A NOTE ON THE TENSOR RANK OF ALGEBRAICALLY-DEFINED TENSORS

ALI KARIM LALANI[†]

Abstract. We resolve an open question posed in Appendix D of Alexeev, Forbes, and Tsimerman (2011) regarding the tensor rank upper bound of a family of algebraically-defined tensors. We first prove the equality between the tensor rank of this particular family and the bilinear complexity of polynomial multiplication in a quotient ring. Using known bounds on bilinear complexity, the tight upper bounds on the rank follow immediately. For completeness and expository purposes, we demonstrate three distinct arguments to bound the bilinear complexity and thus the rank: a simple interpolation argument gives a loose quasi-linear upper bound of $O(n \log^2 n)$, a recursive approach over finite fields improves this to an upper bound of $O(n \cdot 2^{O(\log^* n)})$, and finally, the optimal linear upper bound of $O(n)$ is achieved by using the algebraic curve method of Chudnovsky and Chudnovsky. Matching the linear lower bound of Alexeev et al., this implies the tensor rank of this family is $\Theta(n)$.

1. Introduction

A major goal in theoretical computer science, particularly within algebraic complexity theory [SY10], is finding strong lower bounds on the size of arithmetic circuits and formulas computing explicit polynomials. A result by Raz [Raz10] says that if one can find an explicit family of order-d tensors of low degree with near-maximal rank, it would immediately imply super-polynomial lower bounds for general arithmetic formulas.

This motivates the search for explicit, high-rank tensors, but analyzing tensor rank is extremely difficult. Indeed, Håstad [Has90] proved that determining the exact rank of a given tensor is NP-hard.

To better understand the properties of explicit tensors, Alexeev, Forbes, and Tsimerman [AFT11] explored various constructions. Their work is particularly important because it shows field-independent rank lower bounds that remain state-of-the-art today. Specifically, in Section 5 of their paper, they showed using an analog of Gaussian elimination that over an arbitrary field $\mathbb{F}$, there are explicit tensors $T_n : [n]^3 \to \mathbb{F}$ such that $\operatorname{rank}(T_n) = 3n - \Theta(\lg n)$.

Providing an avenue for possible improvement, the authors introduced an alternative construction in Appendix D of their paper. They constructed a specific family of order-3 tensors $T_n = [M_1|\dots|M_n]$ of size $n \times n \times n$ constructed using multiplication maps in a field extension $\mathbb{K} = \mathbb{F}_q[x]/\langle f(x)\rangle$, where $f(x)$ is an (ideally sparse) irreducible polynomial of degree n.

While the authors were able to show a lower bound of $\frac{2q-1}{q-1}n - \Theta(\log_q n)$ for the this family, the exact upper bound was left as an open question.

Date: July 2026.

[†]Department of Computer Science, University of Toronto. Email: ali@cs.toronto.edu.

1.1. **Our Contributions.** Our main result closes the gap left open by Alexeev, Forbes, and Tsimerman [AFT11] by matching their linear lower bound with a linear upper bound, showing that the exact tensor rank of this family is $\Theta(n)$.

To do this, we reduce the problem to finding the bilinear complexity of polynomial multiplication in a quotient ring. Because this is a well-studied area ([Blä13], [BCS97]), upper bounds on the tensor rank follow immediately from known results. However, for completeness and expository purposes, we survey three distinct techniques to bound this complexity. These arguments become increasingly involved but give us progressively better bounds:

(1) **Evaluation and Interpolation over Extensions:** We begin with a simple argument that uses evaluation and interpolation over a large field extension. Mapping the resulting rank back to the base field via a field transfer step yields a loose quasi-linear upper bound of $O(n \log^2 n)$. This approach is inspired by [BCS97].
(2) **Recursive Reduction:** We avoid this field transfer penalty by recursively changing the base field. We regard the high degree polynomials as low degree polynomials whose coefficients are themselves elements of an extension field. Applying this reduction recursively gives a tighter bound of $O(n \cdot 2^{O(\log^* n)})$.[Kop26]
(3) **Evaluation and Interpolation over Algebraic Curves:** Finally, shifting to algebraic curves over a finite field removes the iterated logarithm factor. Implementing the method of Chudnovsky and Chudnovsky gives the optimal $O(n)$ bound [CC88, STV92, Bal99].

We state our main result formally:

Theorem 1.1. *Let T_n be the family of algebraicly defined tensors as defined in Appendix D of* [AFT11]. *Then, the tensor rank of T_n is $\Theta(n)$.*

This confirms that these specific tensors fall far short of the high rank required for formula lower bounds.

