Graduate School of Natural Sciences

Slice rank and fast matrix multiplication

Nils Van de Berg

Master Thesis

Mathematical Sciences

Supervisor:

Prof. Dr. Dion Gijswijt
Delft University of Technology

Examiners:

Dr. Carla Groenland
Utrecht University

Prof. Dr. Rob Bisseling
Utrecht University

Universiteit Utrecht

16/06/23

1 Acknowledgements

This thesis was made possible by the support of many. Most of all my daily supervisor Prof. Dion Gijswijt, who engaged with me in frequent discussion throughout the year. These discussions have been the root of all new ideas in this thesis. He also guided me through all phases of the writing and I am extremely thankful for that. I want to thank him, Dr. Carla Groenland, Prof. Rob Bisseling but also Ward, Jasper, Sven and Cat for going through earlier versions of this text and helping me eliminate a big portion of the mistakes. In the final months I was also massively supported by everyone else in the 'Wisbieb' in Utrecht who kept my motivation and spirits high whenever I could not do that alone. I want to mention Ludo, Caspar and Gabriëlle explicitly as they were going through the same process and being able to relate my experiences was really helpful.

Abstract

The slice rank is a rank notion for tensors that was introduced in 2016. It can be analysed through the vertex cover number of the support of the tensor. This can give both upper and lower bounds for the slice rank and its asymptotic version. We analyse these bounds for tensors with symmetry. We use this to prove that the asymptotic slice rank is multiplicative for symmetric and oblique tensors. Our motivation for studying the slice rank is the connection with fast matrix multiplication. The problem of fast matrix multiplication asks how many arithmetic operations are needed to multiply two $n \times n$-matrices. The exponent of matrix multiplication ω indicates the asymptotically minimum number of such operations. There are certain methods to find upper bounds for ω and Alman [Alm18] used the slice rank to show that a general class of such methods cannot prove $\omega < 2.16$. To find these explicit limitations one can use upper bounds for the asymptotic slice rank, which are more easily computed for tensors with a specified partition. We add some new results about such upper bounds for partitioned tensors.

The current best upper bound for ω is 2.37187. This bound was obtained using the laser method with base tensor $CW_5^{\otimes 8}$. We describe the laser method and perform some computations for other families of base tensors to conclude that they could potentially be used to obtain better bounds.

Contents

2 Introduction

Matrix multiplication is one of the fundamental operations in linear algebra. The complexity of this operation has thus become a fundamental problem in algebraic complexity theory. The complexity of many other important computations, such as inverting matrices and finding LUP decompositions, is determined by the complexity of matrix multiplication [BCS96]. In this thesis we look at the asymptotic behaviour of matrix multiplication. This problem can be restated in terms of tensors and we study the tensor rank and slice rank as a way to find bounds on the complexity of matrix multiplication. We also focus on the slice rank as an independent notion of interest.

The product of two $n \times n$-matrices A and B is another $n \times n$-matrix AB. The standard or naive algorithm for computing this product determines each entry of AB one at a time. For 2×2-matrices we get

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix},$$

so each entry in AB requires two multiplications. This means we need 8 multiplications in total. For $n \times n$-matrices the naive algorithm requires n^3 multiplications in total. In 1969 Strassen found a groundbreaking method to multiply two 2×2-matrices with only 7 multiplications [Str69]. First, use one multiplication to compute each of $M_1, \ldots, M_7$ where

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22}),\ M_2 = (A_{21} + A_{22})B_{11},\ M_3 = A_{11}(B_{12} - B_{22}), \quad M_4 = A_{22}(B_{21} - B_{11})$$
$$M_5 = (A_{21} - A_{11})(B_{11} + B_{12}),\ M_6 = (A_{11} + A_{12})B_{22},\ M_7 = (A_{12} - A_{22})(B_{21}+B_{22})$$

and then write the product of A and B purely in terms of sums and differences of the M_i

$$AB = \begin{pmatrix} M_1 + M_4 - M_6 + M_7 & M_3 + M_6 \\ M_2 + M_4 & M_1 - M_2 + M_3 + M_5 \end{pmatrix}. \tag{2.1}$$

By using this method recursively one gets an algorithm that computes the product of two $2^k \times 2^k$-matrices with only $O(7^k)$ multiplications.

Algorithm 1 Strassen(k)

Input: Two $2^k \times 2^k$-matrices A,B.
Output: The matrix product AB.
0. If $k = 0$ multiply the numbers A and B and output the product.
Else:
1. Split the columns and rows of matrices A, B in two. This gives $2^{k-1} \times 2^{k-1}$-submatrices $A_{11}, A_{12}, A_{21}, A_{22}$ and B_1, B_2, B_3, B_4.
2. Add the submatrices together in preparation for calculating the seven products M_i in Strassen's identity.
3. Use Strassen($k-1$) seven times to find each of the products M_i.
4. Add these together according to Equation 2.1 to find each of the four $2^{k-1} \times 2^{k-1}$-submatrices of AB.

If m_k is the number of multiplications needed to run Strassen(k), then $m_k = 7m_{k-1}$, so $m_k = 7^k$. This algorithm uses more additions than the naive one, but this number grows slower than 7^k. Therefore this algorithm implies that $n \times n$-matrices can be multiplied with only $O(n^{\log_2 7})$ arithmetic operations (additions and multiplications).
The fact that $\log_2 7 \approx 2.81$ is smaller than 3 means that for large matrices this algorithm needs fewer operations and is thus faster. These algorithms that break the n^3 complexity are known as fast matrix multiplication algorithms. Strassen's algorithm gets used in practice and can be faster, even for 1000×1000-matrices [DN05]. However Strassen's is not necessarily the fastest algorithm. The question is for which values τ is there an algorithm that uses $O(n^\tau)$ arithmetic operations to multiply $n \times n$-matrices. The infimum of all such values τ is called the exponent of matrix multiplication, denoted by ω. Mathematicians have found increasingly complex ways to upper bound ω. At the moment we know that $2 \leq \omega < 2.37187$, but this leads to the main question of the field.

