

Improved Rectangular Matrix Multiplication using Powers of the Coppersmith-Winograd Tensor*

François Le Gall[†]

Florent Urrutia[‡]

Abstract

In the past few years, successive improvements of the asymptotic complexity of square matrix multiplication have been obtained by developing novel methods to analyze the powers of the *Coppersmith-Winograd tensor*, a basic construction introduced thirty years ago. In this paper we show how to generalize this approach to make progress on the complexity of rectangular matrix multiplication as well, by developing a framework to analyze powers of tensors in an asymmetric way. By applying this methodology to the fourth power of the Coppersmith-Winograd tensor, we succeed in improving the complexity of rectangular matrix multiplication.

Let α denote the maximum value such that the product of an $n \times n^\alpha$ matrix by an $n^\alpha \times n$ matrix can be computed with $O(n^{2+\epsilon})$ arithmetic operations for any $\epsilon > 0$. By analyzing the fourth power of the Coppersmith-Winograd tensor using our methods, we obtain the new lower bound $\alpha > 0.31389$, which improves the previous lower bound $\alpha > 0.30298$ obtained by Le Gall (FOCS'12) from the analysis of the second power of the Coppersmith-Winograd tensor. More generally, we give faster algorithms computing the product of an $n \times n^k$ matrix by an $n^k \times n$ matrix for any value $k \neq 1$. (In the case $k = 1$, we recover the bounds recently obtained for square matrix multiplication).

These improvements immediately lead to improvements in the complexity of a multitude of fundamental problems for which the bottleneck is rectangular matrix multiplication, such as computing the all-pair shortest paths in directed graphs with bounded weights.

1 Introduction

1.1 Background. Matrix multiplication is one of the most important problems in mathematics and computer science. In 1969, Strassen discovered the first algorithm with subcubic complexity computing the product of two square matrices [25]. In modern notation, Strassen's result can be stated as an upper bound $\omega < 2.81$ on the exponent of square matrix multiplication ω , defined as the minimum value such that two $n \times n$ matrices can be multiplied using $O(n^{\omega+\epsilon})$ arithmetic operations for any constant $\epsilon < 0$. Strassen's breakthrough initiated intense work on the complexity of matrix multiplication, which

in a span of a few decades lead to several improvements, culminating in the celebrated $O(n^{2.376})$ -time algorithm for square matrix multiplication by Coppersmith and Winograd [10], i.e., the upper bound $\omega < 2.376$ on the exponent of square matrix multiplication. This algorithm is obtained from a basic construction, which is nowadays often called the *Coppersmith-Winograd tensor*. Coppersmith and Winograd showed that analyzing this tensor gives the upper bound $\omega < 2.388$, and next showed that analyzing the second power of this tensor gives the improved upper bound $\omega < 2.376$.

A natural question, already mentioned in Coppersmith and Winograd's paper [10], was whether higher powers of the Coppersmith-Winograd tensor can lead to further improvement to the complexity of matrix multiplication. Most efforts to investigate this direction quickly stopped after discovering that the third power does not seem to lead to any further improvement. More than twenty year later, however, Stothers [24] (see also [11]) and Vassilevska Williams [28] showed that the fourth power does give an improvement: the fourth power leads to the upper bound $\omega \leq 2.373$. The technically challenging analysis of the fourth power was made possible by the introduction of powerful general recursive techniques to analyze powers of tensors. Extending these techniques, Vassilevska Williams [28] and then Le Gall [18] succeeded in analyzing higher powers up to the 32nd power, which gave additional small improvements and lead to the current best known upper bound on the exponent of square matrix multiplication $\omega < 2.3728639$. Table 1 summarises all these results. Ambainis et al. [3] finally showed that further improving this upper bound will be hard: they showed that analyzing higher powers of the Coppersmith-Winograd tensor (e.g., powers 64, 128,...) using the same methodology cannot give any further significant improvement on ω (in particular it cannot lead to a proof of the popular conjecture $\omega = 2$).

Besides square matrix multiplication, rectangular matrix multiplication plays a central role in many algorithms as well. In addition to natural applications to computational problems in linear algebra, typical examples of application include the construction of

*This work is partially supported by ERC QCC, ANR grant RDAM, JSPS Grant-in-Aid for Young Scientists (A) No. 16H05853 and JSPS Grant-in-Aids for Scientific Research (A) No. 15H01677 and 16H01705.

[†]Graduate School of Informatics, Kyoto University

[‡]IRIF, Université Paris Diderot

fast algorithms for the all-pairs shortest paths problem [2, 21, 31, 34, 35], dynamic computation of the transitive closure [12, 22], detection of subgraphs [29, 32], speed-up of sparse square matrix multiplication [4, 15, 33] and algorithms for bounded-difference min-plus square matrix multiplication [6]. Rectangular matrix multiplication has also been used in computational complexity [1, 20, 30] and computational geometry [14, 15].

The typical problem considered when studying rectangular matrix multiplication is computing the product of an $n \times \lceil n^k \rceil$ matrix by an $\lceil n^k \rceil \times n$ matrix, for some parameter $k \geq 0$.¹ In analogy to the square case, the exponent of rectangular matrix multiplication, denoted $\omega(k)$, is defined as the minimum value such that this product can be computed using $O(n^{\omega(k)+\epsilon})$ arithmetic operations for any constant $\epsilon > 0$. Also note that for $k = 1$ (i.e., for square matrices), we have $\omega(1) = \omega$.

Coppersmith [8] showed in 1982 that $\omega(0.172) = 2$. This surprising result means that the product of an $n \times \lceil n^{0.172} \rceil$ matrix by an $\lceil n^{0.172} \rceil \times n$ matrix can be computed in time almost linear in the size of the output (which contains n^2 entries). This discovery led to the introduction of the following quantity α :

$$\alpha = \sup\{k \mid \omega(k) = 2\}.$$

Since proving that $\alpha = 1$ is equivalent to proving that $\omega = 2$, the quantity α is sometimes called the dual exponent of matrix multiplication. Coppersmith's result [8] then corresponds to the bound $\alpha \geq 0.172$. Coppersmith [9] later showed that $\alpha > 0.29462$ by analyzing the Coppersmith-Winograd tensor in the context of rectangular matrix multiplication. Fifteen years later, Le Gall [17] showed that the second power of the Coppersmith-Winograd tensor can also be analyzed in the context of rectangular matrix multiplication, which led to the improved lower bound $\alpha > 0.30298$. This analysis was actually much more general and gave bounds on $\omega(k)$ that improved prior bounds [13, 16] for any $k \neq 1$. (For $k = 1$, i.e., square matrix multiplication, this approach recovered the upper bound $\omega < 2.376$ from [10]). The results from [17] are presented in Table 2.

1.2 Our results. In view of the recent progress in square matrix multiplication algorithms obtained by analyzing higher powers of the Coppersmith-Winograd

¹Note that a basic result in algebraic complexity theory states that the algebraic complexities of the following three problems are the same: computing the product of an $n \times \lceil n^k \rceil$ matrix by an $\lceil n^k \rceil \times n$ matrix, computing the product of an $\lceil n^k \rceil \times n$ matrix by an $n \times n$ matrix, and computing the product of an $n \times n$ matrix by an $n \times \lceil n^k \rceil$ matrix. In this paper for concreteness we discuss only the first type of products, but all our bounds naturally hold for the two other types as well.

m	Upper bound	Reference
1	$\omega < 2.3871900$	Coppersmith-Winograd [10]
2	$\omega < 2.3754770$	Coppersmith-Winograd [10]
4	$\omega < 2.372927$	Vassilevska Williams [28]
8	$\omega < 2.372873$	Vassilevska Williams [28]
16	$\omega < 2.3728640$	Le Gall [18]
32	$\omega < 2.3728639$	Le Gall [18]

Table 1: Upper bounds on ω obtained by analyzing the m -th power of the Coppersmith and Winograd tensor.

k	upper bound on $\omega(k)$	k	upper bound on $\omega(k)$
0.30298	2	0.85	2.260830
0.31	2.000063	0.90	2.298048
0.32	2.000371	0.95	2.336306
0.33	2.000939	1.00	2.375477
0.34	2.001771	1.10	2.456151
0.35	2.002870	1.20	2.539392
0.40	2.012175	1.30	2.624703
0.45	2.027102	1.40	2.711707
0.50	2.046681	1.50	2.800116
0.5302	2.060396	1.75	3.025906
0.55	2.070063	2.00	3.256689
0.60	2.096571	2.50	3.727808
0.65	2.125676	3.00	4.207372
0.70	2.156959	4.00	5.180715
0.75	2.190087	5.00	6.166736
0.80	2.224790		

Table 2: Upper bounds from [17] on the exponent of the multiplication of an $n \times n^k$ matrix by an $n^k \times n$ matrix, obtained by analyzing the second power of the Coppersmith-Winograd tensor.

tensor, it is natural to ask whether the same approach can be applied to obtain further improvements on the complexity of rectangular matrix multiplication as well. We investigate this question in this paper, and present a framework to extend the analysis of higher powers to the case of rectangular matrix multiplication. We concretely focus on the analysis of the fourth power of the Coppersmith-Winograd tensor and show that this analysis leads to non-negligible improvements. The new upper bounds we obtain on the exponent of rectangular matrix multiplication $\omega(k)$ are given in Table 3 and the values for $k \leq 1$ are plotted in Figure 1.² We obtain in particular the new lower bound

$$\alpha \geq 0.31389$$

²Note that the curve of Figure 1 has the same shape as the curve for the second power given in [17].

k	upper bound on $\omega(k)$	k	upper bound on $\omega(k)$
0.31389	2	0.85	2.258317
0.32	2.000064	0.90	2.295544
0.33	2.000448	0.95	2.333789
0.34	2.001118	1.00	2.372927
0.35	2.001957	1.10	2.453481
0.40	2.010314	1.20	2.536550
0.45	2.024801	1.30	2.621644
0.50	2.044183	1.40	2.708400
0.5286	2.057085	1.50	2.796537
0.55	2.067488	1.75	3.021591
0.60	2.093981	2.00	3.251640
0.65	2.123097	2.50	3.721503
0.70	2.154399	3.00	4.199712
0.75	2.187543	4.00	5.171210
0.80	2.222256	5.00	6.157233

Table 3: Our upper bounds on the exponent of the multiplication of an $n \times n^k$ matrix by an $n^k \times n$ matrix, obtained by analyzing the fourth power of the Coppersmith-Winograd tensor.

on the dual exponent of matrix multiplication, as stated in the following theorem.

THEOREM 1.1. *The product of an $n \times \lceil n^{0.31389} \rceil$ matrix by an $\lceil n^{0.31389} \rceil \times n$ matrix can be computed with $O(n^{2+\epsilon})$ arithmetic operations for any constant $\epsilon > 0$.*

This new bound improves the previous best known lower bound $\alpha > 0.30298$ by Le Gall [17]. For other values of k as well, our new upper bounds on $\omega(k)$ are systematically better than those of [17], as can be seen by comparing Table 2 and Table 3. For instance we obtain $\omega(3) \leq 4.199712$, which improves the previous upper bound $\omega(3) \leq 4.207372$. Note that for $k = 1$ (i.e., for square matrix multiplication), we obtain the same upper bound $\omega \leq 2.372927$ on the exponent of square matrix multiplication as the bound obtained by analyzing the fourth power [11, 18, 24, 28]. Indeed, for $k = 1$ our analysis becomes essentially the same as the analysis for the square case in those prior works.

A surprising, or at least unexpected, aspect of the result of Theorem 1.1 is that the improvement from the second power to the fourth power (from $\alpha > 0.30298$ to $\alpha > 0.31389$) exceeds the improvement known from the first power to the second power (from $\alpha > 0.29462$ to $\alpha > 0.30298$). This is completely different from the improvements achieved on ω when analyzing successive powers of the Coppersmith-Winograd tensor, which are decreasing, as summarized in Table 1. Actually, all the numerical results we have obtained confirm that for

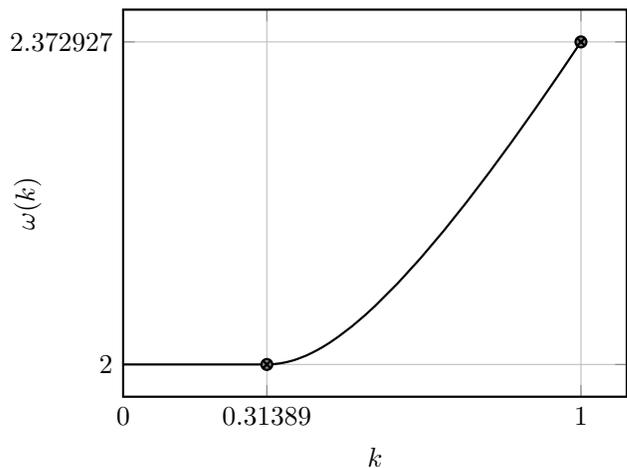


Figure 1: Our upper bounds on the exponent of the multiplication of an $n \times n^k$ matrix by an $n^k \times n$ matrix.

any fixed value of k the improvements on $\omega(k)$ decrease similarly to the square case when analyzing successive powers. For instance for $k = 0.8$ and $k = 2$ the first power gives $\omega(0.8) < 2.2356$ and $\omega(2) < 3.2699$; by examining Tables 2 and 3 we observe that the improvement is larger from the first power to the second power. The situation happens to be different, however, for lower bounds on α . Since the curves representing the upper bounds on $\omega(k)$ have horizontal asymptotes at the lower bound on α (see Figure 1 of the present paper and Figure 1 in [17]), even small improvements on $\omega(k)$ can lead to fairly significant improvements on α , as our results show.