Organization of the Paper. The rest of this note is organized as follows. In Section 2, we cover the necessary background on tensors, tensor rank, and bilinear complexity. Section 3 proves the general equivalence between the bilinear complexity of a set of bilinear forms and the rank of its associated tensor. In Section 4, we formally define the tensors from Appendix D and prove the equality between its tensor rank and the bilinear complexity of polynomial multiplication modulo an irreducible polynomial. From there, we move onto upper bounds: Section 5 shows the quasi-linear bound using basic interpolation, Section 6 improves this to $O(n \cdot 2^{O(\log^* n)})$ via recursion, and finally Section 7 applies algebraic curves to get the optimal linear bound, wrapping up the proof of Theorem 1.1.

2. Preliminaries

In this section, we lay out the background regarding tensors, tensor rank, and bilinear complexity. For consistency, our notation and definitions in Subsection 2.1 adhere closely to those in [AFT11].

Throughout this paper, $\mathbb{F}$ shall denote an arbitrary field, the variable n for the tensor size, and d for the order. $\mathbb{F}_q$ will denote the finite field of q elements.

2.1. **Introduction to Tensors.**

Definition 2.1 (Tensors)**.** A tensor over a field $\mathbb{F}$ is a function $T : \prod_{j=1}^{d}[n_j] \to \mathbb{F}$. It is said to have order d and size $(n_1, \ldots, n_d)$. If all of the n_j are equal to n, then T is said to have size n^d. T is said to belong to the tensor product space $\bigotimes_{j=1}^{d} \mathbb{F}^{n_j}$.

Definition 2.2 (Simple Tensors)**.** A tensor $T : \prod_{j=1}^{d}[n_j] \to \mathbb{F}$ is *simple* if for $j \in [d]$ there are vectors $\vec{v}_j \in \mathbb{F}^{n_j}$ such that $T = \bigotimes_{j=1}^{d} \vec{v}_j$. That is, for all $i_j \in [n_j]$, $T(i_1, \ldots, i_d) = \prod_{j=1}^{d} \vec{v}_j(i_j)$ where $\vec{v}_j(i_j)$ denotes the i_j-th coordinate of $\vec{v}_j$.

Definition 2.3 (Tensor Rank)**.** The *rank* of a tensor $T : \prod_{j=1}^{d}[n_j] \to \mathbb{F}$, is defined as the minimum number of terms in a summation of simple tensors expressing T. That is,

$$\operatorname{rank}_{\mathbb{F}}(T) = \min \left\{ r : T = \sum_{k=1}^{r} \bigotimes_{j=1}^{d} \vec{v}_{j,k},\ \vec{v}_{j,k} \in \mathbb{F}^{n_j} \right\}$$

By definition, a non-zero tensor is simple if and only if it is of rank one.

Definition 2.4 (Tensor Layers/Slices)**.** For $T_1, \ldots, T_{n_d} \in \bigotimes_{j=1}^{d-1} \mathbb{F}^{n_j}$, define $T = [T_1 | \ldots | T_{n_d}]$ by the equation $T(i_1, \ldots, i_{d-1}, i_d) = T_{i_d}(i_1, \ldots, i_{d-1})$. The T_{i_j} are said to be the *layers* of T (along the d-th axis). Layers along other axes are defined analogously.

Remark 2.5 (Permutation Invariance of Rank)**.** Let T' be a tensor obtained by permuting the axes of T (for example, swapping the second and third coordinates). The tensor rank is invariant under any such permutation. This holds because any rank-r decomposition of T into a sum of simple tensors naturally induces a rank-r decomposition of T' by simply reordering the vector factors in each summand.

2.2. **Introduction to Bilinear Complexity.**

Definition 2.6 (Bilinear Complexity [Blä13], Section 4.2)**.** Let $X = \{x_1, \ldots, x_M\}$ and $Y = \{y_1, \ldots, y_N\}$ be two disjoint sets of variables, and let $F = \{f_1, \ldots, f_K\}$ be a set of bilinear forms over a field $\mathbb{F}$. We consider a computational model, the *Ostrowski measure*, where all linear operations (addition and scalar multiplication) are free of cost.