Open problem 2.0.1. *What is the exact value of ω?*

As mentioned before the matrix multiplication problem is related to many other problems in algebraic complexity theory. It has been shown that finding the determinant, inverting matrices, finding LUP decompositions, and determining the transitive closure of a graph also have complexity $O(n^\omega)$ asymptotically [BCS96].

Finding upper bounds for ω quickly moved away from establishing explicit algorithms. Instead, it was observed that a certain algebraic object, the matrix multiplication tensor, contained all the information about matrix multiplication that is needed to analyse its computational complexity. The $\langle 2,2,2\rangle$ matrix multiplication tensor is

$$\langle 2,2,2\rangle := \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} x_{ij}y_{jk}z_{ki}$$

and represents the multiplication of two 2×2-matrices. If one multiplies a matrix with entries x_{ij} with a matrix with entries y_{jk} then the ik entry of the product corresponds to the sum $\sum_j x_{ij}y_{jk}$. These are exactly all terms in the tensor that have z_{ki} in it. Strassen's identity can also be rewritten in terms of this tensor via

$$\begin{aligned}\langle 2,2,2\rangle = (x_{12}-x_{22})(y_{12}+y_{22})z_{11} &+ (x_{21}+x_{22})y_{11}(z_{12}-z_{22}) + x_{11}(y_{12}-y_{11})(z_{21}+z_{22})\\ + x_{22}(y_{21}-y_{11})(z_{11}+z_{12}) &+ (x_{11}+x_{12})y_{22}(z_{21}-z_{11}) + (x_{21}-x_{11})(y_{11}+y_{12})z_{22}\\ + (x_{11}+x_{22})(y_{11}+y_{22})(z_{11}+z_{22}).&\end{aligned} \tag{2.2}$$

This is a sum of only 7 terms, with each term consisting of a product of linear combinations of x-variables, y-variables and z-variables. In general, tensors can intuitively be thought of as higher-dimensional matrices. These can be studied in their own right and have their own version of matrix rank called tensor rank R. In Chapter 2 it will become clear that the expression above shows that $R(\langle 2,2,2\rangle)\leq 7$. It was shown that the tensor rank of matrix multiplication tensors is equal to the minimum number of multiplications in any matrix multiplication algorithm. Asymptotically the multiplications are the dominating term in the number of arithmetic operations, which means that tensor rank determines ω [BCS96]. Establishing the tensor rank of relevant tensors became the new challenge of the field. This is generally an NP-hard problem [HL13], but it can be determined for some families of tensors. The idea was to use these tensors with known rank to upper bound the tensor rank of matrix multiplication tensors, which leads to upper bounds on ω.
This idea was applied by Strassen in the form of the laser method. We will go into the details of this method in Chapter 5. With this method Strassen showed that $\omega < 2.48$ [Str87]. In this method you start with a certain base tensor. This is usually the tensor CW_q, which is the sum of 6 matrix multiplication tensors. Then we take a suitable power of this tensor, $CW_q^{\otimes n}$, and analyse the matrix multiplication tensors this power contains. By comparing this to the rank of $CW_q^{\otimes n}$ we find a bound on ω. Coppersmith and Winograd improved upon his method and could then apply it with base tensor $CW_q^{\otimes 2}$ [CW90]. This gave $\omega < 2.3755$. Later improvements [Wil12][Gal14][AW20] analysed even higher powers of CW_q and made some small improvements to reach $\omega < 2.373$. In 2022 Duan, Wu and Zhou found the current best implementation of the laser method, which established $\omega < 2.37187$ [DWZ22]. The main conjecture in the field is that $\omega = 2$. These methods do not seem to get us close to this value of ω. This phenomenon has also been researched and multiple limitations were found to both the laser method [AFG15] and more general methods, such as the universal method [Alm18].

There are other notions of ranks for tensors. One such rank was defined in 2016. Ellenberg and Gijswijt independently found an exponential improvement on the bound for the capset problem [EG16]. Their proof was analysed and put into tensorial context by Tao [Tao16]. This resulted in the introduction of the slice rank SR. This notion has found use in multiple combinatorial problems, such as for sunflower-free sets [NS17], and can also be applied to the analysis of fast matrix multiplication. In particular, Alman [Alm18] used it to give lower bounds on the exponent ω_u that the universal method can obtain. For these lower bounds we need an asymptotic version of the slice rank $\widetilde{\mathrm{SR}}$. The slice rank is hard to determine for a general tensor [Blä+19], but it can be determined in some cases through analysis of the support. If we think of a tensor as a k-dimensional matrix, then the support is the set of non-zero entries. In a follow-up blogpost [TS16] Tao and Sawin showed multiple connections between the slice rank and the support. In this thesis we improve upon one of their propositions with the following proposition.