The most pressing question is now to investigate what will happen for even higher powers of the Coppersmith-Winograd tensor (e.g., power 8 or 16). We believe that this question is important since, besides its theoretical interest, further significant improvements for α may be obtained in this way. A concrete approach would be to adapt to the rectangular case the numerically efficient methods based on convex optimization developed, in the setting of square matrix multiplication, to study high powers of the Coppersmith-Winograd tensor [18]. In the other direction, it may be possible to show some limitations on the improvements achievable when studying higher powers, by generalizing the recent approach developed for the square case [3].

Applications of our results. Our new bounds can be used to improve essentially all the known algorithms based on rectangular matrix multiplication algorithms (e.g., the algorithms in [2, 14, 15, 21, 31, 29, 32, 34, 35]). Following [17], we discuss below one concrete example.

Zwick [35] has shown how to use rectangular matrix

multiplication to compute the all-pairs shortest paths in weighted directed graphs where the weights are bounded integers. The time complexity obtained by Zwick for graphs with constant weights is $O(n^{2+\mu+\epsilon})$, for any constant $\epsilon > 0$, where μ is the solution of the equation $\omega(\mu) = 1 + 2\mu$. The results from [17] (see Table 2) show that $\omega(0.5302) < 2.0604$, which gives the upper bound $\mu < 0.5302$. The results of the present paper (see Table 3) show that $\omega(0.5286) < 2.0572$, which gives the upper bound $\mu < 0.5286$.

1.3 Overview of our approach. Before presenting an overview of the techniques used in this paper, we give an informal description of algebraic complexity theory (a more detailed presentation of these notions is given in Section 2).

Trilinear forms and the asymptotic sum inequality. The matrix multiplication of an $m \times n$ matrix by an $n \times p$ matrix can be represented by the following trilinear form, denoted as $\langle m, n, p \rangle$:

$$\langle m, n, p \rangle = \sum_{r=1}^m \sum_{s=1}^n \sum_{t=1}^p x_{rs} y_{st} z_{rt},$$

where x_{rs} , y_{st} and z_{rt} are formal variables. This form can be interpreted as follows: the (r, t) -th entry of the product of an $m \times n$ matrix M by an $n \times p$ matrix M' can be obtained by setting $x_{ij} = M_{ij}$ for all $(i, j) \in [m] \times [n]$ and $y_{ij} = M'_{ij}$ for all $(i, j) \in [n] \times [p]$, setting $z_{rt} = 1$ and setting all the other z -variables to zero. One can then think of the z -variables as formal variables used to record the entries of the matrix product.

More generally, a trilinear form t is represented as

$$t = \sum_{u \in A} \sum_{v \in B} \sum_{w \in C} t_{uvw} x_u y_v z_w.$$

where A, B and C are three sets, x_u , y_v and z_w are formal variables and the t_{uvw} 's are coefficients in a field \mathbb{F} . The rank of the trilinear form t , denoted $R(t)$, represents the number of multiplications needed for the computation. The border rank of t , denoted $\underline{R}(t)$, is a generalization of the concept of rank, which is also related to the complexity of computing the tensor.

A sum $\sum_i t_i$ of trilinear forms is a direct sum if the t_i 's do not share variables. Informally, Schönhage's asymptotic sum inequality [23] for rectangular matrix multiplication states that, if the form t can be converted into a direct sum of c trilinear forms, each form being isomorphic to $\langle m, m, m^k \rangle$, then

$$c \cdot m^{\omega(k)} \leq \underline{R}(t).$$

This implies that to obtain good upper bounds on the exponent of rectangular matrix multiplication, it is

enough to find a tensor t of low border rank that can be converted into many independent (i.e., not sharing any variables) products of large enough rectangular matrices.

Overview of the analysis of the second power. We now give a brief overview of the analysis of the second power of the Coppersmith-Winograd tensor given in [17] to derive the upper bounds on $\omega(k)$ of Table 2. The Coppersmith-Winograd tensor is a trilinear form F_q introduced in [10]. Here q is a parameter (concretely, q is an integer between 2 and 10). Its second power $F_q \otimes F_q$ can actually be written as a sum of fifteen terms T_{uvw} :

$$F_q \otimes F_q = \sum_{\substack{0 \leq u, v, w \leq 4 \\ u+v+w=4}} T_{uvw}.$$

In order to apply Schönhage's asymptotic sum inequality, this sum must first be converted into a direct sum. This is done using a powerful general technique known as the laser method, first introduced by Strassen [26] and then successively generalized and refined [10, 11, 13, 16, 17, 18, 24, 28]. The first step is to take the N -th tensor product of the basic construction, where N is a large integer, and then zero variables so that the remaining terms do not share variables. Since we want each remaining term to be isomorphic to a rectangular matrix product in order to obtain an upper bound on $\omega(k)$ via Schönhage's asymptotic sum inequality, the choice of zeroed variables has to be done carefully. The laser method allows us, for any choice of the fifteen parameters $a_{uvw} \in [0, 1]$ satisfying specific constraints, to convert the N -th tensor product of the basic construction into a direct sum of many terms (the number of these terms depending on the values of the a_{uvw} 's), each isomorphic to

$$(1.1) \quad \bigotimes_{\substack{0 \leq u, v, w \leq 4 \\ u+v+w=4}} T_{uvw}^{\otimes a_{uvw} N}.$$

The next step is to analyze each term (1.1) and show that it corresponds to a direct sum of matrix products of the form $\langle m, m, m^k \rangle$ (the number of terms in the direct sum and the value of m will depend on the values of the a_{uvw} 's and q). Some of the T_{uvw} 's (more precisely, all the T_{uvw} 's except T_{112} , T_{121} and T_{211}) can be analyzed in a straightforward way, since they correspond to matrix products. The main technical contribution of the approach from [17] was to show that each of the remaining three terms can be converted into a large number of objects called "C-tensors" in Strassen's terminology [27]. This conversion is done again via the laser method, which introduces

additional parameters. Finally, Ref. [17] explained how to convert these \mathcal{C} -tensors into a direct sum of matrix multiplication tensors. Combining the analysis of these fifteen terms shows that (1.1) corresponds to a direct sum of matrix products of the form $\langle m, m, m^k \rangle$, as wanted. Schönhage’s asymptotic sum inequality then gives the upper bound on $\omega(k)$ presented in Table 2 by numerically optimizing the choice of the parameters (the choice of q , the a_{uvw} ’s and the additional parameters arising in the second extraction).

Overview of our analysis of the fourth power. The fourth power of the Coppersmith-Winograd tensor can be written as a sum of 45 terms T_{uvw} :

$$F_q^{\otimes 4} = \sum_{\substack{0 \leq u, v, w \leq 8 \\ u+v+w=8}} T_{uvw}.$$

For conciseness, this tensor will be denoted F through the paper. Similarly to the analysis of the second power, the laser method allows us, for any choice of parameters $a_{uvw} \in [0, 1]$ satisfying specific constraints, to convert the N -th tensor product of the basic construction into a direct sum of many terms, each isomorphic to

$$(1.2) \quad \bigotimes_{\substack{0 \leq u, v, w \leq 8 \\ u+v+w=8}} T_{uvw}^{\otimes a_{uvw} N}.$$

We call this process the *first extraction*, which is explained in detail in Section 3. Note that while this extraction is more complicated than for the second power since the number of variables is larger and deriving the constraints that the parameters should satisfy is more complex, conceptually the analysis is fairly standard.

The main technical contribution of this work is a methodology to analyze each term (1.2). A natural strategy would be to mimic the analysis done in [17] for the second power and analyze each component T_{uvw} individually. While this leads to some improvement over the second power when k is close to 1 (in particular, this leads to the same upper bound $\omega < 2.372927$ as in prior works analyzing the fourth power in the context of square matrix multiplication [24, 28]), this strategy does not give any improvement for smaller values of k (in particular, no improved lower bound on the dual exponent of matrix multiplication α). Our strategy, instead, is to analyze all the terms T_{uvw} together via the laser method. As in the term-by-term analysis done for the second power in [17], this introduces a set of new parameters for each term and a set of constraints that these parameters should satisfy. A difference is that now some of the constraints are global: they can involve the parameters of all the 45 terms. We call this process the

second extraction, which is explained in detail in Section 4. Note that this methodology appears to be more powerful than the term-by-term conversion to \mathcal{C} -tensors done in [17]: First, as already mentioned, the latter approach does not seem to lead to any improvement on α for the fourth power. Second, our new methodology, when applied to the analysis of the second power in replacement of the conversion into \mathcal{C} -tensors done in [17], already leads to upper bounds on $\omega(k)$ slightly better than those found in [17] for some values of k (more precisely, we observed such improvements for values in the range $k \in [0.37, 0.46]$).

The second extraction outlined in the previous paragraph actually does not completely analyze (1.2): it simply decomposes each term T_{uvw} into a direct sum of products of the fifteen terms arising in the analysis of the second power. To complete the analysis, we recursively apply the same strategy as for the second extraction and analyse the contribution of all these fifteen terms together, again using the laser method (which introduce two additional parameters). We call this process the *third extraction*, which is explained in detail in Section 5.

Finally, combining our three extractions, we conclude that the tensor $F^{\otimes N}$ can be converted into a direct sum of c trilinear forms, each form being isomorphic to $\langle m, m, m^k \rangle$, for some values c and m depending on all the parameters introduced. Applying Schönhage’s asymptotic sum inequality then gives an inequality involving $\omega(k)$ and all these parameters (the formal statement is Theorem 6.1 in Section 6). Optimizing numerically the choice of parameters, under the constraints derived on those parameters, gives the upper bounds of Table 3 and the lower bound on α of Theorem 1.1.

2 Preliminaries

We present various known results and tools related to matrix multiplication. Two good references for an extensive treatment of this topic are [7] and [5].

2.1 General notations and definitions. In this paper we will use, for any positive integer n , the notation $\llbracket 1, n \rrbracket$ to represent the set $\{1, \dots, n\}$. Given a finite set X we denote $\mathcal{P}(X)$ the set of all function $a: X \rightarrow]0, 1[\cap \mathbb{Q}$ such that $\sum_{x \in X} a(x) = 1$. Note that the functions in $\mathcal{P}(X)$ are simply probability distributions on X where the probabilities are rational and strictly between 0 and 1.

We define the notion of *type*. A type can be seen as a frequency vector.

DEFINITION 2.1. Given $N \in \mathbb{N}$, a finite set U , two

mappings $u : \llbracket 1, N \rrbracket \rightarrow U$ and $t \in \mathcal{P}(U)$, we say that u is of type t if $|u^{-1}(x)| = t(x)N$ holds for all $x \in U$.

2.2 Tensors, matrix multiplication and the asymptotic sum inequality.

Let \mathbb{F} be a field, let u, v, w be three positive integers, and let $U = \mathbb{F}^u, V = \mathbb{F}^v$ and $W = \mathbb{F}^w$. A tensor t of format (u, v, w) is an element of $U \otimes V \otimes W = \mathbb{F}^{u \times v \times w}$, where \otimes denotes the tensor product. Fix a base (x_i) of U , a base (y_j) of V and a base (z_k) of W . We write $x_i y_j z_k$ as a short cut for $x_i \otimes y_j \otimes z_k$. The family $(x_i y_j z_k)$ is a base of $U \otimes V \otimes W$, and thus t can be written in this base as $t = \sum_{ijk} t_{ijk} x_i y_j z_k$, where the t_{ijk} are coefficients in \mathbb{F} .

For a tensor written under this form, we call the (x_i) the x variables, and similarly we call the (y_j) and the (z_k) the y variables and the z variables.

Let $t \in U \otimes V \otimes W$ and $t' \in U' \otimes V' \otimes W'$ be two tensors. The direct sum $t \oplus t'$, is a tensor in $(U \oplus U') \otimes (V \oplus V') \otimes (W \oplus W')$. The tensor product $t \otimes t'$ is a tensor in $(U \otimes U') \otimes (V \otimes V') \otimes (W \otimes W')$. For any positive c , we will denote the tensor $t \oplus \dots \oplus t$ (with c occurrences of t) by $c \cdot t$ and the tensor $t \otimes \dots \otimes t$ (with c occurrences of t) by $t^{\otimes c}$.

The tensor representing the multiplication of an $m \times n$ matrix by an $n \times p$ matrix over the field \mathbb{F} , denoted $\langle m, n, p \rangle$, is the tensor of format (mn, np, mp) defined as

$$\langle m, n, p \rangle = \sum_{ijk} t_{ijk} x_i y_j z_k$$

where i spans $\llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket$, j spans $\llbracket 1, n \rrbracket \times \llbracket 1, p \rrbracket$, k spans $\llbracket 1, m \rrbracket \times \llbracket 1, p \rrbracket$ and

$$t_{ijk} = \begin{cases} 1 & \text{if } i = (r, s), j = (s, t), k = (r, t) \text{ for some} \\ & \text{integers } (r, s, t) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket \times \llbracket 1, p \rrbracket \\ 0 & \text{otherwise.} \end{cases}$$

We also consider the tensor of format (n, n, n) which represents n independent scalar products. It is denoted by $\langle n \rangle$ and is defined as $\langle n \rangle = \sum_{i=1}^n x_i y_i z_i$.