The *bilinear complexity* of F, denoted by $R(F)$, is the minimal number of products

$$P_\lambda = \left(\sum_{\mu=1}^{M} u_{\lambda\mu} x_\mu \right) \left(\sum_{\nu=1}^{N} v_{\lambda\nu} y_\nu \right), \quad 1 \le \lambda \le r$$

where the coefficients $u_{\lambda\mu}$ and $v_{\lambda\nu}$ are scalars from the base field $\mathbb{F}$ such that $F \subseteq \operatorname{span}_{\mathbb{F}}\{P_1, \ldots, P_r\}$. That is, each target bilinear form $f_k \in F$ can be expressed as an $\mathbb{F}$-linear combination of these r products.

To illustrate this definition concretely, we consider the example of complex multiplication viewed as an $\mathbb{R}$-algebra.

Example 2.7 (Complex Multiplication [Blä13], Example 4.6)**.** Let x_0, x_1 and y_0, y_1 be real variables representing two complex numbers $x_0 + x_1 i$ and $y_0 + y_1 i$. Their product expands as:

$$(x_0 + x_1 i)(y_0 + y_1 i) = (x_0 y_0 - x_1 y_1) + (x_0 y_1 + x_1 y_0) i$$

This multiplication is completely described by evaluating the set of two bilinear target forms $F = \{f_0, f_1\}$ defined by:

$$f_0 = x_0y_0 - x_1y_1$$
$$f_1 = x_0y_1 + x_1y_0$$

While a naive calculation uses 4 multiplications, we can compute F using only $r = 3$ products of linear forms by defining:

$$P_1 = x_0y_0$$
$$P_2 = x_1y_1$$
$$P_3 = (x_0 + x_1)(y_0 + y_1)$$

By taking linear combinations of these products, we reconstruct the target forms:

$$f_0 = P_1 - P_2$$
$$f_1 = P_3 - P_1 - P_2$$

This is essentially Karatsuba's algorithm on $\mathbb{R}[i]/\langle x^2 + 1\rangle$. Because the other operations required to form P_3 (adding $x_0 + x_1$ and $y_0 + y_1$) and the final operations required to reconstruct the outputs are just linear additions and subtractions, they are free of cost.

Since $F \subseteq \text{span}_{\mathbb{R}}\{P_1, P_2, P_3\}$, it follows $R(F) \leq 3$.

3. Associated Tensors and Tensor Rank Equivalence

To every set of bilinear forms F there is a corresponding order 3-tensor t and vice versa. As we will see, these notions of rank coincide.

Definition 3.1 (Associated Tensor)**.** Let $F = \{f_1, \ldots, f_K\}$ be a set of bilinear forms over a field $\mathbb{F}$ in the variable sets $X = \{x_1, \ldots, x_M\}$ and $Y = \{y_1, \ldots, y_N\}$. Each bilinear form can be uniquely expanded as:

$$f_\kappa = \sum_{\mu=1}^{M}\sum_{\nu=1}^{N} t_{\kappa\mu\nu}x_\mu y_\nu, \quad 1 \leq \kappa \leq K$$

where the coefficients $t_{\kappa\mu\nu} \in \mathbb{F}$ form a 3-tensor $t \in \mathbb{F}^{K\times M\times N}$. In line with the convention in [AFT11], we may view this tensor as a map from its indices to the base field, $t : [K] \times [M] \times [N] \to \mathbb{F}$, defined by $t(\kappa, \mu, \nu) = t_{\kappa\mu\nu}$. We call this map the *associated tensor* of F.

Recall from Definition 2.3 that the tensor rank, $\text{rank}_{\mathbb{F}}(T)$, is the minimum number of simple tensors required to sum to T.

Claim 3.2 ([Blä13])**.** *Let F be a set of bilinear forms, and let t be its associated structural tensor. Then the bilinear complexity of F equals the tensor rank of t, i.e., $R(F) = \text{rank}(t)$.*

Proof. This follows from writing the target forms as linear combinations of non-scalar products and comparing the monomial coefficients on both sides. For the complete computation, see pages 16 and 17 of [Blä13]. □

4. The Appendix D Tensors

In Appendix D of [AFT11], the authors construct a family of order-3 tensors based on polynomial multiplication maps in field extensions. We formalize this construction below.

Definition 4.1 (Appendix D Multiplication Tensors [AFT11])**.** Let $f(x)$ be an irreducible polynomial of degree n over a finite field $\mathbb{F}_q$. Thus the field extension $\mathbb{K} = \mathbb{F}_q[x]/\langle f(x)\rangle$ is well-defined, which we regard as an n-dimensional $\mathbb{F}_q$-vector space with the standard polynomial basis $e_i = x^{i-1}$ for $1 \le i \le n$.