Proposition 2.0.2. *Let $k \geq 6$ be an integer and let T be a k-tensor defined on the k-fold product $G \times \cdots \times G$ where G is some finite abelian group. If all elements in the support* $\operatorname{supp}(T)$ *of T are of the form $(a, a+b, \ldots, a+(k-1)b)$ for some $a, b \in G$ and all elements of the form $(g, g, \ldots, g)$ are in* $\operatorname{supp}(T)$*, then $SR(T) = |G|$.*

From their propositions it also follows that the support exactly determines the slice rank if it is of a specific form we call oblique. We will define this term in Chapter 3, but intuitively one can say that obliqueness is a condition for which supports are sparse enough that we cannot find linear combinations of the variables that decrease the slice rank. If the support is oblique, then the slice rank can be thought of as a purely combinatorial object called the vertex cover number. We study this vertex cover number in particular in the case that the support is symmetric. From this we deduce new equalities for the asymptotic slice rank of symmetric tensors.

Theorem 2.0.3. *If S and T are two symmetric and oblique tensors, then $\widetilde{\mathrm{SR}}(S \otimes T) = \widetilde{\mathrm{SR}}(S)\widetilde{\mathrm{SR}}(T)$ and $\widetilde{\mathrm{SR}}(S \oplus T) = \widetilde{\mathrm{SR}}(S) + \widetilde{\mathrm{SR}}(T)$.*

The study of the slice rank and tensor rank can be simplified through partitions. In the laser method each of the sets of variables is given a partition and we treat all variables in one partition class as being the same. The partitions can also be used to obtain upper bounds on the asymptotic slice rank [Alm18]. We show that Alman's upper bound, which is some expression $h_\Lambda(\Phi)$, follows directly from other known bounds by showing that $h_\Lambda(\Phi)$ decreases when changing to finer partitions.

Proposition 2.0.4. *Let $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ be a triple of partitions of X, Y, Z and let $\Lambda' = (\mathcal{X}', \mathcal{Y}', \mathcal{Z}')$ be a refinement of Λ in the sense that $\mathcal{X}'$ is a refinement of $\mathcal{X}$, $\mathcal{Y}'$ is a refinement of $\mathcal{Y}$ and $\mathcal{Z}'$ is a refinement of $\mathcal{Z}$. Consider a subset $\Phi \subseteq X \times Y \times Z$ then $h_\Lambda(\Phi) \geq h_{\Lambda'}(\Phi)$.*

2.1 Overview of this thesis

Chapter 2 formally introduces all the necessary concepts, such as tensors, rank and slice rank to understand the later chapters. This chapter also includes most proofs as it was believed this would help the reader's understanding.

The main body of this thesis starts in Chapter 3, where we explore the combinatorial aspects of the slice rank by analysing the support of tensors and their vertex cover number. This chapter largely includes reinterpretations of known results, but also gives the proofs of Proposition 2.0.2 and Proposition 2.0.4. The equality case in this last proposition leads us to introduce the notion of a regular support, about which we also prove some small statements. In Chapter 4 we relate the results from Chapter 3 to the asymptotic slice rank, which leads to Theorem 2.0.3.

In Chapter 5 we turn our attention to the laser method. The method is described in detail and we prove the general bound it obtains. In addition, we sketch the ideas of more advanced implementations. As a suggestion for a way forward we look at some potential other starting tensors for the laser method in Chapter 6. We have computed lower bounds for the ω that these starting tensors could obtain.

We close this thesis in Chapter 7 by discussing some questions that arose but were left unanswered in this thesis.

3 Preliminaries

3.1 Tensors

There are many representations of tensors. First, we introduce tensors in a basis-independent way, but we shall also adopt basis-dependent notation in order to speak more easily about them in combinatorial terms.

Definition 3.1.1. A *k-tensor* is any element of a tensor product of k finite-dimensional vector spaces with common base field.

A 1-tensor is just a normal vector and a 2-tensor is an element of $V \otimes W$ for two vector finite-dimensional spaces V, W. There are isomorphisms $V \otimes W \cong V \otimes W^* \cong \hom(W, V)$ with W^* the dual space of W. Given bases, a 2-tensor can thus be regarded as a matrix.

For vector spaces $(V_i)_{1\leq i\leq k}$ over $\mathbb{F}$ each with basis $(x^i_j)_{1\leq j\leq n_i}$ the tensor product $V_1\otimes\ldots\otimes V_k$ has basis $\{x^1_{j_1}\otimes\ldots\otimes x^k_{j_k}|(j_1,\ldots,j_k)\in[n_1]\times\ldots\times[n_k]\}$. We can write each tensor T uniquely in terms of this basis with coefficients $t(j_1,\ldots,j_k)\in\mathbb{F}$

$$T=\sum_{(j_1,\ldots,j_k)\in[n_1]\times\ldots\times[n_k]} t(j_1,\ldots,j_k)x^1_{j_1}\otimes\ldots\otimes x^k_{j_k}$$

This way each tensor T defines a coefficient function $t:[n_1]\times\cdots\times[n_k]\to\mathbb{F}$ for any set of bases of the V_i. Such a function also uniquely defines a tensor given a basis, thus we can give a (basis-dependent) representation of tensors as functions $X_1\times\ldots\times X_k\to\mathbb{F}$ where each X_i is a finite set.

Definition 3.1.2. The *support* of a tensor T with respect to a certain basis is the support of the coefficient function t.

To write down tensors concisely we use the notation $\sum_{i_1,\ldots,i_k} t(i_1,i_2,\ldots,i_k)x^1_{i_1}\cdots x^k_{i_k}$ (without the $\otimes$ between the basis elements). In this sum we leave out any terms which are not in the support. We will mostly deal with 3-tensors, because our main tensor of interest, the matrix multiplication tensor, is a 3-tensor. For these we use sets X,Y,Z and write a tensor as $\sum_{ijk} t_{ijk}x_iy_jz_k$. Any time we work with this notation we will have fixed bases in advance, but we do not mention this explicitly. Many results also do not depend on the field over which we work or which vector spaces are chosen. We only mention these explicitly whenever it matters. We can now introduce two types of tensors that are especially important to us.