The following definitions allow us to relate the properties of different tensors. Let λ denote an indeterminate, and $\mathbb{F}[\lambda]$ the space of polynomials in λ with coefficients in \mathbb{F} .

DEFINITION 2.2. Let $t \in \mathbb{F}^{u \times v \times w}$ and $t' \in \mathbb{F}[\lambda]^{u' \times v' \times w'}$ be two tensors. We say that t' is a restriction of t , and denote $t' \leq t$, if there are linear maps $\alpha : \mathbb{F}^u \rightarrow \mathbb{F}[\lambda]^{u'}$, $\beta : \mathbb{F}^v \rightarrow \mathbb{F}[\lambda]^{v'}$, $\gamma : \mathbb{F}^w \rightarrow \mathbb{F}[\lambda]^{w'}$ such that $(\alpha \otimes \beta \otimes \gamma)(t) = t'$ where $(\alpha \otimes \beta \otimes \gamma)$ is the linear map $\mathbb{F}^u \otimes \mathbb{F}^v \otimes \mathbb{F}^w \rightarrow \mathbb{F}[\lambda]^{u'} \otimes \mathbb{F}[\lambda]^{v'} \otimes \mathbb{F}[\lambda]^{w'}$ obtained by taking the tensor product of α, β, γ .

The intuition behind this notion is that the restriction of a tensor is easier to compute than the original tensor, in the sense that an algorithm computing a tensor t can be converted into an algorithm computing a tensor $t' \leq t$ with the same complexity.

DEFINITION 2.3. Let $t \in \mathbb{F}[\lambda]^{u \times v \times w}$ and $t' \in \mathbb{F}^{u \times v \times w}$ be two tensors. We say that t' is an approximation of t , if there exists a tensor $t'' \in \mathbb{F}[\lambda]^{u \times v \times w}$ and some $s \in \mathbb{N}$ such that $t = \lambda^s t' + \lambda^{s+1} t''$. We may also write : $t = \lambda^s t' + \mathcal{O}(\lambda^{s+1})$.

This is analogous to the notion of approximate computation.

DEFINITION 2.4. Let $t \in \mathbb{F}^{u \times v \times w}$ and $t' \in \mathbb{F}^{u' \times v' \times w'}$ be two tensors. We say that t' is a degeneration of t , and denote $t' \leq t$, if t' is an approximation of a restriction of t .

Note that by definition, $t' \leq t \implies t' \leq t$. The notion of degeneration can be seen as an approximate conversion. It has the following property.

PROPOSITION 2.1. (PROPOSITION 15.25 IN [7]) Let t_1, t'_1, t_2 and t'_2 be four tensors. Suppose that $t'_1 \leq t_1$ and $t'_2 \leq t_2$. Then $t'_1 \oplus t'_2 \leq t_1 \oplus t_2$ and $t'_1 \otimes t'_2 \leq t_1 \otimes t_2$.

DEFINITION 2.5. Let t be a tensor. The border rank of t is $\underline{R}(t) = \min\{r \in \mathbb{N} \mid t \leq \langle r \rangle\}$.

The notion of border rank enables us to formally define the exponent of rectangular matrix multiplication, as follows. For any $k \geq 0$,

$$\omega(k) = \inf\{\beta \mid \underline{R}(\langle n, \lfloor n^k \rfloor, n \rangle) = O(n^\beta)\}.$$

The exponent of square matrix multiplication is $\omega = \omega(1)$.

Similarly to almost all recent works on matrix multiplications, our main tool for proving lower bounds on $\omega(k)$ will be Schönhage's asymptotic sum inequality [23] (see [13, 16, 17, 19] for the version of the inequality given below).

THEOREM 2.1. (ASYMPTOTIC SUM INEQUALITY) Let k, m and c be three positive integers. Let t be a tensor such that $c \cdot \langle m, m, m^k \rangle \leq t$. Then

$$c \cdot m^{\omega(k)} \leq \underline{R}(t).$$

2.3 The fourth power of Coppersmith-Winograd tensor.

For any positive integer q , the Coppersmith-Winograd tensor [10] is the tensor of

format $(q + 2, q + 2, q + 2)$ defined as

$$F_q = \sum_{i=1}^q (x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0) + x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + x_{q+1} y_0 z_0.$$

Coppersmith and Winograd showed that $\underline{R}(F_q) \leq q + 2$.

They also considered the square of this tensor. For any $t \in \mathbb{N}$, define the set

$$S_t = \{(i, j, k) \in \mathbb{N}^3 \mid i + j + k = t\}.$$

Define the tensors $x_{ii'} y_{jj'} z_{kk'} = x_i y_j z_k \otimes x_{i'} y_{j'} z_{k'}$. By regrouping terms, we can write

$$(F_q)^{\otimes 2} = \sum_{(i,j,k) \in S_4} T_{ijk}$$

where

$$\begin{aligned} T_{004} &= x_{0,0} y_{0,0} z_{q+1,q+1} \\ T_{013} &= \sum_{i=1}^q x_{0,0} y_{i,0} z_{i,q+1} + \sum_{k=1}^q x_{0,0} y_{0,k} z_{q+1,k} \\ T_{022} &= x_{0,0} y_{q+1,0} z_{0,q+1} + x_{0,0} y_{0,q+1} z_{q+1,0} + \sum_{i,k=1}^q x_{0,0} y_{i,k} z_{i,k} \\ T_{112} &= \sum_{i=1}^q x_{i,0} y_{i,0} z_{0,q+1} + \sum_{k=1}^q x_{0,k} y_{0,k} z_{q+1,0} + \sum_{i,k=1}^q x_{i,0} y_{0,k} z_{i,k} + \sum_{i,k=1}^q x_{0,k} y_{i,0} z_{i,k} \end{aligned}$$

and the other eleven terms are obtained by permuting the indexes of the x variables, the y variables and z variables in the above expressions (e.g., $T_{040} = x_{0,0} y_{q+1,q+1} z_{0,0}$ and $T_{400} = x_{q+1,q+1} y_{0,0} z_{0,0}$). Note that $\underline{R}(F_q^{\otimes 2}) \leq (q + 2)^2$ from the submultiplicativity of the border rank.

Let us now consider the fourth power of the Coppersmith-Winograd tensor F_q (already studied, in the context of square matrix multiplication, in Refs. [11, 24, 18, 28]). For any $(i, j, k) \in S_8$ define the set

$$S_{ijk} = \{((u, v, w), (u', v', w')) \in S_4 \times S_4 \mid u + u' = i, v + v' = j, w + w' = k\}$$

and the tensor

$$T_{ijk} = \sum_{((u,v,w),(u',v',w')) \in S_{ijk}} T_{uvw} \otimes T_{u'v'w'}.$$

By regrouping terms, the fourth power of F_q , which hereafter we simply denote F (the value of q will be

implicit until the very end of the paper), can be written as

$$\begin{aligned} F &= (F_q)^{\otimes 4} \\ &= (F_q)^{\otimes 2} \otimes (F_q)^{\otimes 2} \\ &= \left(\sum_{(u,v,w) \in S_4} T_{uvw} \right) \otimes \left(\sum_{(u',v',w') \in S_4} T_{u'v'w'} \right) \\ &= \sum_{((u,v,w),(u',v',w')) \in S_4^2} T_{uvw} \otimes T_{u'v'w'} \\ &= \sum_{(i,j,k) \in S_8} \sum_{((u,v,w),(u',v',w')) \in S_{ijk}} T_{uvw} \otimes T_{u'v'w'} \\ &= \sum_{(i,j,k) \in S_8} T_{ijk}. \end{aligned}$$

Note that $\underline{R}(F) \leq (q + 2)^4$, again from the submultiplicativity of the border rank.

When later working with the terms T_{ijk} , we will sometimes consider the equivalent decomposition

$$T_{ijk} = \sum_{(u,v,w) \in \bar{S}_{ijk}} V_{ijk}[uvw]$$

where

$$\bar{S}_{ijk} = \{(u, v, w) \in S_4, \exists (u', v', w') \in S_4 \mid u + u' = i, v + v' = j, w + w' = k\}$$

and

$$\forall (u, v, w) \in \bar{S}_{ijk}, V_{ijk}[uvw] = T_{uvw} \otimes T_{i-u, j-v, k-w}.$$

For any integer $N \in \mathbb{N}$ and triple (I, J, K) with $I = (I(1), \dots, I(N))$, $J = (J(1), \dots, J(N))$, $K = (K(1), \dots, K(N))$ such that $(I(1), J(1), K(1)), \dots, (I(N), J(N), K(N)) \in S_8$, define

$$T_{IJK} = T_{I(1)J(1)K(1)} \otimes \dots \otimes T_{I(N)J(N)K(N)}.$$

Notice that we define T_{IJK} only for triple (I, J, K) from the set

$$\{(I, J, K), \forall l \in [1, N], (I(l), J(l), K(l)) \in S_8\} \cong S_8^N.$$

We can then write:

$$F^{\otimes N} = \sum_{(I,J,K) \in S_8^N} T_{IJK}.$$

Finally, for any triple $(a, b, c) \in S_8$ and any triple $(I, J, K) = (I(l), J(l), K(l))_{l \in [1, N]} \in \bar{S}_{abc}^N$, we define

$$V_{abc}[IJK] = \bigotimes_{l=1}^N V_{abc}[I(l)J(l)K(l)].$$

2.4 Extraction from a tensor. In this subsection we explain our main tool to realize an *extraction* from a sum of tensors. An extraction consists in assigning some variables to zero in a tensor (thus eliminating all their contributions to the sum). If a tensor T' is extracted from T , then $T' \leq T$ trivially holds. Our primary goal is to guarantee that the resulting tensor T' is a direct sum of isomorphic tensors, so that the asymptotic sum inequality can be used.

All recent progresses on square or rectangular matrix multiplication have been obtained by performing extractions based on the so-called laser method [10, 11, 13, 16, 17, 18, 24, 26, 28]. Le Gall [17] introduced the following convenient framework to interpret such reductions in the rectangular case. In this framework, a sum of tensors corresponds to a graph whose vertices are the tensors in the sum. There is an edge between two vertices in the graph if and only if the two corresponding terms in the sum of tensors share a variable. Let G denote this graph, and U denote its set of vertices. Zeroing a term in the sum corresponds to removing one vertex from the graph. As mentioned above, however, terms can be zeroed only by zeroing the variables it contains. This means that such a zeroing operation may actually remove more than one vertex from the graph. Extracting a direct sum from the original tensor is then equivalent to removing vertices from the graph by such zeroing operations and reaching an edgeless graph. When using this methodology, we will like to additionally guarantee that the vertices remaining in the final graph are from a specified subset $U^* \subseteq U$. Concretely, the set U^* will be the set of vertices of terms matching a certain type, which will ensure that all the tensors remaining after the extraction are of this type. In our extractions we will use the following theorem from [17], which is tailored for this goal and was already used for the analysis of the second power of the Coppersmith-Winograd tensor.

THEOREM 2.2. (THEOREM 4.2 IN [17]) *Let τ be a fixed positive integer. Let N be a large integer and define the set*

$$\Lambda = \{(I, J, K) \in [\tau]^N \times [\tau]^N \times [\tau]^N \mid I_\ell + J_\ell + K_\ell = \tau \text{ for all } \ell \in \{1, \dots, N\}\}.$$

Define the following three coordinate functions $f_1, f_2, f_3: [\tau]^N \times [\tau]^N \times [\tau]^N \rightarrow [\tau]^N$ by

$$\begin{aligned} f_1((I, J, K)) &= I \\ f_2((I, J, K)) &= J \\ f_3((I, J, K)) &= K \end{aligned}$$

Let U be a subset of Λ such that there exist integers $\mathcal{N}_1, \mathcal{N}_2$ and \mathcal{N}_3 for which the following property holds:

for any $I \in [\tau]^N$,

$$\begin{aligned} |\{u \in U \mid f_1(u) = I\}| &\in \{0, \mathcal{N}_1\} \\ |\{u \in U \mid f_2(u) = I\}| &\in \{0, \mathcal{N}_2\} \\ |\{u \in U \mid f_3(u) = I\}| &\in \{0, \mathcal{N}_3\}. \end{aligned}$$

Let $\mathcal{T}_1 = |f_1(U)|$, $\mathcal{T}_2 = |f_2(U)|$ and $\mathcal{T}_3 = |f_3(U)|$. Let G be the (simple and undirected) graph with vertex set U in which two distinct vertices u and v are connected if and only if there exists one index $i \in \{1, 2, 3\}$ such that $f_i(u) = f_i(v)$.

Assume there exists a set $U^ \subseteq U$ such that*

- $|f_i(U^*)| = \mathcal{T}_i$ for each $i \in \{1, 2, 3\}$;
- *there exist integers $\mathcal{N}_1^*, \mathcal{N}_2^*$ and \mathcal{N}_3^* such that $|\{u \in U \mid f_i(u) = I\}| = \mathcal{N}_i^*$ if and only if $|\{u \in U^* \mid f_i(u) = I\}| = \mathcal{N}_i^*$ for each $I \in [\tau]^N$ and each $i \in \{1, 2, 3\}$.*

Define a removal operation as removing all the vertices u (if any) such that $f_i(u) = I$, for a fixed sequence $I \in [\tau]^N$ and a fixed position $i \in \{1, 2, 3\}$. Then, for any constant $\epsilon > 0$, the graph G can be converted, with only removal operations, into an edgeless graph with

$$\Omega\left(\frac{\mathcal{T}_1 \mathcal{N}_1^*}{(\mathcal{N}_1 + \mathcal{N}_2 + \mathcal{N}_3)^{1+\epsilon}}\right)$$

vertices, all of them being in U^ .*

3 First extraction

In this section we describe our first extraction.