For each basis element e_i, left-multiplication modulo $f(x)$ defines an $\mathbb{F}_q$-linear map on $\mathbb{K}$. Let $M_i \in \mathbb{F}_q^{n\times n}$ be the matrix representing this linear map. The order-3 tensor T_n is defined by stacking these n matrix slices along the third axis:

$$T_n = [M_1|M_2|\dots|M_n]$$

Over $\mathbb{F}_2$, the original construction uses the sparse irreducible polynomial $f(x) = x^{2\cdot 3^\ell} + x^{3^\ell} + 1$ for degrees $n = 2 \cdot 3^\ell$ to ensure the tensor family is explicit.[1]

For these explicit tensors, Alexeev, Forbes, and Tsimerman proved the following linear lower bound.

Theorem 4.2 (Tensor Rank Lower Bound [AFT11])**.** *Let T_n be the multiplication tensor defined above over $\mathbb{F}_q$. The tensor rank of T_n satisfies:*

$$\mathit{rank}_{\mathbb{F}_q}(T_n) \ge \frac{2q-1}{q-1}n - \Theta(\log_q n)$$

While this gives a lower bound for the rank, non-trivial upper bounds were left open. Note that the equivalence and the upper bounds we prove in this note do not depend on the sparsity of $f(x)$, nor are they restricted to $\mathbb{F}_2$. We show below that for *any* arbitrary irreducible polynomial $f(x)$ over *any* finite field $\mathbb{F}_q$, the tensor rank of T_n identically equals the bilinear complexity of the associated polynomial multiplication map.

First, we define the bilinear complexity of a map explicitly in terms of its target coordinate forms.

Definition 4.3 (Bilinear Complexity of a Map)**.** Let $\phi : \mathbb{F}^M \times \mathbb{F}^N \to \mathbb{F}^K$ be a bilinear map. Fixing standard bases, the k-th coordinate of the output vector $\phi(\vec{u}, \vec{v})$ is given by a bilinear target form $c_k(\vec{u}, \vec{v})$. We define the *bilinear complexity* of the map ϕ, denoted $R(\phi)$, as the bilinear complexity of its set of coordinate forms:

$$R(\phi) := R(\{c_1, \dots, c_K\}).$$

Proposition 4.4. Let $\phi : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^n$ be the bilinear map defined by multiplication modulo $f(x)$ in $\mathbb{K}$. Then the bilinear complexity of ϕ equals the tensor rank of the associated tensor T_n constructed in [AFT11], i.e., $R(\phi) = \mathrm{rank}_{\mathbb{F}_q}(T_n)$.

Proof. Regard $\mathbb{K}$ as an n-dimensional $\mathbb{F}_q$-vector space with the standard polynomial basis $e_1, \dots, e_n$ where $e_i = x^{i-1}$. Let $A, B \in \mathbb{K}$ be represented by their coordinate vectors $\vec{a}, \vec{b} \in \mathbb{F}_q^n$. The map ϕ computes their product $C = AB \pmod{f(x)}$.

[1] A tensor is called explicit if $T_n(i_1, i_2, i_3)$ can be computed by algebraic circuits of size at most polynomial in $\mathrm{poly}(\lg n)$, that is, at most polynomial in the size of the input. We believe it is an open problem to find an explicit family of irreducible polynomials of degree n over $\mathbb{F}_q$ where the number of nonzero coefficients is bounded by $O(\log n)$.

By linearity in the first argument, we can expand the product as:

$$C = \phi\left(\sum_{i=1}^{n} a_i e_i, B\right) = \sum_{i=1}^{n} a_i \phi(e_i, B) = \sum_{i=1}^{n} a_i (e_i B)$$

In the authors' construction, $M_i \in \mathbb{F}_q^{n \times n}$ is defined as the matrix of the linear map representing left-multiplication by the basis element e_i. Therefore, the coordinate vector of the product $(e_i B)$ is precisely $M_i \vec{b}$.