For any positive integers k and n the *independent k-tensor* $\langle n\rangle_k$ is the tensor

$$\sum_{i=1}^n x^1_ix^2_i\cdots x^k_i.$$

For any positive integers a,b,c the *matrix multiplication tensor* $\langle a,b,c\rangle$ is the tensor

$$\sum_{i=1}^a\sum_{j=1}^b\sum_{k=1}^c x_{ij}y_{jk}z_{ki}.$$

If we multiply an $a\times b$-matrix with entries x_{ij} with a $b\times c$-matrix with entries y_{jk}, then the i,k entry of the product is $\sum_{j=1}^b x_{ij}y_{jk}$. This sum also appears in the tensor as those terms that include z_{ki}. The order of the indices is flipped to uphold symmetry between X,Y,Z variables in the tensor. More precisely, the tensor lives in $\mathbb{F}^{a\times b}\otimes\mathbb{F}^{b\times c}\otimes\mathbb{F}^{a\times c}$ and by using the dual and hom isomorphisms this is isomorphic to $\hom(\mathbb{F}^{a\times b}\otimes\mathbb{F}^{b\times c},\mathbb{F}^{a\times c})$. The matrix multiplication tensor exactly corresponds to matrix multiplication under this isomorphism. This tensor thus encapsulates all the information from matrix multiplication. Following this correspondence, we regard each term xyz in the support as a multiplication. In the way the tensor is written here, there are abc multiplications, just as in the naive algorithm for matrix multiplication. However, Strassen and his fast matrix multiplication algorithm showed that it can be done with fewer multiplications. Equivalently, we should be able to write this tensor with fewer terms. This can indeed be done by choosing different bases. The exact correspondence between multiplications and terms is captured in the tensor rank.

Definition 3.1.3. A *simple tensor* in $V_1\otimes\cdots\otimes V_k$ is a tensor of the form $v_1\otimes\cdots\otimes v_k$ where $v_i\in V_i$ for each i. For any $T\in V_1\otimes\cdots\otimes V_k$ the *tensor rank* $R(T)$ of T is the smallest number of simple tensors needed to sum up to T.

For matrix multiplications we regard each simple tensor as one multiplication and thus the rank $R(\langle a,b,c\rangle)$ is the minimum number of multiplications needed to multiply an $a\times b$-matrix with a $b\times c$-matrix.

Example 3.1.4.

- The zero tensor has rank 0 as the empty sum already adds up to 0.

- Any simple tensor is the sum of 1 simple tensor, so has rank 1.
- For a fixed basis the tensor $\langle n\rangle_k$ has n terms in the support, so $R(\langle n\rangle_k) \leq n$.
- Strassen's identity for 2×2 matrices and equivalently Equation 2.2 show that $R(\langle 2,2,2\rangle) \leq 7$.

From a decomposition of $\langle a,b,c\rangle$ into r simple tensors, as is done in Equation 2.2, one can deduce an algorithm to multiply an $a \times b$-matrix by an $b \times c$-matrix with r multiplications. Therefore the ambitious end goal is to determine $R(\langle a,b,c\rangle)$ and tensor rank decompositions for all a,b,c. The problem is that it is already hard to find good upper bounds on $R(\langle a,b,c\rangle)$. Most of the research is focused on finding increasingly better upper bounds on the tensor rank, but it is also relevant to know lower bounds on R. We record one such lower bound that is relevant for most tensors of interest.

Definition 3.1.5. Let $T \in V_1 \otimes \cdots \otimes V_k$ be a k-tensor with $k \geq 2$. For each i there is an associated map $\phi_i : V_i^\star \to \bigotimes_{j\neq i} V_j$. Then T is *i-concise* if ϕ_i is injective. We say a tensor is concise if it is i-concise for all i.

Given a basis, this is equivalent to requiring the set $\{t_{x_i} := t(-,x_i,-) : x_i \in X_i\}$ of $(k-1)$-tensors to be linearly independent. Often we can see a tensor is concise by looking at the support and seeing that it is linearly independent.

Lemma 3.1.6. *If T is i-concise, then $R(T) \geq \dim V_i$.*

Proof. We only give a short sketch of the proof. Each rank 1 tensor can only define a rank 1 map $\phi_i : V_i^\star \to \bigotimes_{j\neq i} V_j$. As our final map has to be rank $\dim V_i$ we need at least $\dim V_i$ rank 1 tensors to sum to T. □

Example 3.1.7.

- The independent tensor $\langle n\rangle_k$ considered as an element of $\bigotimes_{i=1}^k \mathbb{F}^n$ is 1-concise if $k \geq 2$. Thus $R(\langle n\rangle_k) \geq n$ for all $k \geq 2$.
- Any matrix multiplication tensor $\langle a,b,c\rangle$ considered as an element $\mathbb{F}^{ab} \otimes \mathbb{F}^{bc} \otimes \mathbb{F}^{ac}$ is 1-, 2- and 3-concise. Thus $R(\langle a,b,c\rangle) \geq \max(ab,bc,ab)$.

If a concise tensor has rank equal to the dimension lower bound, we say that it has minimal rank.

3.2 Restrictions

The advantage of the framework of tensors is that it is more general than analysing matrix multiplication algorithms. Other tensors can give us information about the matrix multiplication tensor. In order to use this there should be a way to compare tensors.