Let us consider any function $a \in \mathcal{P}(S_8)$. For any $(i, j, k) \in S_8$ we will often write $a(ijk)$ instead of $a(i, j, k)$. Given a , we define the following three mappings:

$$A: \llbracket 0, 8 \rrbracket \longrightarrow]0, 1[\quad i \longmapsto A(i) = \sum_{j, k \in \mathbb{N} \mid (i, j, k) \in S_8} a(ijk),$$

$$B: \llbracket 0, 8 \rrbracket \longrightarrow]0, 1[\quad j \longmapsto B(j) = \sum_{i, k \in \mathbb{N} \mid (i, j, k) \in S_8} a(ijk),$$

$$C: \llbracket 0, 8 \rrbracket \longrightarrow]0, 1[\quad k \longmapsto C(k) = \sum_{i, j \in \mathbb{N} \mid (i, j, k) \in S_8} a(ijk).$$

A, B, C are the projections of a on each of the three coordinates. We thus have

$$\sum_{i \in \llbracket 0, 8 \rrbracket} A(i) = \sum_{j \in \llbracket 0, 8 \rrbracket} B(j) = \sum_{k \in \llbracket 0, 8 \rrbracket} C(k) = 1.$$

We are going to realize a first extraction from the tensor $F^{\otimes N}$, where N is such that $\forall (i, j, k) \in S_8, a(ijk)N \in \mathbb{N}$ (such an N always exists

since the values of a are rational). The goal is to zero variables so that, from the sum

$$F^{\otimes N} = \sum_{(I,J,K) \in S_8^N} T_{IJK},$$

we are left only with tensors T_{IJK} where (I, J, K) is of type a , i.e, tensors isomorphic to

$$\bigotimes_{(i,j,k) \in S_8} T_{ijk}^{a(ijk)N}.$$

We now explain how to achieve this goal.

The extraction. First notice that for any $(I, J, K), (I', J', K') \in S_8^N$ such that $I \neq I'$, the x variables in T_{IJK} and in $T_{I'J'K'}$ are disjoint. We set to zero all the x variables except the ones which appear in a T_{IJK} where I is of type A , that is to say we set to zero all variables which appear in a T_{IJK} with I not of type A . We are thus left with only the tensors T_{IJK} with I of type A .

We apply the same process for the y and z variables so that only remain the tensors T_{IJK} with I of type A , J of type B and K of type C . Let $[a]$ denote the set of mappings $\bar{a} : S_8 \rightarrow]0, 1[$ which have the same projections as a and satisfy $\forall (i, j, k) \in S_8, \bar{a}(ijk)N \in \mathbb{N}$. The tensors which remain are exactly the tensors T_{IJK} with (I, J, K) of type $\bar{a} \in [a]$.

The number of sequences $I \in \llbracket 0, 8 \rrbracket^N$ of type A is

$$\mathcal{T}_X = \binom{N}{(A(i)N)_{i \in \llbracket 0, 8 \rrbracket}}$$

as choosing a sequence I of type A is equivalent to choose the location of the $A(i)$ elements i for $i \in \llbracket 0, 8 \rrbracket$. Using the Stirling formula, we get, with the $A(i)$ fixed and $N \rightarrow \infty$,

$$\mathcal{T}_X = \Theta \left(\frac{1}{N^4} \left(\frac{1}{\prod_{i=0}^8 A(i)^{A(i)}} \right)^N \right).$$

Similarly, we define the number of sequences $J \in \llbracket 0, 8 \rrbracket^N$ of type B as \mathcal{T}_Y and the number of sequences $K \in \llbracket 0, 8 \rrbracket^N$ of type C as \mathcal{T}_Z .

For any fixed sequence I of type A , the number of remaining forms T_{IJK} of type a is

$$\mathcal{N}_X^* = \prod_{i=0}^8 \binom{A(i)N}{(a(ijk)N)_{j,k \in \mathbb{N} \setminus \{i,j,k\} \in S_8}},$$

while the total number of remaining forms T_{IJK} is

$$\mathcal{N}_X = \sum_{\bar{a} \in [a]} \prod_{i=0}^8 \binom{A(i)N}{(\bar{a}(ijk)N)_{j,k \in \mathbb{N} \setminus \{i,j,k\} \in S_8}}.$$

Define, the function g which associates to any mapping $x : S_8 \rightarrow]0, 1[$ $(i, j, k) \mapsto x(ijk)$ the value $g(x) = \left(\prod_{(i,j,k) \in S_8} x(ijk)^{x(ijk)} \right)^{-1}$. Using Stirling's

formula, and the fact that $|S_8| = \sum_{l=1}^9 l = 45$, we get that

$$\mathcal{N}_X^* = \Theta \left(\frac{\left[g(a) \prod_{i=0}^8 A(i)^{A(i)} \right]^N}{N^{18}} \right),$$

$$\mathcal{N}_X = \Theta \left(\sum_{\bar{a} \in [a]} \frac{\left[g(\bar{a}) \prod_{i=0}^8 A(i)^{A(i)} \right]^N}{N^{18}} \right).$$

Similarly, for any sequence J of type B , the number of remaining forms T_{IJK} of type a and the total number of remaining forms T_{IJK} are

$$\mathcal{N}_Y^* = \Theta \left(\frac{\left[g(a) \prod_{j=0}^8 B(j)^{B(j)} \right]^N}{N^{18}} \right),$$

$$\mathcal{N}_Y = \Theta \left(\sum_{\bar{a} \in [a]} \frac{\left[g(\bar{a}) \prod_{j=0}^8 B(j)^{B(j)} \right]^N}{N^{18}} \right),$$

and for any sequence K of type C , the number of remaining forms T_{IJK} of type a and the total number of remaining forms T_{IJK} are

$$\mathcal{N}_Z^* = \Theta \left(\frac{\left[g(a) \prod_{k=0}^8 C(k)^{C(k)} \right]^N}{N^{18}} \right),$$

$$\mathcal{N}_Z = \Theta \left(\sum_{\bar{a} \in [a]} \frac{\left[g(\bar{a}) \prod_{k=0}^8 C(k)^{C(k)} \right]^N}{N^{18}} \right).$$

Using the framework presented in Subsection 2.4, we get by Theorem 2.2 that for any $\epsilon > 0$ we can further extract from the remaining T_{IJK} a direct sum of

$$\Omega \left(\frac{\mathcal{T}_X \mathcal{N}_X^*}{(\mathcal{N}_X + \mathcal{N}_Y + \mathcal{N}_Z)^{1+\epsilon}} \right)$$

tensors T_{IJK} , all of which are of type a . We want the number of tensors is this direct sum to be high. For this to happen, we now formulate some conditions on a .

Conditions on a . We first want to find $a \in \mathcal{P}(S_8)$ such that $g(a) = \max_{\bar{a} \in [a]} g(\bar{a})$. Let us see g as a function from the set $]0, 1[^{45}$ to the set of positive real numbers. We want to find the maximum of g on the domain $[a] \subset]0, 1[^{45}$. For this, we will find the maximum of g on the domain $\widetilde{[a]}$, the set of mappings $S_8 \rightarrow]0, 1[$ which have the same projections as a . Note that $\widetilde{[a]} \supseteq [a]$, and that $\widetilde{[a]}$ is a convex subset of $]0, 1[^{45}$. Consider the function $\ln g : \widetilde{[a]} \rightarrow \mathbb{R}$. For any $\bar{a} \in \widetilde{[a]}$, $\ln g(\bar{a}) = \sum_{(i,j,k) \in S_8} -\bar{a}(ijk) \ln \bar{a}(ijk)$. Since $\ln g$ is a concave function on a convex domain, any critical point of $\ln g$ is a global maximum of $\ln g$, and thus of g . We express the conditions satisfied by the critical points below.

We first observe that $\ln g$ can actually be written as a (concave) function of only 21 variables, namely $\bar{a}(215), \bar{a}(224), \bar{a}(233), \bar{a}(242), \bar{a}(251), \bar{a}(260), \bar{a}(314), \bar{a}(323), \bar{a}(332), \bar{a}(341), \bar{a}(350), \bar{a}(413), \bar{a}(422), \bar{a}(431), \bar{a}(440), \bar{a}(512), \bar{a}(521), \bar{a}(530), \bar{a}(611), \bar{a}(620), \bar{a}(710)$. This is because $\ln g$ is defined on $\widetilde{[a]}$, and the elements of $\widetilde{[a]}$ have, by definition, the same projections as a , and thus for any $\bar{a} \in \widetilde{[a]}$ the $\bar{a}(ijk)$ satisfy the following system of linear equations:

$$\begin{aligned} \forall i \in \llbracket 0, 8 \rrbracket, A(i) &= \sum_{j,k \in \mathbb{N} \mid (i,j,k) \in S_8} \bar{a}(ijk), \\ \forall j \in \llbracket 0, 8 \rrbracket, B(j) &= \sum_{i,k \in \mathbb{N} \mid (i,j,k) \in S_8} \bar{a}(ijk), \\ \forall k \in \llbracket 0, 8 \rrbracket, C(k) &= \sum_{i,j \in \mathbb{N} \mid (i,j,k) \in S_8} \bar{a}(ijk). \end{aligned}$$

Resolving of the (homogeneous) linear system³ reduces the number of variables to 21, as claimed.

From now we will assume that a satisfies the symmetry condition

$$(C1) \quad \forall (i, j, k) \in S_8, a(ijk) = a(ikj).$$

Computing each one of the 21 partial differential equations $\frac{\partial \ln g}{\partial \bar{a}(ijk)} = 0$ leads, after simplification and by con-

dition (C1), to a system (C2) of 10 non linear equations:

$$\begin{aligned} 0 &= -\ln(\bar{a}(017)) + \ln(\bar{a}(026)) + \ln(\bar{a}(107)) - \ln(\bar{a}(125)) - \ln(\bar{a}(206)) + \ln(\bar{a}(215)), \\ 0 &= -\ln(\bar{a}(017)) + \ln(\bar{a}(026)) + \ln(\bar{a}(107)) - \ln(\bar{a}(116)) - \ln(\bar{a}(602)) + \ln(\bar{a}(611)), \\ 0 &= -\ln(\bar{a}(017)) + \ln(\bar{a}(035)) + \ln(\bar{a}(107)) - \ln(\bar{a}(134)) - \ln(\bar{a}(305)) + \ln(\bar{a}(314)), \\ 0 &= -\ln(\bar{a}(017)) + \ln(\bar{a}(044)) + \ln(\bar{a}(107)) - \ln(\bar{a}(134)) - \ln(\bar{a}(404)) + \ln(\bar{a}(413)), \\ 0 &= -\ln(\bar{a}(017)) + \ln(\bar{a}(035)) + \ln(\bar{a}(107)) - \ln(\bar{a}(125)) - \ln(\bar{a}(503)) + \ln(\bar{a}(512)), \\ 0 &= -\ln(\bar{a}(017)) + \ln(\bar{a}(035)) + \ln(\bar{a}(107)) + \ln(\bar{a}(116)) - \ln(\bar{a}(125)) - \ln(\bar{a}(134)) - \ln(\bar{a}(206)) + \ln(\bar{a}(224)), \\ 0 &= -\ln(\bar{a}(017)) + \ln(\bar{a}(044)) + \ln(\bar{a}(107)) + \ln(\bar{a}(116)) - \ln(\bar{a}(134)) - \ln(\bar{a}(134)) - \ln(\bar{a}(206)) + \ln(\bar{a}(233)), \\ 0 &= -\ln(\bar{a}(017)) - \ln(\bar{a}(026)) + \ln(\bar{a}(035)) + \ln(\bar{a}(044)) + \ln(\bar{a}(107)) + \ln(\bar{a}(116)) - \ln(\bar{a}(134)) - \ln(\bar{a}(134)) - \ln(\bar{a}(305)) + \ln(\bar{a}(323)), \\ 0 &= -\ln(\bar{a}(017)) - \ln(\bar{a}(026)) + \ln(\bar{a}(044)) + \ln(\bar{a}(035)) + \ln(\bar{a}(107)) + \ln(\bar{a}(116)) - \ln(\bar{a}(134)) - \ln(\bar{a}(134)) - \ln(\bar{a}(305)) + \ln(\bar{a}(323)), \\ 0 &= -\ln(\bar{a}(017)) - \ln(\bar{a}(026)) + \ln(\bar{a}(044)) + \ln(\bar{a}(035)) + \ln(\bar{a}(107)) + \ln(\bar{a}(116)) - \ln(\bar{a}(134)) - \ln(\bar{a}(125)) - \ln(\bar{a}(404)) + \ln(\bar{a}(422)). \end{aligned}$$

Note that, as the value of any $\bar{a} \in [a]$ is fixed from the values of only 21 variables, and as $\forall (i, j, k) \in S_8, a(ijk)N \in \llbracket 1, N \rrbracket$, we have $|[a]| \leq N^{21}$. For any a satisfying these 10 equations, we have $g(a) = \max_{\bar{a} \in [a]} g(\bar{a})$ and thus, $\mathcal{N}_X = \mathcal{O}(N^{21} \mathcal{N}_X^*)$, $\mathcal{N}_Y = \mathcal{O}(N^{21} \mathcal{N}_Y^*)$, $\mathcal{N}_Z = \mathcal{O}(N^{21} \mathcal{N}_Z^*)$.