Extracting the k-th coordinate of the output C, denoted c_k, we sum the k-th coordinates of each term:

$$c_k = \sum_{i=1}^{n} a_i \left(\sum_{j=1}^{n} (M_i)_{k,j} b_j\right) = \sum_{i=1}^{n} \sum_{j=1}^{n} (M_i)_{k,j} a_i b_j$$

The order-3 tensor $T_n = [M_1 | \ldots | M_n]$ is defined such that its slices along the third dimension are the matrices M_i. Thus, its entries are given by $T_n(k, j, i) = (M_i)_{k,j}$. Substituting this into our coordinate equation gives us:

$$c_k = \sum_{i=1}^{n} \sum_{j=1}^{n} T_n(k, j, i) a_i b_j$$

Thus, T_n is precisely the associated tensor of $\{c_1, \ldots, c_n\}$ up to a permutation of its second and third axes. By Remark 2.5 and Claim 3.2, the bilinear complexity of these target forms is equal to the tensor rank of this permuted tensor.

Therefore, $R(\phi) = R(\{c_1, \ldots, c_n\}) = \text{rank}_{\mathbb{F}_q}(T_n)$. □

5. A Quasi-Linear Upper Bound

Knowing that the tensor rank of T_n equals the bilinear complexity of polynomial multiplication modulo $f(x)$, we now bound this. Our approach for large fields is inspired by evaluation and interpolation techniques from Chapter 3.3 of [BCS97]. We first bound the complexity assuming a large base field in Subsection 5.1, and then show how to transfer this result to smaller fields in Subsection 5.2.

5.1. Large Field Upper Bound via Interpolation.

If the base field contains a sufficient number of distinct elements, we can compute this polynomial product efficiently via interpolation.

Lemma 5.1 (Large Field Upper Bound)**.** *Let $f(x)$ be a polynomial of degree n over a finite field $\mathbb{F}_q$. If the field is sufficiently large, specifically $q \geq 2n-1$, the map which does polynomial multiplication modulo $f(x)$, ϕ, satisfies $R(\phi) \leq 2n-1$. Consequently, the associated tensor T_n of ϕ satisfies $\text{rank}_{\mathbb{F}_q}(T_n) \leq 2n-1$.*

Proof. Let the two inputs to the multiplication map be $A(x), B(x)$ in the field extension $\mathbb{K} = \mathbb{F}_q[x]/\langle f(x)\rangle$. Let $\vec{a}, \vec{b} \in \mathbb{F}_q^n$ be their respective coordinate vectors in the standard polynomial basis $1, x, \ldots, x^{n-1}$. We wish to compute the coordinate vector of their reduced product $C(x) = A(x)B(x) \pmod{f(x)}$.

Because $\deg(A) \leq n-1$ and $\deg(B) \leq n-1$ within the quotient space, the standard polynomial product before reduction, $P(x) = A(x)B(x)$, has a degree strictly bounded by $(n-1)+(n-1) = 2n-2$. A polynomial of degree at most $2n-2$ is uniquely determined by its evaluations at $2n-1$ distinct points. Because we are given $q \geq 2n-1$, we can fix $2n-1$ distinct evaluation points $\alpha_1, \ldots, \alpha_{2n-1} \in \mathbb{F}_q$.

We construct the desired forms via the following:

1. **Evaluation:** For each $1 \leq k \leq 2n-1$, compute the scalars $u_k = A(\alpha_k)$ and $v_k = B(\alpha_k)$. Because evaluating a polynomial at a fixed point is equivalent to taking a fixed linear combination of its coefficients, the forms $u_k(\vec{a})$ and $v_k(\vec{b})$ are strictly linear. Under our computational model, these linear operations over $\mathbb{F}_q$ carry zero bilinear cost.
2. **Point-wise Multiplication:** Compute the $2n-1$ scalar products $m_k = u_k \cdot v_k$. Because both operands depend on the inputs, these are non-scalar multiplications over the base field $\mathbb{F}_q$. This step incurs a total bilinear cost of exactly $2n-1$.
3. **Interpolation:** By construction, $m_k = P(\alpha_k)$. Using Lagrange interpolation, the product $P(x)$ can be reconstructed as:
$$P(x) = \sum_{k=1}^{2n-1} m_k L_k(x)$$
where $L_k(x) = \prod_{j \neq k} \frac{x-\alpha_j}{\alpha_k - \alpha_j}$ are the fixed Lagrange polynomials.
4. **Reduction:** Finally, we apply the modulo reduction map $\pi_f : \mathbb{F}_q[x] \to \mathbb{F}_q[x]/\langle f(x)\rangle$. By linearity of π_f over $\mathbb{F}_q$:
$$C(x) = \pi_f(P(x)) = \sum_{k=1}^{2n-1} m_k \pi_f(L_k(x))$$