Definition 3.2.1. If $\alpha_i : V_i \to W_i$ is a collection of linear maps, then $\alpha = \alpha_1 \otimes \ldots \otimes \alpha_k$ is a linear map $V_1 \otimes \ldots \otimes V_k \to W_1 \otimes \ldots \otimes W_k$. For any tensor $T \in V_1 \otimes \ldots \otimes V_k$ and any such map α we say that $\alpha(T)$, or alternatively $\alpha \cdot T$, is a *restriction* from T and write $T \geq \alpha(T)$. We also say α is a restriction from T to $\alpha(T)$. If there are restrictions $T \geq S$ and $S \geq T$, then we say that S and T are *equivalent*.

Remark 3.2.2. As the name suggests, tensors being equivalent defines an equivalence relation. If T and S are equivalent, also denoted $T \simeq S$, then the restriction from T to S is invertibleafter extending the vector spaces so that dimensions match. If two tensors are equivalent then there are different bases for S and T such that T and S look the same if written in their respective bases. Any basis-independent property must thus be the same on all equivalent tensors. Conciseness falls in this class of properties and the proposition below shows that tensor rank does too. Other notions ,such as the slice rank and the value, that are encountered later will also belong to this category.

Restriction allows us to compare tensors. The important property is that tensor rank is non-increasing with respect to restriction.

Proposition 3.2.3. *If T,S are k-tensors with $T \geq S$, then $R(T) \geq R(S)$.*

Proof. Let $\alpha = \alpha_1 \otimes \cdots \otimes \alpha_k$ be the restriction from T to S. Note that for any simple tensor $v_1 \otimes \cdots \otimes v_k$ we get $\alpha(v_1 \otimes \cdots \otimes v_k) = \alpha_1(v_1) \otimes \cdots \otimes \alpha_k(v_k)$. Thus for any simple tensor decomposition $T_1 + \cdots + T_r$ of T we get a simple tensor decomposition $\alpha(T_1) + \cdots + \alpha(T_r)$ of S. Therefore $R(S) \leq R(T)$. □

There are two types of restrictions that are of particular interest because they are so simple.

Definition 3.2.4. Consider two k-tensors S, T each written in some bases. A restriction $\alpha = \alpha_1 \otimes \cdots \otimes \alpha_k$ from S to T is called a *zeroing out* if each α_i, written as a matrix with respect to the bases, is diagonal with only ones and zeros on the diagonal. It is called an *isomorphism* if each matrix is a permutation matrix. If there is an isomorphism relative to some fixed bases between S and T, we write $S \cong T$.

In the notation $S = \sum_{j_1,\ldots,j_k} s(j_1, j_2, \ldots, j_k) x^1_{j_1} \cdots x^k_{j_k}$ a zeroing out corresponds to choosing subsets $Y_i \subseteq X_i$ and removing all terms $x^1_{i_1} \cdots x^k_{i_k}$ for which one of $x^i_{j_i}$ is not in Y_i. An isomorphism is just a relabeling of the x^i_j variables.

3.3 Tensor product

Definition 3.3.1. Let $S \in V_1 \otimes \cdots \otimes V_k$ and $T \in W_1 \otimes \cdots \otimes W_k$ be two k-tensors over the same field. The *(Kronecker) tensor product* of S and T is the k-tensor $S \otimes T$ considered as an element in $(V_1 \otimes W_1) \otimes (V_2 \otimes W_2) \otimes \cdots \otimes (V_k \otimes W_k)$. It satisfies $(s \otimes t)((i_1, j_1), \ldots, (i_k, j_k)) = s(i_1, \ldots, i_k)t(j_1, \ldots, j_k)$ for any bases of all the V_i and W_i. We write $T^{\otimes n}$ for the n-times multiplication of T by itself.

The reason this product is so interesting for matrix multiplication is the identity below.

Lemma 3.3.2. *For all positive integers a, b, c and a', b', c',*

$$\langle a, b, c\rangle \otimes \langle a', b', c'\rangle \cong \langle aa', bb', cc'\rangle.$$

Proof. This can be seen by simply applying the definitions. Take the matrix multiplication tensors $M_1 := \langle a, b, c\rangle = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{c} x_{ij} y_{jk} z_{ki}$ and say $M_2 := \langle a', b', c'\rangle = \sum_{i=1}^{a'} \sum_{j=1}^{b'} \sum_{k=1}^{c'} x'_{ij} y'_{jk} z'_{ki}$. We write out the product

$$M_1 \otimes M_2 = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{c} \sum_{i'=1}^{a'} \sum_{j'=1}^{b'} \sum_{k'=1}^{c'} (x_{ij} \otimes x'_{i'j'})(y_{jk} \otimes y'_{j'k'})(z_{ki} \otimes z'_{k'i'}).$$

The vectors $x_{ij} \otimes x'_{i'j'}$ form a basis of $\mathbb{F}^{a\times b} \otimes \mathbb{F}^{a'\times b'}$. This vector space is isomorphic to $\mathbb{F}^{aa'\times bb'}$. Denote the image of $x_{ij} \otimes x'_{i'j'}$ under this isomorphism by $x^*_{ii'jj'}$. This can be reindexed and then we have i^* based on ii' and j^* based on jj' and basis elements $x^*_{i^*j^*}$ with $1 \leq i^* \leq aa'$ and $1 \leq j^* \leq bb'$. We can similarly get isomorphisms for the Y- and Z-variables. Under these isomorphisms the j^* index must still match for the X- and Y-variables. Similarly the i^* and k^* indices must also match and we get

$$M_1 \otimes M_2 \cong \sum_{i^*=1}^{aa'} \sum_{j^*=1}^{bb'} \sum_{k^*=1}^{cc'} x^*_{i^*j^*} y^*_{j^*k^*} z^*_{k^*i^*}.$$

This is exactly $\langle aa', bb', cc'\rangle$ up to relabeling the basis elements. □

The Kronecker tensor product has a few properties which it inherits from the standard tensor product. These are properties that are not impacted by the bracketing in the definition of the Kronecker tensor product.