Final statement. Let us also impose the condition

$$(C3) \quad \prod_{i=0}^8 A(i)^{A(i)} \geq \prod_{j=0}^8 B(j)^{B(j)}.$$

By the symmetry condition (C1), this implies that $\prod_{i=0}^8 A(i)^{A(i)} \geq \prod_{k=0}^8 C(k)^{C(k)}$. We get $\mathcal{N}_Y^* = \mathcal{N}_Z^* = \mathcal{O}(\mathcal{N}_X^*)$, and thus we obtain $(\mathcal{N}_X + \mathcal{N}_Y + \mathcal{N}_Z) = \mathcal{O}(N^{21} \mathcal{N}_X^*)$ and

$$\frac{\mathcal{T}_X \mathcal{N}_X^*}{(\mathcal{N}_X + \mathcal{N}_Y + \mathcal{N}_Z)^{1+\epsilon}} = \Omega \left(\frac{\mathcal{T}_X}{(N^{21(1+\epsilon)} (\mathcal{N}_X^*)^\epsilon)} \right).$$

³A Maple file deriving the symbolic solution of this system is available at [36].

As by definition $\mathcal{N}_X^* \leq |S_8|^N = 45^N$, this is equal to

$$r_1 = \Omega \left(\frac{1}{N^{25+21\epsilon} 45^{N\epsilon}} \left[\frac{1}{\prod_{i=0}^8 A(i)^{A(i)}} \right]^N \right).$$

We obtain the following final result.

THEOREM 3.1. *Let q be any positive integer. Let a be any function from $\mathcal{P}(S_8)$ satisfying the constraints (C1), (C2) and (C3). Then for any $\epsilon > 0$, the trilinear form $F^{\otimes N}$ admits a restriction which is a direct sum of r_1 trilinear forms, each of which is isomorphic to*

$$\bigotimes_{(i,j,k) \in S_8} T_{ijk}^{\otimes a(ijk)N}.$$

4 Second extraction

As we will see later in details in Section 6, the tensors T_{ijk} for $(i, j, k) \in S_8$ with one or more of their indices i, j, k equal to 0 are actually matrix products tensors. No further work is required for them. In contrast, the tensors T_{ijk} for $(i, j, k) \in \bar{S}_8$ where $\bar{S}_8 = \{(i, j, k) \in S_8 \mid i > 0, j > 0, k > 0\}$ do not correspond to matrix products.

We are now going to realize an extraction on all the tensors $T_{ijk}, (i, j, k) \in \bar{S}_8$. We first study the properties of the $T_{ijk}, (i, j, k) \in \bar{S}_8$. In Subsection 4.1, we consider the particular case of the tensors T_{233}, T_{323} and T_{332} . In Subsection 4.2, we consider the remaining tensors, i.e., the tensors T_{ijk} for $(i, j, k) \in S_8' = \bar{S}_8 \setminus \{(2, 3, 3), (3, 2, 3), (3, 3, 2)\}$, which are actually easier to analyze. Then, in Subsection 4.3, we explain the limitations of independent extractions and introduce our method to realize a joint extraction.

4.1 The tensors $T_{233}, T_{323}, T_{332}$. The extractions from the tensors $T_{233}, T_{323}, T_{332}$ can be realized similarly to the extraction from the tensor F that we realized in Section 3. As the situation is similar for the three tensors, we only detail the extraction from the tensor T_{233} .

We start from the decomposition

$$T_{233} = \sum_{(u,v,w) \in \bar{S}_{233}} V_{233}[uvw].$$

By definition of the $(T_{ijk})_{(i,j,k) \in S_4}$, for any $(i, j, k), (i', j', k') \in \bar{S}_{233}$ such that $i \neq i'$, the x variables in T_{ijk} and in $T_{i'j'k'}$ are disjoint, and thus the x variables in $V_{233}[ijk]$ and in $V_{233}[i'j'k']$ are disjoint. We can thus realize an extraction from $T_{233}^{a(233)N}$ just as in Section 3. We consider

a mapping $a_{233} \in \mathcal{P}(\bar{S}_{233})$, with projections $A_{233}, B_{233}, C_{233}$. As the $(a_{233}(ijk)a(233))$ are rational numbers, and as N will later go to infinity, we can assume that $\forall (i, j, k) \in \bar{S}_{233}, a_{233}(ijk)a(233)N \in \mathbb{N}$. We impose the symmetry condition $\forall (i, j, k) \in \bar{S}_{233}, a_{233}(ijk) = a_{233}(ikj)$, and thus $B_{233} = C_{233}$. We also impose the symmetry condition

$$\forall (i, j, k) \in \bar{S}_{233}, a_{233}(ijk) = a_{233}(2-i, 3-j, 3-k),$$

as $V_{233}[ijk] = V_{233}[2-i, 3-j, 3-k]$. By realizing an extraction successively on the x variables, the y variables and the z variables, we are left with the tensors $V_{233}[IJK]$ where $(I, J, K) \in \bar{S}_{233}^{a(233)N}$ is of type $\bar{a}_{233} \in [a_{233}]$. Here $[a_{233}]$ denotes the set of mappings $\bar{a}_{233} : \bar{S}_{233} \rightarrow]0, 1[$ which have the same projections as a_{233} and satisfy $\forall (i, j, k) \in \bar{S}_{233}, \bar{a}_{233}(ijk)a(233)N \in \mathbb{N}$.

The number of sequences $I \in \llbracket 0, 2 \rrbracket^{a(233)N}$ of type A_{233} is

$$\begin{aligned} \mathcal{T}_{233,X} &= \left(\frac{a(233)N}{(A_{233}(i)a(233)N)_{i \in \llbracket 0, 2 \rrbracket}} \right) \\ &= \Theta \left(\frac{1}{N} \left(\frac{1}{\prod_{i=0}^2 A_{233}(i)^{A_{233}(i)}} \right)^{a(233)N} \right). \end{aligned}$$

Define the function g_{233} which associates to any mapping $x : \bar{S}_{233} \rightarrow]0, 1[$ $(i, j, k) \mapsto x(ijk)$ the value

$$g_{233}(x) = \left(\prod_{(i,j,k) \in \bar{S}_{233}} x(ijk)^{x(ijk)} \right)^{-1}.$$

For any fixed sequence I of type A_{233} , the number of remaining forms $V_{233}[IJK]$ with (I, J, K) of type a_{233} is

$$\begin{aligned} \mathcal{N}_{233,X}^* &= \prod_{i=0}^2 \left(\frac{A_{233}(i)a(233)N}{(a_{233}(ijk)a(233)N)_{j,k \in \mathbb{N} \mid (i,j,k) \in \bar{S}_{233}}} \right) \\ &= \Theta \left(\frac{\left[g_{233}(a_{233}) \prod_{i=0}^2 A_{233}(i)^{A_{233}(i)} \right]^{a(233)N}}{(a(233)N)^{7/2}} \right), \end{aligned}$$

while the total number of remaining forms $V_{233}[IJK]$ is

$$\begin{aligned} \mathcal{N}_{233,X} &= \sum_{\bar{a}_{233} \in [a_{233}]} \prod_{i=0}^2 \left(\frac{A_{233}(i)N}{(\bar{a}_{233}(ijk)a(233)N)_{j,k \in \mathbb{N} \mid (i,j,k) \in \bar{S}_{233}}} \right) \\ &= \Theta \left(\sum_{\bar{a}_{233} \in [a_{233}]} \frac{\left[g_{233}(\bar{a}_{233}) \prod_{i=0}^2 A_{233}(i)^{A_{233}(i)} \right]^{a(233)N}}{(a(233)N)^{7/2}} \right). \end{aligned}$$

Similarly, for any fixed sequence J of type B_{233} , the number of remaining forms $V_{233}[IJK]$ with (I, J, K) of type a_{233} is

$$\mathcal{N}_{233,Y}^* = \Theta \left(\frac{\left[g_{233}(a_{233}) \prod_{j=0}^3 B_{233}(j)^{B_{233}(j)} \right]^{a(233)N}}{(a(233)N)^3} \right),$$

the total number of remaining forms $V_{233}[IJK]$ is

$$\mathcal{N}_{233,Y} = \Theta \left(\sum_{\bar{a}_{233} \in [a_{233}]} \frac{\left[g_{233}(\bar{a}_{233}) \prod_{j=0}^3 B_{233}(j)^{B_{233}(j)} \right]^{a(233)N}}{(a(233)N)^3} \right),$$

and for any fixed sequence K of type C_{233} , the number of remaining forms $V_{233}[IJK]$ with (I, J, K) of type a_{233} is $\mathcal{N}_{233,Z} = \mathcal{N}_{233,Y}$ and the total number of remaining forms $V_{233}[IJK]$ is $\mathcal{N}_{233,Z}^* = \mathcal{N}_{233,Y}^*$.

By studying the function g_{233} in a similar way as we studied the function g in Section 3, we get that $g(a_{233}) = \max_{\bar{a}_{233} \in [a_{233}]} g(\bar{a}_{233})$ for a_{233} satisfying the constraint

$$2 \ln(a_{233}(211)) + \ln(a_{233}(130)) - \ln(a_{233}(202)) - \ln(a_{233}(220)) - \ln(a_{233}(112)) = 0.$$

Calculations show that $\bar{a}_{233} \in [a_{233}]$ can be written as a function of $\bar{a}_{233}(130)$ and $\bar{a}_{233}(103)$ only, and as $\bar{a}_{233}(130), \bar{a}_{233}(103) \in \llbracket 1, a(233)N \rrbracket$, we have that $|\llbracket a_{233} \rrbracket| \leq (a(233)N)^2$ and thus we obtain $\mathcal{N}_{233,X} = \mathcal{O}((a(233)N)^2 \mathcal{N}_{233,X}^*)$, $\mathcal{N}_{233,Y} = \mathcal{O}((a(233)N)^2 \mathcal{N}_{233,Y}^*)$ and $\mathcal{N}_{233,Z} = \mathcal{O}((a(233)N)^2 \mathcal{N}_{233,Z}^*)$.

The tensors T_{323} and T_{332} are analysed similarly. Imposing the constraints

$$\begin{aligned} 2 \ln(a_{323}(121)) + \ln(a_{323}(310)) - \ln(a_{323}(022)) - \ln(a_{323}(220)) - \ln(a_{323}(112)) &= 0 \\ 2 \ln(a_{332}(112)) + \ln(a_{332}(031)) - \ln(a_{332}(202)) - \ln(a_{332}(022)) - \ln(a_{332}(211)) &= 0 \end{aligned}$$

implies that $\mathcal{N}_{323,X} = \mathcal{O}(N^2 \mathcal{N}_{323,X}^*)$ and $\mathcal{N}_{332,X} = \mathcal{O}(N^2 \mathcal{N}_{332,X}^*)$.

To summarize, when analyzing T_{233} , T_{323} and T_{332} we need to impose the following system (D2) of constraints (in addition to other constraints discussed

later):

$$\begin{aligned} 2 \ln(a_{233}(211)) + \ln(a_{233}(130)) - \ln(a_{233}(202)) - \ln(a_{233}(220)) - \ln(a_{233}(112)) &= 0 \\ 2 \ln(a_{323}(121)) + \ln(a_{323}(310)) - \ln(a_{323}(022)) - \ln(a_{323}(220)) - \ln(a_{323}(112)) &= 0 \\ 2 \ln(a_{332}(112)) + \ln(a_{332}(031)) - \ln(a_{332}(202)) - \ln(a_{332}(022)) - \ln(a_{332}(211)) &= 0. \end{aligned}$$

4.2 The tensors of S'_8 . We adopt the same notations as in the previous subsection. We consider as before mappings $a_{ijk} \in \mathcal{P}(\bar{S}_{ijk})$ for $(i, j, k) \in S'_8$, with projections A_{ijk} , B_{ijk} and C_{ijk} , and assume as before $\forall (u, v, w) \in \bar{S}_{ijk}$, $a_{ijk}(uvw)a(ijk)N \in \mathbb{N}$. For $(i, j, k) \in S'_8$, the number of sequences $I \in \llbracket 0, i \rrbracket^{a(ijk)N}$ of type A_{ijk} is

$$\begin{aligned} \mathcal{T}_{ijk,X} &= \left((A_{ijk}(u)a(ijk)N)_{u \in \llbracket 0, i \rrbracket} \right) \\ &= \Theta \left(\frac{1}{N^{\frac{i}{2}}} \left(\frac{1}{\prod_{u=0}^i A_{ijk}(u)^{A_{ijk}(u)}} \right)^{a(ijk)N} \right). \end{aligned}$$