Let $\vec{w}_k \in \mathbb{F}_q^n$ denote the fixed coordinate vector of the reduced basis polynomial $\pi_f(L_k(x))$. Substituting our linear forms back into the expression, the final output

coordinate vector $\vec{c}$ expands as:

$$\vec{c} = \sum_{k=1}^{2n-1} \left(u_k(\vec{a}) \cdot v_k(\vec{b}) \right) \vec{w}_k$$

This procedure constructs the output forms using exactly $2n-1$ non-scalar products. Therefore, the bilinear complexity satisfies $R(\phi) \leq 2n-1$. Since T_n is the associated tensor of the map ϕ, it directly follows from Proposition 4.4 that $\text{rank}_{\mathbb{F}_q}(T_n) \leq 2n-1$, as desired. □

5.2. **Base Field Upper Bound via Field Transfer.** We now consider the case for an arbitrary finite field $\mathbb{F}_q$.

Lemma 5.2 (Field Transfer, Lemma 6.10 in [AFT11])**.** *Let $\mathbb{F}$ be a base field, and let $\mathbb{E}$ be a field extension of $\mathbb{F}$. For any order-d tensor T, the tensor rank satisfies*

$$\mathit{rank}_{\mathbb{F}}(T) \leq (\dim_{\mathbb{F}} \mathbb{E})^{d-1} \cdot \mathit{rank}_{\mathbb{E}}(T).$$

Theorem 5.3. *Let T_n be the associated tensor representing polynomial multiplication modulo an irreducible polynomial $f(x)$ of degree n over $\mathbb{F}_q$. The tensor rank of T_n satisfies $\mathit{rank}_{\mathbb{F}_q}(T_n) \leq 2n \log_q^2 n + O(n \log_q n)$, which is in particular $O(n \log_q^2 n)$.*

Proof. Let $\mathbb{E} = \mathbb{F}_{q^k}$ be an extension field of $\mathbb{F}_q$ of degree k. To ensure a sufficient number of evaluation points to execute the interpolation step of Lemma 5.1, we require at least $2n-1$ distinct points. Thus, we choose the smallest field extension degree k such that

$$|\mathbb{E}| = q^k \geq 2n-1.$$

This implies $k = \lceil \log_q(2n-1) \rceil$. Using the properties of logarithms and the ceiling function ($\lceil x \rceil < x+1$), we can explicitly bound this degree:

$$k < \log_q(2n) + 1 = \log_q n + \log_q 2 + 1.$$

Since $|\mathbb{E}| \geq 2n-1$, Lemma 5.1 guarantees that $\text{rank}_{\mathbb{E}}(T_n) \leq 2n-1$.

We can now map this rank back to the base field. By applying Lemma 5.2 with $\dim_{\mathbb{F}_q} \mathbb{E} = k$ and substituting our explicit bound for k, we obtain:

$$\begin{aligned} \text{rank}_{\mathbb{F}_q}(T_n) &\leq k^{3-1} \cdot \text{rank}_{\mathbb{E}}(T_n) \\ &= k^2 \cdot \text{rank}_{\mathbb{E}}(T_n) \\ &< (\log_q n + \log_q 2 + 1)^2 \cdot (2n-1) \\ &= 2n \log_q^2 n + O(n \log_q n). \end{aligned}$$

This completes the proof of the quasi-linear upper bound. □

6. Near-Linear Upper Bound via Recursion

While the previous section provides a quasi-linear bound, the field-transfer step introduces a penalty. In this section, we present an algorithm which avoids this by recursively changing the base field. Grouping the terms of the polynomials allows them to be regarded as lower-degree polynomials whose coefficients live in an extension field, yielding a far tighter bound. This algorithm is based on an unpublished observation of Swastik Kopparty [Kop26].

Theorem 6.1. *Let T_n be the associated tensor representing polynomial multiplication modulo an irreducible polynomial $f(x)$ of degree n over $\mathbb{F}_q$. The tensor rank of T_n satisfies $\text{rank}_{\mathbb{F}_q}(T_n) \leq O(n \cdot 2^{O(\log^* n)})$.*

Proof. Let $F(n)$ denote the bilinear complexity of multiplying two polynomials of degree strictly less than n over the base field $\mathbb{F}_q$.

Base Case: For a sufficiently small constant degree (e.g., $n \leq 16$), compute the product using standard polynomial multiplication or Karatsuba's algorithm. Because the degree is bounded by a constant, the bilinear complexity is strictly bounded by a constant, yielding $F(n) = O(1)$.