Lemma 3.3.3. *For any k-tensors S, T, T_1, T_2,*

- $S \otimes (T_1 + T_2) = S \otimes T_1 + S \otimes T_2$,
- $S \otimes T$ *is isomorphic to* $T \otimes S$.

The tensor product also interacts nicely with restrictions.

Lemma 3.3.4. *If there are restrictions $T \geq T'$ and $S \geq S'$, then there is a restriction $T \otimes S \geq T' \otimes S'$*

Proof. If α and β are the restrictions, then $\alpha \otimes \beta$, seen as a map between the Kronecker products, is a restriction from $T \otimes S$ to $T' \otimes S'$. $\square$

We will encounter multiple rank notions and these behave differently with respect to the Kronecker tensor product. A function f from the set of k-tensors to an ordered field, such as tensor rank, is called submultiplicative if $f(S \otimes T) \leq f(S)f(T)$, multiplicative if $f(S \otimes T) = f(S)f(T)$, and supermultiplicative if $f(S \otimes T) \geq f(S)f(T)$ for all k-tensors S and T. Tensors are vectors in the tensor product of vector spaces, so we can add them together and analogously speak of f as being subadditive, additive or superadditive.

Lemma 3.3.5. *Tensor rank is submultiplicative and subadditive.*

Proof. Take any two k-tensors S, T. By definition of $R(T)$ we can write $T = \sum_{i=1}^{R(T)} T_i$ and $S = \sum_{j=1}^{R(S)} S_j$ with each T_i, S_j being simple. Thus

$$S + T = \sum_{j=1}^{R(S)} S_j + \sum_{i=1}^{R(T)} T_i$$

which is a sum of $R(S) + R(T)$ simple tensors, so $R(S + T) \leq R(S) + R(T)$.
Note further that the tensor product of $v_1 \otimes \cdots \otimes v_k$ and $w_1 \otimes \cdots w_k$ is $(v_1 \otimes w_1) \otimes \cdots \otimes (v_k \otimes w_k)$ which is also a simple tensor. Then by Lemma 3.3.3

$$S \otimes T = \left(\sum_{i=1}^{R(T)} T_i \right) \otimes \left(\sum_{j=1}^{R(S)} S_j \right) = \sum_{i=1}^{R(T)} \sum_{j=1}^{R(S)} T_i \otimes S_j$$

is a sum of $R(T)R(S)$ simple tensors, so $R(S \otimes T) \leq R(S)R(T)$. $\square$

Remark 3.3.6. We can immediately see that $R(\langle 2^n, 2^n, 2^n \rangle) = R(\langle 2, 2, 2 \rangle^{\otimes n}) \leq R(\langle 2, 2, 2 \rangle)^n = 7^n$. In Strassen's algorithm this product corresponds to the recursive application of the identity for 2×2-matrices. The inequality can be strict and later results will imply that $R(\langle 2^n, 2^n, 2^n \rangle) < 7^n$ for large enough n.

3.4 Slice rank

In [EG16] Ellenberg and Gijswijt showed that capsets have exponentially small density. Sawin and Tao [TS16] formalised their approach and in doing so introduced a different notion of rank.

Definition 3.4.1. An *i-slice* is a tensor $T \in V_1 \otimes \cdots \otimes V_{i-1} \otimes \mathbb{F} v_i \otimes V_{i+1} \otimes \cdots \otimes V_k$ where $v_i \in V_i$. A tensor is a *slice* if it is an i-slice for some i. The *slice rank* of a tensor T is the minimum number of slices needed to write T as the sum of slices.

Any simple tensor is also a slice, therefore a decomposition of T into simple tensors is also a decomposition into slices. This immediately implies that $SR(T) \leq R(T)$. Slice rank has some of the properties that tensor rank has.

Proposition 3.4.2. *Slice rank satisfies*

- $SR(T_1 + T_2) \leq SR(T_1) + SR(T_2)$ *for any tensors* T_1 *and* T_2,
- *if there is a restriction* $T_1 \geq T_2$, *then* $SR(T_1) \geq SR(T_2)$ *and*
- $SR(\langle n \rangle_k) = n$ *for any* $n \geq 1$ *and* $k \geq 2$.

Proof. The first two properties have a similar proof as the analogous statement for tensor rank and the third property was shown by Tao in the original blogpost [Tao16]. $\square$

If we can ensure that two tensors S and T have no common basis elements, then slice rank is actually additive.

Definition 3.4.3. For any two k-tensors $S \in V_1 \otimes \cdots \otimes V_k$ and $T \in W_1 \otimes \cdots \otimes W_k$ over the same field *the direct sum* $S \oplus T$ is the sum of the vectors S and T in the vector space $(V_1 \oplus W_1) \otimes \cdots \otimes (V_k \oplus W_k)$.

Lemma 3.4.4 ([Gow21])**.** *For any two k-tensors S and T, there is the equality* $SR(S \oplus T) = SR(S) + SR(T)$.

Remark 3.4.5. It was conjectured by Strassen that tensor rank was also additive under direct sums, but this was disproven by Shitov in 2017 [Shi17].

Slice rank is less well-behaved with respect to the tensor product. We will show in Example 4.2.7 that $SR(\langle a,b,c\rangle) = \min\{ab, bc, ac\}$. We can use this to show slice rank is not submultiplicative in general.