The extractions on the elements of S'_8 are simpler in the sense that we have $\forall (i, j, k) \in S'_8$, $[a_{ijk}] = \{a_{ijk}\}$, i.e., fixing the projections A_{ijk} , B_{ijk} and C_{ijk} fixes all the values of the mapping a_{ijk} . We thus have

$$\mathcal{N}_{ijk,X} = \mathcal{N}_{ijk,X}^*, \quad \mathcal{N}_{ijk,Y} = \mathcal{N}_{ijk,Y}^*, \quad \mathcal{N}_{ijk,Z} = \mathcal{N}_{ijk,Z}^*,$$

with

$$\begin{aligned} \mathcal{N}_{ijk,X} &= \prod_{x=0}^i \left((A_{ijk}(x)a(ijk)N)_{y,z \in \mathbb{N} \mid (x,y,z) \in \bar{S}_{ijk}} \right) \\ &= \Theta \left(\frac{\left[g_{ijk}(a_{ijk}) \prod_{x=0}^i A_{ijk}(x)^{A_{ijk}(x)} \right]^{a(ijk)N}}{(a(ijk)N) \frac{|\bar{S}_{ijk}| - (i+1)}{2}} \right), \end{aligned}$$

$$\begin{aligned} \mathcal{N}_{ijk,Y} &= \prod_{y=0}^j \left((A_{ijk}(y)a(ijk)N)_{x,z \in \mathbb{N} \mid (x,y,z) \in \bar{S}_{ijk}} \right) \\ &= \Theta \left(\frac{\left[g_{ijk}(a_{ijk}) \prod_{y=0}^j B_{ijk}(y)^{B_{ijk}(y)} \right]^{a(ijk)N}}{(a(ijk)N) \frac{|\bar{S}_{ijk}| - (j+1)}{2}} \right), \end{aligned}$$

$$\mathcal{N}_{ijk,Z} = \prod_{z=0}^k \left(\frac{A_{ijk}(z)a(ijk)N}{(a_{ijk}(xyz)a(ijk)N)_{x,y \in \mathbb{N}^{|(x,y,z) \in \bar{S}_{ijk}}}} \right) \Theta \left(\frac{\left[g_{ijk}(a_{ijk}) \prod_{z=0}^k C_{ijk}(z)C_{ijk}(z) \right]^{a(ijk)N}}{(\overline{S}_{ijk}) - (k+1)} \right) \frac{1}{2} \frac{1}{(a(ijk)N)}$$

4.3 The joint extraction. If we were to realize the extraction on each of the $T_{ijk}, (i, j, k) \in \bar{S}_8$ independently, we would have to impose either the constraints

$$\forall (i, j, k) \in \bar{S}_8, \prod_{x=0}^i A_{ijk}(x)^{A_{ijk}(x)} \geq \prod_{y=0}^j B_{ijk}(y)^{B_{ijk}(y)}$$

$$\text{and } \prod_{x=0}^i A_{ijk}(x)^{A_{ijk}(x)} \geq \prod_{z=0}^k C_{ijk}(z)^{C_{ijk}(z)}$$

or, as in [17], introduce the notion of \mathcal{C} -tensor to perform the analysis. It does not seem, however, that any of these two approaches is helpful for analyzing the fourth power of the Coppersmith-Winograd tensor in the rectangular setting (in particular, the above constraints are too strong and do not lead to any improved lower bound on α). Instead, we are going to realize a global extraction directly on the tensor

$$(4.3) \quad T = \bigotimes_{(i,j,k) \in \bar{S}_8} T_{ijk}^{\otimes a(ijk)N}$$

Note that this tensor can be decomposed as follows:

$$T = \bigotimes_{(i,j,k) \in \bar{S}_8} \left(\sum_{(u,v,w) \in \bar{S}_{ijk}} V_{ijk}[uvw] \right)^{\otimes a(ijk)N}$$

$$= \bigotimes_{(i,j,k) \in \bar{S}_8} \left(\sum_{(I,J,K) \in \bar{S}_{ijk}^{a(ijk)N}} V_{ijk}[IJK] \right)$$

where for $(I, J, K) \in \bar{S}_{ijk}^{a(ijk)N}$,

$$V_{ijk}[IJK] = \bigotimes_{l \in [1, a(ijk)N]} V_{ijk}[I(l)J(l)K(l)].$$

$$T = \sum_{(I,J,K) \in \prod_{(i,j,k) \in \bar{S}_8} \bar{S}_{ijk}^{a(ijk)N}} \left(\bigotimes_{(i,j,k) \in \bar{S}_8} V_{ijk}[I_{ijk}J_{ijk}K_{ijk}] \right)$$

where we see $(I, J, K) \in \prod_{(i,j,k) \in \bar{S}_8} \bar{S}_{ijk}^{a(ijk)N}$ as a family

$(I_{ijk}, J_{ijk}, K_{ijk})_{(i,j,k)}$ indexed by $(i, j, k) \in \bar{S}_8$ with $(I_{ijk}, J_{ijk}, K_{ijk}) \in \bar{S}_{ijk}^{a(ijk)N}$.

Let $(i, j, k) \in \bar{S}_8$. By definition of the $(T_{uvw})_{(u,v,w) \in S_4}$, for any $(u, v, w), (u', v', w') \in \bar{S}_{ijk}$ such that $u \neq u'$, the x variables in T_{uvw} and in $T_{u'v'w'}$ are disjoint, and thus the x variables in $V_{ijk}[uvw]$ and in $V_{ijk}[u'v'w']$ are disjoint. This implies that for any $(I_{ijk}, J_{ijk}, K_{ijk}), (I'_{ijk}, J'_{ijk}, K'_{ijk}) \in \bar{S}_{ijk}^{a(ijk)N}$ with $I_{ijk} \neq I'_{ijk}$, the x variables in $V_{ijk}[I_{ijk}J_{ijk}K_{ijk}]$ and in $V_{ijk}[I'_{ijk}J'_{ijk}K'_{ijk}]$ are disjoint.

The x variables of the tensor (4.3) are indexed by a sequence of indices. This sequence can be divided in a partition of subsequences, each subsequence being associated to a $V_{ijk}[I_{ijk}, J_{ijk}, K_{ijk}]$ for a $(i, j, k) \in \bar{S}_8$. Hence, for any $(I, J, K), (I', J', K') \in \prod_{(i,j,k) \in \bar{S}_8} \bar{S}_{ijk}^{a(ijk)N}$ with $I \neq I'$, the x variables in

$$V[IJK] = \bigotimes_{(i,j,k) \in \bar{S}_8} V_{ijk}[I_{ijk}J_{ijk}K_{ijk}]$$

and

$$V[I'J'K'] = \bigotimes_{(i,j,k) \in \bar{S}_8} V_{ijk}[I'_{ijk}J'_{ijk}K'_{ijk}]$$

are disjoint.

We rewrite the tensor (4.3) as

$$\bigotimes_{(i,j,k) \in \bar{S}_8} T_{ijk}^{\otimes a(ijk)N} = \sum_{(I,J,K) \in \prod_{(i,j,k) \in \bar{S}_8} \bar{S}_{ijk}^{a(ijk)N}} V[IJK].$$

Define $\tilde{a} = (a_{ijk})_{(i,j,k) \in \bar{S}_8}$ and the three projections $\tilde{A} = (A_{ijk})_{(i,j,k) \in \bar{S}_8}$, $\tilde{B} = (B_{ijk})_{(i,j,k) \in \bar{S}_8}$, $\tilde{C} = (C_{ijk})_{(i,j,k) \in \bar{S}_8}$. We extend the definition of type (Definition 2.1) to a product of types: we say, for $(I, J, K) \in \prod_{(i,j,k) \in \bar{S}_8} \bar{S}_{ijk}^{a(ijk)N}$, that I is of *multi-type* \tilde{A} if for every $(i, j, k) \in \bar{S}_8$, I_{ijk} is of type A_{ijk} , and that (I, J, K) is of multi-type \tilde{a} if for every $(i, j, k) \in \bar{S}_8$ $(I_{ijk}, J_{ijk}, K_{ijk})$ is of type a_{ijk} .

We set to 0 all the x variables except those that appear in a $V[IJK]$ where I is of multi-type \tilde{A} . We apply the same procedure for the y variables with the multi-type \tilde{B} and for the z variables with the multi-type \tilde{C} . We are thus left with only the tensors $V[IJK]$ with I of multi-type \tilde{A} , J of multi-type \tilde{B} , K of multi-type \tilde{C} .

By definition of a multi-type, the number of I of multi-type \tilde{A} is

$$\tilde{\mathcal{T}}_X = \prod_{(i,j,k) \in \bar{S}_8} \mathcal{T}_{ijk,X}$$

For any fixed I of multi-type \tilde{A} , the number of remaining

tensors $V[IJK]$ with (I, J, K) of multi-type \tilde{a} is

$$\tilde{\mathcal{N}}_X^* = \prod_{(i,j,k) \in \bar{S}_8} \mathcal{N}_{ijk,X}^*.$$

For any fixed I of multi-type \tilde{A} , the number of remaining tensors $V[IJK]$ is

$$\tilde{\mathcal{N}}_X = \prod_{(i,j,k) \in \bar{S}_8} \mathcal{N}_{ijk,X}.$$

For any fixed J of multi-type \tilde{B} , the number of remaining tensors $V[IJK]$ is

$$\tilde{\mathcal{N}}_Y = \prod_{(i,j,k) \in \bar{S}_8} \mathcal{N}_{ijk,Y}.$$

For any fixed K of multi-type \tilde{C} , the number of remaining tensors $V[IJK]$ is

$$\tilde{\mathcal{N}}_Z = \prod_{(i,j,k) \in \bar{S}_8} \mathcal{N}_{ijk,Z}.$$

We now show that we are under the conditions of Theorem 2.2. We set

$$\Lambda' = \left\{ (I, J, K) \in \prod_{(i,j,k) \in \bar{S}_8} S_4^{a(ijk)N} \right\}.$$

Note that as $\forall (i, j, k) \in \bar{S}_8, \bar{S}_{ijk}^{a(ijk)N} \subseteq S_4^{a(ijk)N}$, for any remaining tensor $V[IJK]$, we have that $(I, J, K) \in \Lambda'$. Let $\gamma = \sum_{(i,j,k) \in \bar{S}_8} a(ijk)$. By defining an arbitrary ordering $\theta : (i, j, k, l) \mapsto \llbracket 1, \gamma N \rrbracket$ for $(i, j, k) \in \bar{S}_8$ and $l \in \llbracket 1, a(ijk)N \rrbracket$, we have $\Lambda' \cong \Lambda$ where

$$\Lambda = \left\{ (I, J, K) \in \llbracket 0, 4 \rrbracket^{\gamma N} \times \llbracket 0, 4 \rrbracket^{\gamma N} \times \llbracket 0, 4 \rrbracket^{\gamma N} \mid \forall d \in \llbracket 1, \gamma N \rrbracket, I[\theta^{-1}(d)] + J[\theta^{-1}(d)] + K[\theta^{-1}(d)] = 4 \right\}$$

where we define the notations $I[(i, j, k, l)] = I_{ijk}(l)$, $J[(i, j, k, l)] = J_{ijk}(l)$, $K[(i, j, k, l)] = K_{ijk}(l)$.

We set U to be set of (I, J, K) of the remaining $V[IJK]$. We set $\mathcal{N}_1 = \tilde{\mathcal{N}}_X$, $\mathcal{N}_2 = \tilde{\mathcal{N}}_Y$, $\mathcal{N}_3 = \tilde{\mathcal{N}}_Z$. We set U^* to be the set of (I, J, K) of the remaining $V[IJK]$ with (I, J, K) of multi-type \tilde{a} . We set $\mathcal{N}_1^* = \tilde{\mathcal{N}}_X^*$, $\mathcal{N}_2^* = \tilde{\mathcal{N}}_Y^*$, $\mathcal{N}_3^* = \tilde{\mathcal{N}}_Z^*$. We set $\mathcal{T}_1 = \tilde{\mathcal{T}}_X$, $\mathcal{T}_2 = \tilde{\mathcal{T}}_Y$, $\mathcal{T}_3 = \tilde{\mathcal{T}}_Z$, and all the conditions of Theorem 2.2 are satisfied. Fix $\epsilon > 0$. We obtain, from

$\bigotimes_{(i,j,k) \in \bar{S}_8} T_{ijk}^{\otimes a(ijk)N}$, a direct sum of

$$\Omega \left(\frac{\tilde{\mathcal{T}}_X \tilde{\mathcal{N}}_X^*}{(\tilde{\mathcal{N}}_X + \tilde{\mathcal{N}}_Y + \tilde{\mathcal{N}}_Z)^{1+\epsilon}} \right)$$

trilinear forms, each of which being a $V[IJK]$ with (I, J, K) of multi-type \tilde{a} , i.e., isomorphic to

$$\bigotimes_{(u,v,w) \in \bar{S}_8} \left(\bigotimes_{(i,j,k) \in \bar{S}_{uvw}} V_{uvw}(ijk)^{\otimes a_{uvw}(ijk)a(uvw)N} \right).$$

We now impose the constraint

$$(D3) \quad \prod_{(i,j,k) \in \bar{S}_8} \left(\prod_{x=0}^i A_{ijk}(x)^{A_{ijk}(x)} \right)^{a(ijk)} \geq \prod_{(i,j,k) \in \bar{S}_8} \left(\prod_{y=0}^j B_{ijk}(y)^{B_{ijk}(y)} \right)^{a(ijk)}.$$

This implies that

$$\prod_{(i,j,k) \in \bar{S}_8} \mathcal{N}_{ijk,Y} = \mathcal{O} \left(\prod_{(i,j,k) \in \bar{S}_8} \mathcal{N}_{ijk,X} \right)$$

i.e., $\tilde{\mathcal{N}}_Y = \mathcal{O}(\tilde{\mathcal{N}}_X)$.