Grouping: Introduce a parameter for the block size $t = \lfloor \log_q n \rfloor$ and define an outer degree $k = \lceil n/t \rceil$. Group the coefficients of the input polynomials $A(x), B(x) \in \mathbb{F}_q[x]$ into blocks of size t. Defining a new variable $y = x^t$, expressing the inputs as polynomials $A'(y)$ and $B'(y)$ over these grouped coefficients gives:

$$A(x) = \sum_{i=0}^{k-1} a_i(x)(x^t)^i \implies A'(y) = \sum_{i=0}^{k-1} a_i(x)y^i$$

where each coefficient $a_i(x) \in \mathbb{F}_q[x]$ is a polynomial of degree strictly less than t.

Evaluating over an Extension Field: Regard the inner polynomial coefficients $a_i(x)$ as elements of the field $\mathbb{E} = \mathbb{F}_{q^d} \cong \mathbb{F}_q[z]/\langle E(z) \rangle$, where $E(z) \in \mathbb{F}_q[z]$ is an irreducible polynomial of degree $d = 2t$ (see Remark 6.2 for the choice of this degree), by substituting x with the field variable z. Consequently, the block polynomials can be viewed as elements of $\mathbb{E}[y]$.

Compute the product $H(y) = A'(y)B'(y)$ via the following operations:

1. **Evaluation:** Choose $2k$ distinct evaluation points $\alpha_1, \ldots, \alpha_{2k} \in \mathbb{E}$. Evaluate $A'(y)$ and $B'(y)$ at these points. Linear combinations and scalar multiplications over field elements carry zero bilinear cost.
2. **Recursive Pointwise Multiplication:** For each of the $2k$ evaluation points, compute the pointwise product $A'(\alpha_i) \cdot B'(\alpha_i)$. Because the evaluated values are elements of $\mathbb{E}$, each product is equivalent to multiplying two polynomials of degree strictly less than $d = 2t$ over $\mathbb{F}_q$. Compute each of these $2k$ products by recursively invoking this algorithm.
3. **Interpolation and Unpacking:** Using Lagrange interpolation, reconstruct the coefficients of $H(y)$ from the $2k$ pointwise products. This reconstruction only involves fixed linear combinations and so incurs no bilinear cost. Finally, map z back to x and y back to x^t to recover the standard polynomial product in $\mathbb{F}_q[x]$.

Recurrence Relation: To handle issues with the ceiling function cleanly, we may implicitly pad our input polynomials with leading zeros to the nearest multiple $n' = t\lceil n/t \rceil$. Since $t \leq \log_q n$, the padded size satisfies $n' \leq n + \log_q n$, so $n' = \Theta(n)$. Since bilinear complexity is monotone in degree, bounding $F(n')$ bounds $F(n)$, so it suffices to assume n is an exact multiple of t in the analysis below.

The total bilinear complexity is bottlenecked entirely by the $2k$ recursive pointwise multiplications in the extension field. This yields the recurrence relation:

$$F(n) \leq 2k \cdot F(2t) \leq \frac{2n}{t} F(2t)$$

Dividing both sides by n:

$$\frac{F(n)}{n} \leq \frac{2}{t} F(2t)$$

To simplify the recurrence, let $m = 2t$.

$$\frac{F(n)}{n} \leq 4 \cdot \frac{F(m)}{m}$$

Because $m = 2\lfloor \log_q n \rfloor \leq 2 \log_q n$, the degree of the problem shrinks logarithmically at each level of the recursion. The unrolling of this recurrence relation terminates when it reaches the constant base case, and the number of steps is given by the iterated logarithm function, $\log^* n$.

Unrolling the recurrence $\log^* n$ times yields:

$$\frac{F(n)}{n} \leq 4^{\log^* n} \cdot O(1) = 2^{2 \log^* n} \cdot O(1)$$

Finally, multiplying both sides by n gives:

$$F(n) \leq O\big(n \cdot 2^{O(\log^* n)}\big)$$

Recall the tensor rank of T_n equals the bilinear complexity of polynomial multiplication modulo $f(x)$ by Proposition 4.4. The final reduction modulo $f(x)$ step is linear so no bilinear cost, thus it follows that $\text{rank}_{\mathbb{F}_q}(T_n) \leq O(n \cdot 2^{O(\log^* n)})$. □

Remark 6.2. We use the extension degree $d = 2t$ to avoid issues when recovering the polynomial coefficients. Multiplying two blocks of degree up to $t - 1$ yields a product of degree up to $2t - 2$. So any extension of degree at least $2t - 1$ would work. For if we evaluated over an extension field of degree matching the block size exactly (such as $\mathbb{F}_{q^t}$), the field arithmetic would reduce the product modulo a polynomial of degree t, potentially causing a loss of higher degree data before we can recover the coefficients.