Example 3.4.6. Consider $\langle 1,1,k\rangle$ and $\langle 1,k,1\rangle$ for some integer $k > 1$. We know that $\langle 1,1,k\rangle \otimes \langle 1,k,1\rangle \cong \langle 1,k,k\rangle$. We also know that slice rank is non-increasing under restriction and thus $SR(T) = SR(S)$ if T and S are equivalent. In particular this also holds if they are isomorphic. Thus $SR(\langle 1,1,k\rangle \otimes \langle 1,k,1\rangle) = SR(\langle 1,k,k\rangle) = k > 1 = SR(\langle 1,1,k\rangle) \cdot SR(\langle 1,k,1\rangle)$.

In order to show the slice rank is also not supermultiplicative in general, we can look at the capset tensor. This tensor and its slice rank were the initial motivation for introducing the concept of slice rank.

Example 3.4.7. The capset problem asks what the size of the largest set $A \subseteq \mathbb{F}_3^n$ is without an arithmetic progression, so without distinct $a,b,c \in A$ with $a+b+c = 0$. It was shown by Ellenberg and Gijswijt [EG16] that the size of these sets grows as $O(2.756^n)$. We can approach this problem through the capset tensor. This is a tensor over $X = \{1,x,x^2\}, Y = \{1,y,y^2\}, Z = \{1,z,z^2\}$ which comes from the function $\mathbb{F}_3 \times \mathbb{F}_3 \times \mathbb{F}_3 \to \mathbb{F}_3$ given by $1-(x+y+z)^2$. We take

$$T = 1 - 2xy - 2xz - 2yz - x^2 - y^2 - z^2 = x^0y^0z^0 - 2x^1y^1z^0 - 2x^1y^0z^1 - 2x^0y^1z^1 - x^2y^0z^0 - x^0y^2z^0 - x^0y^0z^2.$$

With the results in chapter 3 one can show that the slice rank of this tensor is 3, but the proof of the capset problem shows that the slice rank of the nth power of T is $O(2.756^n)$.

As an aside, we quickly sketch how this tensor can be used in the capset problem. The tensor $T^{\otimes n}$ can be considered as $\prod_{i=1}^n 1-(x_i+y_i+z_i)^2$ which is a function $\mathbb{F}_3^n \times \mathbb{F}_3^n \times \mathbb{F}_3^n \to \mathbb{F}_3$. This function is zero at all triples $(a,b,c) \in (\mathbb{F}_3^n)^3$ that do not satisfy $a+b+c=0$. We can study the size of a capset A by zeroing out all variables in $\mathbb{F}_3^n \setminus A$. The resulting tensor is zero except at triples (a,a,a). Thus, it has a basis in which the tensor is isomorphic to $\langle |A|\rangle_3$, so the slice rank is $|A|$. On the other hand we can give a slice decomposition of $T^{\otimes n}$ which grows at rate 2.756^n, so $|A| \leq 2.756^n$.

There are some observations that we can make regarding the slice rank under the tensor product.

Lemma 3.4.8. *For any k-tensors S and T, we have $SR(S \otimes T) \leq SR(S)R(T)$.*

Proof. Take a rank decomposition $T_1, \dots, T_r$ of T and a slice rank decomposition $S_1, \dots, S_t$ of S. The product of a slice and a simple tensor is a slice, so $(S_i \otimes T_j)_{i\leq t, j\leq r}$ is a slice rank decomposition of $S \otimes T$. □

Lemma 3.4.9. *If T is not the zero tensor, then $SR(S \otimes T) \geq SR(S)$.*

Proof. If T is a k-tensor and $T \neq 0$, then there is a restriction to $\langle 1\rangle_k$. This restriction zeroes out all variables except for those in a specific triple in the support of T and then scales the remaining variables to get the coefficient equal to 1. Lemma 3.3.4 says there is a restriction $S \otimes T$ to $S \otimes \langle 1\rangle_k \cong S$ and slice rank is non-increasing under restriction, so $SR(S \otimes T) \geq SR(S)$. □

There is one more useful bound for the slice rank.

Lemma 3.4.10. *If T is a k-tensor on finite sets $X_1, \dots, X_k$, then $SR(T) \leq \min_i |X_i|$.*

Proof. We can always write T as a sum of i-slices where each i-slice has i-term equal to one of the elements in X_i. This gives $|X_i|$ slices. This holds true for all i so $SR(T)$ is also at most the minimum of the $|X_i|$. □

In case $SR(T) = \min_i |X_i|$ we say that T has maximal slice rank.

3.5 Asymptotic ranks

What is the asymptotically fastest way one can multiply two $n \times n$-matrices? This was the question that bore fruit to the definition of the exponent ω. So far we have been able to rephrase the 'fastest way one can multiply two $n \times n$-matrices' into a question about the tensor rank of $\langle n, n, n\rangle$. We now move our attention to the 'asymptotic' part.

Lemma 3.5.1 (Fekete's lemma)**.** *For every subadditive sequence* $(a_n)_{n\in\mathbb{N}}$ *the limit* $\lim_{n\to\infty} \frac{a_n}{n}$ *exists and is equal to* $\inf \frac{a_n}{n}$.

This lemma can also be used to say that the limit $\frac{a_n}{n}$ exists for superadditive sequences and that $a_n^{1/n}$ exists for sub- and supermultiplicative sequences. These all reduce to the original formulation by considering sequences $(-a_n)_n, (\log a_n)_n$ and $(-\log a_n)_n$ respectively.

Definition 3.5.2. The *asymptotic tensor rank* of a tensor T is

$$\tilde{R}(T) := \lim_{n\to\infty} R(T^{\otimes n})^{1/n}.$$

The sequence $(R(T^{\otimes n}))_n$ is submultiplicative for any tensor T, so the limit exists by Fekete's lemma. Now that we have this definition, we can give a formal definition of ω in terms of the tensor rank. In this thesis log will always denote the logarithm with base 2.