We impose the symmetry condition

$$(D1) \quad \forall (i, j, k) \in \bar{S}_8, \forall (u, v, w) \in \bar{S}_{ijk}, a_{ijk}(uvw) = a_{ijk}(i - u, j - v, k - w) = a_{ikj}(u, w, v).$$

The symmetry conditions (C1) and (D1) give $\tilde{\mathcal{N}}_Y = \tilde{\mathcal{N}}_Z$ and $\tilde{\mathcal{N}}_Z = \mathcal{O}(\tilde{\mathcal{N}}_X)$. The number of terms in the direct sum that we obtain is

$$\Omega \left(\frac{\tilde{\mathcal{T}}_X \tilde{\mathcal{N}}_X^*}{\tilde{\mathcal{N}}_X^{1+\epsilon}} \right).$$

We have seen previously in Subsection 4.2 that $\forall (i, j, k) \in S'_8, \mathcal{N}_{ijk,X} = \mathcal{N}_{ijk,X}^*$. Let us rewrite $\tilde{\mathcal{N}}_X$ as

$$\left(\prod_{(i,j,k) \in S'_8} \mathcal{N}_{ijk,X} \right) \left(\prod_{(i,j,k) \in \{(2,3,3), (3,2,3), (3,3,2)\}} \mathcal{N}_{ijk,X} \right)$$

and $\tilde{\mathcal{N}}_X^*$ as

$$\left(\prod_{(i,j,k) \in S'_8} \mathcal{N}_{ijk,X}^* \right) \left(\prod_{(i,j,k) \in \{(2,3,3), (3,2,3), (3,3,2)\}} \mathcal{N}_{ijk,X}^* \right).$$

We have seen in Section 4.1 that by imposing the constraints (D2) we have $\mathcal{N}_{233,X} = \mathcal{O}(N^2 \mathcal{N}_{233,X}^*)$, $\mathcal{N}_{323,X} = \mathcal{O}(N^2 \mathcal{N}_{323,X}^*)$ and $\mathcal{N}_{332,X} = \mathcal{O}(N^2 \mathcal{N}_{332,X}^*)$. Therefore we have that

$$\prod_{(i,j,k) \in \{(2,3,3), (3,2,3), (3,3,2)\}} \mathcal{N}_{ijk,X}$$

is equal to

$$\mathcal{O} \left(N^6 \prod_{(i,j,k) \in \{(2,3,3), (3,2,3), (3,3,2)\}} \mathcal{N}_{ijk,X}^* \right),$$

and thus $\widetilde{\mathcal{N}}_X = \mathcal{O}(N^6 \widetilde{\mathcal{N}}_X^*)$ and the number of terms in the direct sum that we obtain is

$$\Omega \left(\frac{\widetilde{\mathcal{T}}_X}{N^{6(1+\epsilon)} \widetilde{\mathcal{N}}_X^{*\epsilon}} \right).$$

For any $(i, j, k) \in \overline{S}_8$, by definition we have

$$\begin{aligned} \mathcal{N}_{ijk,X}^* &\leq |\overline{S}_{ijk}|^{a(ijk)N} \\ &\leq |S_4|^{a(ijk)N} = \left(\sum_{l=1}^5 l \right)^{a(ijk)N} = 15^{a(ijk)N} \end{aligned}$$

and thus $\widetilde{\mathcal{N}}_X^* \leq 15^N$, and the number of terms in the direct sum is

$$\Omega \left(\frac{\widetilde{\mathcal{T}}_X}{15^{N\epsilon} N^{6(1+\epsilon)}} \right).$$

Replacing $\widetilde{\mathcal{T}}_X$ by its expression, this is equal to

$$r_2 = \Omega \left(\frac{1}{15^{N\epsilon} N^{6\epsilon+6+\sum_{(i,j,k) \in \overline{S}_8} \frac{1}{2}}} \times \left[\prod_{(i,j,k) \in \overline{S}_8} \left(\frac{1}{\prod_{u=0}^i A_{ijk}(u)^{A_{ijk}(u)}} \right)^{a(ijk)N} \right]^N \right).$$

We summarize the result of this section in the following theorem.

THEOREM 4.1. *Let q be any positive integer. Let a be any function from $\mathcal{P}(S_8)$ satisfying the symmetry constraint (C1). For all $(i, j, k) \in \overline{S}_8$, let a_{ijk} be functions from $\mathcal{P}(\overline{S}_{ijk})$ satisfying the constraints (D1), (D2) and (D3). Then for any $\epsilon > 0$, the trilinear form*

$\bigotimes_{(i,j,k) \in \overline{S}_8} T_{ijk}^{\otimes a(ijk)N}$ *admits a restriction which is a direct sum of r_2 tensors, all isomorphic to*

$$\bigotimes_{(u,v,w) \in \overline{S}_8} \left(\bigotimes_{(i,j,k) \in \overline{S}_{uvw}} V_{uvw}(ijk)^{\otimes a_{uvw}(ijk)a(uvw)N} \right).$$

5 Third extraction

From the second extraction of Section 4 appeared tensors of the form $V_{uvw}(ijk)$, which are formed from

tensors $T_{ijk}, (i, j, k) \in S_4$. We will see in Section 6 that all the tensors T_{ijk} with $(i, j, k) \in S_4 \setminus \overline{S}_4$, where $\overline{S}_4 = \{(2, 1, 1), (1, 1, 2), (2, 1, 2)\}$, correspond to matrix products. The last extraction, that we now realize, deals with the tensors T_{211}, T_{121} and T_{112} .

5.1 Properties of the tensors T_{211}, T_{121} and T_{112} . We first focus on the tensor T_{211} and analyze it as done in Ref. [17]. This tensor can be written as $T_{211} = t_{011} + t_{101} + t_{110} + t_{200}$, where

$$\begin{aligned} t_{011} &= \sum_{i=1}^q x_{0,q+1} y_{i,0} z_{i,0}, \\ t_{101} &= \sum_{i,k=1}^q x_{i,k} y_{0,k} z_{i,0}, \\ t_{110} &= \sum_{i,k=1}^q x_{i,k} y_{i,0} z_{0,k}, \\ t_{200} &= \sum_{k=1}^q x_{q+1,0} y_{0,k} z_{0,k}. \end{aligned}$$

For $(I, J, K) = (I_l, J_l, K_l)_{l \in [1,n]} \in S_2^n$, define $t_{IJK} = \bigotimes_{l \in [1,n]} t_{I_l J_l K_l}$.

We now describe an extraction from

$$T_{211}^{\otimes 2m} = \sum_{(I,J,K) \in S_2^{2m}} t_{IJK},$$

where m is an integer. Note that for $(i, j, k) \in S_2$, $(i', j', k') \in S_2$, if $i \neq i'$ the x variables in t_{ijk} and $t_{i'j'k'}$ are distinct, if $j \neq j'$ the y variables in t_{ijk} and $t_{i'j'k'}$ are distinct, and if $k \neq k'$ the z variables in t_{ijk} and $t_{i'j'k'}$ are distinct. Thus, for $(I, J, K), (I', J', K') \in S_2^{2m}$, if $I \neq I'$ the x variables in t_{IJK} and $t_{I'J'K'}$ are distinct, if $J \neq J'$ the y variables in t_{IJK} and $t_{I'J'K'}$ are distinct, and if $K \neq K'$ the z variables in t_{IJK} and $t_{I'J'K'}$ are distinct. Fix $b \in]0, 1[\cap \mathbb{Q}$ and assume (m will later go to infinity) that $bm \in \mathbb{N}$. Define $a_{211}(011) = a_{211}(200) = (1-b)/2$, $a_{211}(101) = a_{211}(110) = b/2$, with projections $A_{211}, B_{211}, C_{211}$. Note that by definition of a_{211} , $B_{211} = C_{211}$. We set to 0 all the x variables but the ones which appear in a t_{IJK} with I of type A_{211} , and all the y (resp. z) except the ones which appear in a t_{IJK} with J (resp. K) of type B_{211} .

It was shown in Subsection 6.1 of [17] that this leads to a sum of forms t_{IJK} isomorphic to

$$t_{011}^{\otimes (1-b)m} \otimes t_{101}^{\otimes bm} \otimes t_{110}^{\otimes bm} \otimes t_{200}^{\otimes (1-b)m} \cong \langle q^{2bm}, q^{2bm}, q^{2(1-b)m} \rangle$$

and that, adopting the same notations as in the previous

extractions of Sections 3 and 4, we have

$$\begin{aligned} \mathcal{T}_{211,X} &= \binom{2m}{(1-b)m, (1-b)m, 2mb} \\ &= \Theta \left(\frac{1}{m} \cdot \left[\frac{2}{(2b)^b(1-b)^{1-b}} \right]^{2m} \right) \\ \mathcal{N}_{211,X} = \mathcal{N}_{211,X}^* &= \binom{2mb}{mb} = \Theta \left(\frac{1}{\sqrt{m}} \cdot [2^b]^{2m} \right) \\ \mathcal{T}_{211,Y} = \mathcal{T}_{211,Z} &= \binom{2m}{m} = \Theta \left(\frac{1}{\sqrt{m}} \cdot [2]^{2m} \right) \\ \mathcal{N}_{211,Y} = \mathcal{N}_{211,Z} = \mathcal{N}_{211,Y}^* = \mathcal{N}_{211,Z}^* & \\ &= \binom{m}{m(1-b)} \binom{m}{m(1-b)} \\ &= \Theta \left(\frac{1}{m} \cdot \left[\frac{1}{b^b(1-b)^{1-b}} \right]^{2m} \right). \end{aligned}$$

The forms T_{112} and T_{121} can be analyzed in the same way as T_{211} by permuting the roles of the x variables, the y variables and the z variables.

5.2 Joint extraction of the tensors T_{211} , T_{121} and T_{112} . From the extraction of Sections 3 and 4, we obtain tensors isomorphic to

$$T_{112}^{\otimes \alpha_{112}N} \otimes T_{121}^{\otimes \alpha_{121}N} \otimes T_{211}^{\otimes \alpha_{211}N}$$

where for $(i, j, k) \in \{(2, 1, 1), (1, 1, 2), (2, 1, 2)\}$, α_{ijk} is defined as

$$\begin{aligned} &\sum_{(u,v,w) \in \bar{S}_8, \exists (i',j',k') \in S_4, ((i,j,k), (i',j',k')) \in S_{uvw}} a(uvw)a_{uvw}(ijk) + \\ &\sum_{(u,v,w) \in \bar{S}_8, \exists (i',j',k') \in S_4, ((i',j',k'), (i,j,k)) \in S_{uvw}} a(uvw)a_{uvw}(i'j'k'). \end{aligned}$$

Note that the symmetry conditions (C1) and (D1) imply that $\alpha_{112} = \alpha_{121}$.

Using the parameters from Subsection 5.1, we realize a joint extraction on the tensor $T_{112}^{\otimes \alpha_{112}N} \otimes T_{121}^{\otimes \alpha_{112}N} \otimes T_{211}^{\otimes \alpha_{211}N}$, just as in Subsection 4.3. We actually use a constant $b \in]0, 1[$ for defining the types used for T_{112} and T_{121} and another constant \tilde{b} for the type used for T_{211} . We impose the constraint

$$(E3) \quad \frac{(2^{\tilde{b}})^{\alpha_{211}}}{(b^b(1-b)^{1-b})^{\alpha_{112}}} \geq \frac{(2^b)^{\alpha_{112}}}{(\tilde{b}^{\tilde{b}}(1-\tilde{b})^{1-\tilde{b}})^{\alpha_{211}}}$$

which ensures that

$$\begin{aligned} \mathcal{N}_{211,Y} \mathcal{N}_{112,Y} \mathcal{N}_{121,Y} &= \mathcal{N}_{211,Z} \mathcal{N}_{112,Z} \mathcal{N}_{121,Z} \\ &= \mathcal{O}(\mathcal{N}_{211,X} \mathcal{N}_{112,X} \mathcal{N}_{121,X}). \end{aligned}$$

We then get a direct sum of

$$\Omega \left(\frac{\mathcal{T}_{211,X} \mathcal{T}_{112,X} \mathcal{T}_{121,X}}{(\mathcal{N}_{211,X} \mathcal{N}_{112,X} \mathcal{N}_{121,X})^\epsilon} \right)$$

trilinear forms, all isomorphic to

$$\widehat{T}_{211} = \langle q^{(\alpha_{112} + \alpha_{211}\tilde{b})N}, q^{(\alpha_{112} + \alpha_{211}\tilde{b})N}, q^{(2\alpha_{112}b + \alpha_{211}(1-\tilde{b}))N} \rangle.$$

Replacing the \mathcal{T} by their values, and using the bounds $\mathcal{N}_{211,X} \leq 4^{\alpha_{211}N}$, $\mathcal{N}_{112,X} = \mathcal{N}_{121,X} \leq 4^{\alpha_{112}N}$, we get that the number of terms in the direct sum is

$$r_3 = \Omega \left(\frac{1}{4^{(\alpha_{211} + 2\alpha_{112})N\epsilon} N^{2\epsilon}} \left[\frac{2^{2\alpha_{112} + \alpha_{211}}}{((2\tilde{b})^{\tilde{b}}(1-\tilde{b})^{1-\tilde{b}})^{\alpha_{211}}} \right]^{N\epsilon} \right).$$

We summarize the result of this last extraction in the following theorem.