For example, consider the blocks $A(x) = x + 1$ and $B(x) = x + 1$ over $\mathbb{F}_2$, where the true polynomial product is $x^2 + 1$. Evaluating this product inside $\mathbb{F}_{2^2}$ using the irreducible $E(z) = z^2 + z + 1$ gives $A(z)B(z) = z^2 + 1 \equiv z \pmod{z^2 + z + 1}$. Reversing the variable substitution gives x, which is incorrect. This issue does not occur when using $E(z) = z^3 + z + 1$.

7. Optimal Linear Bound via Algebraic Curves

To achieve a linear bound of $O(n)$, we evaluate the polynomials on high-genus algebraic curves. Chudnovsky and Chudnovsky [CC88] introduced an algorithm that interpolates polynomials over algebraic curves defined over $\mathbb{F}_q$, proving that the bilinear complexity of polynomial multiplication is linear in n whenever suitable curves with many points relative to their genus are available over $\mathbb{F}_q$. Constructing such curves explicitly is more delicate over small base fields like $\mathbb{F}_2$, and subsequent work [STV92, Bal99] gave linear bilinear complexity bounds that hold uniformly over any finite field $\mathbb{F}_q$.

We treat this construction as a black box and state the resulting bound below.

Theorem 7.1 ([CC88, STV92, Bal99])**.** *Let ϕ be the bilinear map computing polynomial multiplication modulo an irreducible polynomial of degree n over any finite field $\mathbb{F}_q$. The bilinear complexity of ϕ is bounded by $R(\phi) \leq O(n)$.*

Combining Theorem 7.1 with the matching lower bound from Theorem 4.2 yields the exact tensor rank.

Corollary 7.2 (Exact Tensor Rank)**.** The tensor rank of the Appendix D multiplication tensor T_n over $\mathbb{F}_q$ is $\Theta(n)$.

Proof. By Proposition 4.4, $\text{rank}_{\mathbb{F}_q}(T_n) = R(\phi)$. Theorem 7.1 bounds this complexity by $O(n)$, giving the upper bound $\text{rank}_{\mathbb{F}_q}(T_n) \leq O(n)$. Since Theorem 4.2 establishes the matching lower bound of $\Omega(n)$, the exact rank is $\Theta(n)$. □

8. Acknowledgements

The author is deeply grateful to Professor Shubhangi Saraf for her invaluable supervision, support, and guidance.

The author also thanks Professor Swastik Kopparty for outlining the recursive argument in Section 6 and for bringing the linear upper bound to the author's attention.

References

[AFT11] B. Alexeev, M. Forbes, and J. Tsimerman. Tensor rank: Some lower and upper bounds. In *Proceedings of the 26th IEEE Conference on Computational Complexity*, pp. 283–291, 2011.

[Bal99] S. Ballet. Curves with many points and multiplication complexity in any extension of $\mathbb{F}_q$. *Finite Fields and Their Applications*, 5(4):364–377, 1999.

[Blä13] M. Bläser. *Fast Matrix Multiplication*. Theory of Computing, Graduate Surveys 5, 1–60, 2013.

[BCS97] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*. Grundlehren der mathematischen Wissenschaften, Vol. 315. Springer-Verlag, 1997.

[CC88] D. V. Chudnovsky and G. V. Chudnovsky. *Algebraic complexities and algebraic curves over finite fields*. Journal of Complexity, 4(4):285–316, 1988.

[Has90] J. Håstad. *Tensor Rank is NP-complete*. Journal of Algorithms, 11(4):644–654, 1990.

[Kop26] S. Kopparty. Personal communication, 2026.

[Raz10] R. Raz. *Tensor-rank and lower bounds for arithmetic formulas*. STOC '10: Proceedings of the forty-second ACM symposium on Theory of computing, 2010.

[STV92] I. E. Shparlinski, M. A. Tsfasman, and S. G. Vlăduţ. Curves with many points and multiplication in finite fields. In *Coding Theory and Algebraic Geometry (Luminy, 1991)*, Lecture Notes in Mathematics, Vol. 1518, pp. 145–169. Springer, Berlin, 1992.

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

Department of Computer Science, University of Toronto
Email address: ali@cs.toronto.edu