilka. Quasipolynomial-Time Identity Testing of Non-commutative and Read-Once Oblivious Algebraic Branching Programs. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 243–252, 2013.

[FSV17] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving algebraic circuits lower bounds. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 653–664, 2017.

[GKKS14] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *J. ACM*, 61(6):33:1–33:16, 2014.

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016.

[GKL11] Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Efficient Reconstruction of Random Multilinear Formulas. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 778–787, 2011.

[GKQ13] Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random Arithmetic Formulas Can Be Reconstructed Efficiently. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 1–9, 2013.

[GKSS17] Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017.

[Hås90] Johan Håstad. Tensor Rank is NP-Complete. *J. Algorithms*, 11(4):644–654, 1990.

[HW99] Ming-Deh A. Huang and Yiu-Chung Wong. Solvability of systems of polynomial congruences modulo a large prime. *Computational Complexity*, 8(3):227–257, 1999.

[Ier89] Douglas John Ierardi. *The Complexity of Quantifier Elimination in the Theory of an Algebraically Closed Field*. PhD thesis, Department of Computer Science, Cornell University, Ithaca, New York 14853-7501, 1989.

[JKS02] Jeffrey C. Jackson, Adam R. Klivans, and Rocco A. Servedio. Learnability beyond AC0. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 776–784, 2002.

[JLSW08] Jeffrey C. Jackson, Homin K. Lee, Rocco A. Servedio, and Andrew Wan. Learning Random Monotone DNF. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008, Boston, MA, USA, August 25-27, 2008. Proceedings*, pages 483–497, 2008.

[Kay12] Neeraj Kayal. Affine projections of polynomials: extended abstract. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 643–662, 2012.

[KLSS17] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Formulas. *SIAM J. Comput.*, 46(1):307–335, 2017.

[KNST17] Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of Full Rank Algebraic Branching Programs. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 21:1–21:61, 2017.

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012.

[KS06] Adam R. Klivans and Amir Shpilka. Learning Restricted Models of Arithmetic Circuits. *Theory of Computing*, 2(10):185–206, 2006.

[KS16a] Neeraj Kayal and Chandan Saha. Lower Bounds for Depth-Three Arithmetic Circuits with small bottom fanin. *Computational Complexity*, 25(2):419–454, 2016.

[KS16b] Mrinal Kumar and Shubhangi Saraf. Sums of Products of Polynomials in Few Variables: Lower Bounds and Polynomial Identity Testing. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 35:1–35:29, 2016.

[KS17] Mrinal Kumar and Shubhangi Saraf. On the Power of Homogeneous Depth 4 Arithmetic Circuits. *SIAM J. Comput.*, 46(1):336–387, 2017.

[KS18] Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:191, 2018.

[KSS14] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 146–153, 2014.

[KST16] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An Almost Cubic Lower Bound for Depth Three Arithmetic Circuits. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 33:1–33:15, 2016.

[KT90] Erich Kaltofen and Barry M. Trager. Computing with Polynomials Given By Black Boxes for Their Evaluations: Greatest Common Divisors, Factorization, Separation of Numerators and Denominators. *J. Symb. Comput.*, 9(3):301–320, 1990.

[Kum17] Mrinal Kumar. A Quadratic Lower Bound for Homogeneous Algebraic Branching Programs. In *Proceedings of the 32nd Computational Complexity Conference*, CCC '17, pages 19:1–19:16, 2017.

[Laz01] Daniel Lazard. Solving systems of algebraic equations. *ACM SIGSAM Bulletin*, 35(3):11–37, 2001.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant Depth Circuits, Fourier Transform, and Learnability. *J. ACM*, 40(3):607–620, 1993.

[LMW17] Dong Lu, Xiaodong Ma, and Dingkang Wang. A New Algorithm for General Factorizations of Multivariate Polynomial Matrices. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 277–284, 2017.

[LSW06] Homin K. Lee, Rocco A. Servedio, and Andrew Wan. DNF Are Teachable in the Average Case. In *Learning Theory, 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006, Proceedings*, pages 214–228, 2006.

[Nis91] Noam Nisan. Lower Bounds for Non-Commutative Computation (Extended Abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418, 1991.

[NW97] Noam Nisan and Avi Wigderson. Lower Bounds on Arithmetic Circuits Via Partial Derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[Raz85] Alexander A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Soviet Mathematics Doklady*, 31:354–357, 1985.

[Raz09] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009.

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

[RY09] Ran Raz and Amir Yehudayoff. Lower Bounds and Separations for Constant Depth Multilinear Circuits. *Computational Complexity*, 18(2):171–207, 2009.