THEOREM 5.1. *For any positive $\alpha_{211}, \alpha_{112}$, for any $b, \tilde{b} \in]0, 1[\cap \mathbb{Q}$ satisfying (E3) and for any $\epsilon > 0$, the trilinear form*

$$T_{112}^{\otimes \alpha_{112}N} \otimes T_{121}^{\otimes \alpha_{112}N} \otimes T_{211}^{\otimes \alpha_{211}N}$$

admits a restriction which is a direct sum of r_3 trilinear forms, all isomorphic to \widehat{T}_{211} .

6 The full extraction

Let us first extend the definition of the values α_{ijk} , which were introduced only for $(i, j, k) \in \bar{S}_4$ in Section 5.2, to all triples in S_4 : for any $(i, j, k) \in S_4$ define α_{ijk} as

$$\begin{aligned} &\sum_{(u,v,w) \in \bar{S}_8, \exists (i',j',k') \in S_4, ((i,j,k), (i',j',k')) \in S_{uvw}} a(uvw)a_{uvw}(ijk)N + \\ &\sum_{(u,v,w) \in \bar{S}_8, \exists (i',j',k') \in S_4, ((i',j',k'), (i,j,k)) \in S_{uvw}} a(uvw)a_{uvw}(i'j'k')N. \end{aligned}$$

From the three consecutive extractions described in Sections 3–5, we get a direct sum of $r_1 r_2 r_3$ trilinear forms, each of them being isomorphic to

$$\left(\bigotimes_{(i,j,k) \in S_8 \setminus \bar{S}_8} T_{ijk}^{\alpha(ijk)N} \right) \otimes \left(\bigotimes_{(i,j,k) \in S_4 \setminus \bar{S}_4} T_{ijk}^{\alpha_{ijk}N} \right) \otimes \widehat{T}_{211},$$

denoted $(*)$. For any $(i, j, k) \in S_4 \setminus \bar{S}_4$, the trilinear form T_{ijk} represent a matrix product (cf. [10]):

$$\begin{aligned} T_{004} &\cong T_{040} \cong T_{400} \cong \langle 1, 1, 1 \rangle \\ T_{013} &\cong T_{031} \cong \langle 1, 1, 2q \rangle \\ T_{103} &\cong T_{301} \cong \langle 2q, 1, 1 \rangle \\ T_{130} &\cong T_{310} \cong \langle 1, 2q, 1 \rangle \\ T_{022} &\cong \langle 1, 1, q^2 + 2 \rangle \\ T_{202} &\cong \langle q^2 + 2, 1, 1 \rangle \\ T_{220} &\cong \langle 1, q^2 + 2, 1 \rangle. \end{aligned}$$

It can be seen from the definitions of the trilinear form $T_{ijk}, (i, j, k) \in S_8 \setminus \bar{S}_8$ that they also represent a matrix product. We have:

$$\begin{aligned}
 T_{008} &\cong T_{080} \cong T_{800} \cong \langle 1, 1, 1 \rangle \\
 T_{017} &\cong T_{071} \cong \langle 1, 1, 4q \rangle \\
 T_{107} &\cong T_{170} \cong \langle 4q, 1, 1 \rangle \\
 T_{701} &\cong T_{710} \cong \langle 1, 4q, 1 \rangle \\
 T_{026} &\cong T_{062} \cong \langle 1, 1, 6q^2 + 4 \rangle \\
 T_{206} &\cong T_{260} \cong \langle 6q^2 + 4, 1, 1 \rangle \\
 T_{602} &\cong T_{620} \cong \langle 1, 6q^2 + 4, 1 \rangle \\
 T_{035} &\cong T_{053} \cong \langle 1, 1, 4q^3 + 12q \rangle \\
 T_{305} &\cong T_{350} \cong \langle 4q^3 + 12q, 1, 1 \rangle \\
 T_{503} &\cong T_{530} \cong \langle 1, 4q^3 + 12q, 1 \rangle \\
 T_{044} &\cong \langle 1, 1, q^4 + 12q^2 + 6 \rangle \\
 T_{404} &\cong \langle q^4 + 12q^2 + 6, 1, 1 \rangle \\
 T_{440} &\cong \langle 1, q^4 + 12q^2 + 6, 1 \rangle.
 \end{aligned}$$

As we have seen in Section 5, \widehat{T}_{211} is also a matrix product. Hence, the $r_1 r_2 r_3$ isomorphic trilinear forms that we have extracted all represent the same matrix product. By the symmetry constraints (C1) and (D1), we get that this matrix product is of the form $\langle Q^N, Q^N, R^N \rangle$. The expressions of Q and R are obtained by replacing in (*) the T_{ijk} and \widehat{T}_{211} by the matrix products they correspond to. We refrain from giving here the complete expressions for R and Q since the formulas are extremely long (they can be found in the files of the programs used for the numerical analysis [36]). We have shown:

$$r_1 r_2 r_3 \cdot \langle Q^N, Q^N, R^N \rangle \leq F^{\otimes N}.$$

As we already saw in Subsection 2.2, $\underline{R}(F) \leq (q+2)^4$ and thus by submultiplicativity of the border rank $\underline{R}(F^{\otimes N}) \leq (q+2)^{4N}$. By Schönhage's asymptotic sum inequality (Theorem 2.1), we have:

$$r_1 r_2 r_3 Q^{N \omega(\frac{\log R}{\log Q})} \leq (q+2)^{4N},$$

and taking the N -th root, we get:

$$(r_1 r_2 r_3)^{\frac{1}{N}} Q^{\omega(\frac{\log R}{\log Q})} \leq (q+2)^4.$$

Let

$$\begin{aligned}
 \mathcal{M} &= \lim_{\epsilon \rightarrow 0} \lim_{N \rightarrow \infty} (r_1 r_2 r_3)^{\frac{1}{N}} \\
 &= \frac{1}{\prod_{i=0}^8 A(i)^{A(i)}} \prod_{(i,j,k) \in \bar{S}_8} \left(\frac{1}{\prod_{u=0}^i A_{ijk}(u)^{A_{ijk}(u)}} \right)^{a(ijk)} \times \\
 &\quad \frac{2^{2\alpha_{112} + \alpha_{211}}}{((2\tilde{b})^{\tilde{b}}(1-\tilde{b})^{1-\tilde{b}})^{\alpha_{211}}}.
 \end{aligned}$$

We have

$$\mathcal{M} Q^{\omega(\frac{\log R}{\log Q})} \leq (q+2)^4.$$

We summarize the result of the whole process in the following theorem.

THEOREM 6.1. *Let q be any positive integer. Let a be any function from $\mathcal{P}(S_8)$ satisfying the constraints (C1), (C2) and (C3). For all $(u, v, w) \in \bar{S}_8$, let a_{uvw} be functions from $\mathcal{P}(\bar{S}_{uvw})$ satisfying the constraints (D1), (D2) and (D3). Finally, let b and \tilde{b} be any two values from $]0, 1[\cap \mathbb{Q}$ satisfying the constraint (E3). Then the following inequality holds:*

$$\mathcal{M} Q^{\omega(\frac{\log R}{\log Q})} \leq (q+2)^4.$$

Theorem 6.1 enables us to obtain our new upper bounds on $\omega(k)$: for any k , if we find values q, a, a_{uvw} for each $(u, v, w) \in \bar{S}_8, b$ and \tilde{b} satisfying the constraints in Theorem 6.1 such that $\frac{\log R}{\log Q} = k$ and $\mathcal{M} Q^\nu \geq (q+2)^4$ for some ν , then we get, since $\mathcal{M} Q^{\omega(k)} \leq (q+2)^4$, that $\omega(k) \leq \nu$. The bounds given in Table 3 and Theorem 1.1 are obtained by finding the optimal values for q, a, a_{uvw} for each $(u, v, w) \in \bar{S}_8, b$ and \tilde{b} by numerical analysis using Maple. The source code of the Maple programs used for the numerical analysis, which include the complete formulas for the terms R and Q , is available at [36].

References

- [1] Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is Valiant's parser. In *Proceedings of the 56th Symposium on Foundations of Computer Science*, pages 98–117, 2015.
- [2] Noga Alon and Raphael Yuster. Fast algorithms for maximum subset matching and all-pairs shortest paths in graphs with a (not so) small vertex cover. In *Proceedings of the 15th Annual European Symposium on Algorithms*, pages 175–186, 2007.
- [3] Andris Ambainis, Yuval Filmus, and François Le Gall. Fast matrix multiplication: Limitations of the Coppersmith-Winograd method. In *Proceedings of the 47th Symposium on Theory of Computing*, pages 585–593, 2015.
- [4] Rasmus Resen Amossen and Rasmus Pagh. Faster join-projects and sparse matrix multiplications. In *Proceedings of the 12th International Conference on Database Theory*, pages 121–126, 2009.
- [5] Markus Bläser. Fast matrix multiplication. *Theory of Computing, Graduate Surveys*, 5:1–60, 2013.
- [6] Karl Bringmann, Fabrizio Grandoni, Barna Saha, and Virginia Vassilevska Williams. Truly sub-cubic algorithms for language edit distance and RNA-folding via

- fast bounded-difference min-plus product. In *Proceedings of the 57th Symposium on Foundations of Computer Science*, pages 375–384, 2016.
- [7] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. *Algebraic complexity theory*. Springer, 1997.
- [8] Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal on Computing*, 11(3):467–471, 1982.
- [9] Don Coppersmith. Rectangular matrix multiplication revisited. *Journal of Complexity*, 13(1):42–49, 1997.
- [10] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.
- [11] Alexander Munro Davie and Andrew James Stothers. Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh*, 143A:351–370, 2013.
- [12] Camil Demetrescu and Giuseppe F. Italiano. Fully dynamic transitive closure: Breaking through the $o(n^2)$ barrier. In *Proceedings of the 41st Symposium on Foundations of Computer Science*, pages 381–389, 2000.
- [13] Xiaohan Huang and Victor Y. Pan. Fast rectangular matrix multiplication and applications. *Journal of Complexity*, 14(2):257–299, 1998.
- [14] Haim Kaplan, Natan Rubin, Micha Sharir, and Elad Verbin. Counting colors in boxes. In *Proceedings of the 18th Symposium on Discrete Algorithms*, pages 785–794, 2007.
- [15] Haim Kaplan, Micha Sharir, and Elad Verbin. Colored intersection searching via sparse rectangular matrix multiplication. In *Proceedings of the 22nd Symposium on Computational Geometry*, pages 52–60, 2006.
- [16] ShanXue Ke, BenSheng Zeng, WenBao Han, and Victor Y. Pan. Fast rectangular matrix multiplication and some applications. *Science in China Series A: Mathematics*, 51(3):389–406, 2008.
- [17] François Le Gall. Faster algorithms for rectangular matrix multiplication. In *Proceedings of the 53rd Symposium on Foundations of Computer Science*, pages 514–523, 2012.
- [18] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 296–303, 2014.
- [19] Grazia Lotti and Francesco Romani. On the asymptotic complexity of rectangular matrix multiplication. *Theoretical Computer Science*, 23:171–185, 1983.
- [20] Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the 21st Symposium on Discrete Algorithms*, pages 1065–1075, 2010.
- [21] Liam Roditty and Asaf Shapira. All-pairs shortest paths with a sublinear additive error. *ACM Transactions on Algorithms*, 7(4):45, 2011.
- [22] Piotr Sankowski and Marcin Mucha. Fast dynamic transitive closure with lookahead. *Algorithmica*, 56(2):180–197, 2010.
- [23] Arnold Schönhage. Partial and total matrix multiplication. *SIAM Journal on Computing*, 10(3):434–455, 1981.
- [24] Andrew Stothers. *On the Complexity of Matrix Multiplication*. PhD thesis, University of Edinburgh, 2010.
- [25] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.
- [26] Volker Strassen. The asymptotic spectrum of tensors and the exponent of matrix multiplication. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 49–54, 1986.
- [27] Volker Strassen. Relative bilinear complexity and matrix multiplication. *Journal für die reine und angewandte Mathematik*, 375-376:406–443, 1987.
- [28] Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th Symposium on Theory of Computing*, pages 887–898, 2012.
- [29] Virginia Vassilevska Williams, Joshua R. Wang, Richard Ryan Williams, and Huacheng Yu. Finding four-node subgraphs in triangle time. In *Proceedings of the 26th Symposium on Discrete Algorithms*, pages 1671–1680, 2015.
- [30] Ryan Williams. Non-uniform ACC circuit lower bounds. In *Proceedings of the 26th Conference on Computational Complexity*, pages 115–125, 2011.
- [31] Raphael Yuster. Efficient algorithms on sets of permutations, dominance, and real-weighted APSP. In *Proceedings of 20th Symposium on Discrete Algorithms*, pages 950–957, 2009.
- [32] Raphael Yuster and Uri Zwick. Detecting short directed cycles using rectangular matrix multiplication and dynamic programming. In *Proceedings of the 15th Symposium on Discrete Algorithms*, pages 254–260, 2004.
- [33] Raphael Yuster and Uri Zwick. Fast sparse matrix multiplication. *ACM Transactions on Algorithms*, 1(1):2–13, 2005.
- [34] Uri Zwick. All pairs lightest shortest paths. In *Proceedings of the 31st Symposium on Theory of Computing*, pages 61–69, 1999.
- [35] Uri Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication. *Journal of the ACM*, 49(3):289–317, 2002.
- [36] <http://www.francoislegall.com/RMM.zip>.