[Sch80] Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM*, 27(4):701–717, 1980.

[Shi16] Yaroslav Shitov. How hard is the tensor rank? *arXiv*, abs/1611.01559, 2016.

[Sri15] Srikanth Srinivasan. A Compression Algorithm for $\mathrm{AC}^0[\oplus]$ circuits using Certifying Polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:142, 2015.

[SW01] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

[SWZ17] Zhao Song, David P. Woodruff, and Peilin Zhong. Relative Error Tensor Low Rank Approximation. *CoRR*, abs/1704.08246, 2017.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[Tav13] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. In *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, pages 813–824, 2013.

[Uma99] Christopher Umans. Hardness of Approximating $\mathrm{Sigma}_2\mathrm{P}$ Minimization Problems. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 465–474, 1999.

[Vol16] Ilya Volkovich. A Guide to Learning Arithmetic Circuits. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1540–1561, 2016.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra (2. ed.)*. Cambridge University Press, 2003.

[Wil14] Ryan Williams. Nonuniform ACC Circuit Lower Bounds. *J. ACM*, 61(1):2:1–2:32, 2014.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, pages 216–226, 1979.

A Proof of two claims

Claim 2.4 (restated): *If $E = Q_1 \cdots Q_\ell$ is a random (w, ℓ, m)-matrix product over $\mathbb{F}$, where $w^2 + 1 \leq m \leq n$ and $\ell \leq d$, then the entries of E are $\mathbb{F}$-linearly independent with probability $1 - (wdn)^{-\Omega(1)}$.*

Proof. Treat the coefficients of the linear forms in $Q_1, Q_2, \ldots, Q_\ell$ as distinct formal variables. In particular

$$Q_k = \sum_{i=1}^{m} U_i^{(k)} x_i \quad \text{for} \quad k \in [\ell] \, ,$$

where the $U_i^{(k)}$'s are $w \times w$ matrices and the entries of these matrices are distinct $\mathbf{u}$-variables. The entries of the matrix product E are polynomials in the $\mathbf{x}$-variables over $\mathbb{F}(\mathbf{u})$. If we show the w^2 entries of E are $\mathbb{F}(\mathbf{u})$-linearly independent then an application of Schwartz-Zippel lemma implies the statement of the claim. On the other hand, to show that the entries of E are $\mathbb{F}(\mathbf{u})$-linearly independent, it is sufficient to show that the entries are $\mathbb{F}$-linearly independent under a setting of the $\mathbf{u}$-variables to $\mathbb{F}$ elements. Consider such a setting: For every $k \in [\ell] \setminus \{1\}$, let $U_{w^2+1}^{(k)} = I_w$ and $U_i^{(k)} = 0$ for all $i \in [m] \setminus \{w^2+1\}$. Let $U_i^{(1)} = 0$ for all $i \geq w^2 + 1$ and set $U_1^{(1)}, \ldots, U_{w^2}^{(1)}$ in a way so that the linear forms in $\sum_{i=1}^{w^2} U_i^{(1)} x_i$ are $\mathbb{F}$-linearly independent. It is straightforward to check that the entries of E under this setting are $\mathbb{F}$-linearly independent. $\square$

Claim 3.1 (restated): *With probability* $1 - (wdn)^{-\Omega(1)}$*, any subset of* w *vectors in any of the sets* $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{w+1}\}$*,* $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{w+1}\}$*,* $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{w+1}\}$*, or* $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{w+1}\}$ *are* $\mathbb{L}$*-linearly independent.*

Proof. From Observation 3.3, for the sets $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{w+1}\}$ and $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{w+1}\}$ it is sufficient to show that any w columns of the $w \times (w+1)$ matrices $(N_{1i}(\mathbf{a}_j))_{i\in[w],j\in[w+1]}$ and $(N_{1i}(\mathbf{b}_j))_{i\in[w],j\in[w+1]}$ are $\mathbb{L}$-linearly independent with high probability. As the cofactors $N_{11}, \ldots, N_{1w}$ are $\mathbb{L}$-linearly independent, the above follows from Claim 2.2. For the sets $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{w+1}\}$ and $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{w+1}\}$, it follows from Equation 2 that there are $\lambda_k, \rho_k \in \mathbb{L}^\times$ such that $D \cdot \mathbf{v}_k = \lambda_k \mathbf{u}_k$ and $D \cdot \mathbf{s}_k = \rho_k \mathbf{w}_k$ for all $k \in [w+1]$. Since D is invertible, the claim follows for these two sets as well. $\square$

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il