Definition 3.5.3. The *exponent of fast matrix multiplication* ω is $\omega := \log \tilde{R}(\langle 2, 2, 2\rangle)$.

This definition seems restrictive at first as it only concerns multiplying $2^n \times 2^n$-matrices, but it can be shown that for all integers $a, b, c > 1$ we have $\omega = \frac{3\log \tilde{R}(\langle a,b,c\rangle)}{\log abc}$. Furthermore, this definition of ω also satisfies that $O(n^\omega)$ is asymptotically the complexity of multiplying two $n \times n$-matrices [BCS96]. It is not known whether the exponent depends on the underlying field $\mathbb{F}$. For this reason the notation $\omega(\mathbb{F})$ would potentially be more appropiate, but we will just write ω in this thesis.

Lemma 3.5.4. $2 \leq \omega \leq \log 7$.

Proof. Matrix multiplication tensors are concise, so $R(\langle 2^n, 2^n, 2^n\rangle) \geq 2^{2n}$. Thus $\tilde{R}(\langle 2, 2, 2\rangle) \geq 2^2$. On the other hand R is submultiplicative, so Fekete's lemma says that $\tilde{R}(\langle 2, 2, 2\rangle) \leq R(\langle 2, 2, 2\rangle) = 7$. Applying the logarithm gives the claimed bounds. $\square$

We are also interested in the asymptotic behaviour of the slice rank. The slice rank is not sub- or supermultiplicative in general, so we cannot use Fekete's lemma to say a similar limit exists.

Definition 3.5.5. The *asymptotic slice rank* $\widetilde{\mathrm{SR}}(T)$ of a tensor T is

$$\widetilde{\mathrm{SR}}(T) := \limsup_n SR(T^{\otimes n})^{1/n}.$$

These asymptotic rank notions inherit some properties from the normal rank notion. They are also non-increasing under restrictions. For the asymptotic slice rank we also have the dimension upper bound $\widetilde{\mathrm{SR}}(T) \leq \min_i \dim V_i$ for any tensor in $V_1 \otimes \cdots \otimes V_k$ and for i-concise tensors we have $\tilde{R}(T) \geq \dim V_i$. Another useful observation is that $\tilde{R}(T^{\otimes k}) = \tilde{R}(T)^k$ and also $\widetilde{\mathrm{SR}}(T^{\otimes k}) = \widetilde{\mathrm{SR}}(T)^k$, which needs Lemma 3.4.9 because of the limsup.

3.6 Border rank

Tensor rank is an upper bound for the asymptotic tensor rank, but often the asymptotic tensor rank is much smaller. Finding the tensor rank for ever larger powers of T is also difficult as finding the tensor rank is an NP-hard problem. In 1980 the border rank was introduced [BLR80], which gives a better upper bound for the asymptotic rank and is only based on the tensor T. We shall now introduce this concept.

Example 3.6.1. The tensor $W = x_1y_1z_2 + x_1y_2z_1 + x_2y_1z_1$ has tensor rank 3. However, there is some tensor W' such that we can write

$$W = \frac{1}{\varepsilon}\big((x_1 + \varepsilon x_2)(y_1 + \varepsilon y_2)(z_1 + \varepsilon z_2) - x_1y_1z_1 + \varepsilon^2 W'\big). \tag{3.1}$$

This parameter ε enables us to write W as the sum of some rank 2 tensor and some terms which are linear in ε. If W was defined over the real or complex numbers you could say that the rank 2 tensor approaches W as $\varepsilon \to 0$. This idea of approximating tensors, such as W, by lower rank tensors is captured in the border rank.

Definition 3.6.2. Let $T \in V_1 \otimes \cdots \otimes V_k$ be a k-tensor over $\mathbb{F}$. We say a tensor S is a border simple tensor for T if it is a simple tensor in the tensor product $V_1(\varepsilon) \otimes \cdots \otimes V_k(\varepsilon)$ over $\mathbb{F}(\varepsilon)$ where ε is a formal variable. The *border rank* $\underline{R}(T)$ of T is r if this is the minimum integer such that there are r border simple tensors $S_1, \dots, S_r$ and some tensors $T_1, \dots, T_u \in V_1 \otimes \cdots \otimes V_k$ such that $S_1 + \cdots + S_r = \varepsilon^t(T + \sum_{i=1}^{u} \varepsilon^i T_i)$ for some $t \in \mathbb{Z}$.

More generally, the approximating behaviour can be captured in an approximate version of restriction.

Definition 3.6.3. A *degeneration* from $S \in V_1 \otimes \cdots \otimes V_k$ to $T \in W_1 \otimes \cdots \otimes W_k$ is a linear map $\alpha = \alpha_1 \otimes \cdots \otimes \alpha_k$ where each α_i is a $\mathbb{F}(\varepsilon)$-linear map $V_i(\varepsilon) \to W_i(\varepsilon)$ for which there exists $t \in \mathbb{Z}$ and finitely many tensors $T_1, \dots, T_u \in V_1 \otimes \cdots \otimes V_k$ such that $\alpha(S) = T + \sum_{i=1}^{u} \varepsilon^i T_i$. We write $S \trianglerighteq T$.

Any restriction is also a degeneration where $u = 0$. The following lemma says that the border rank can be used to upper bound the asymptotic rank.

Lemma 3.6.4 ([Bin80])**.** *All tensors T have $\tilde{R}(T) \leq \underline{R}(T) \leq R(T)$.*

3.7 Entropy and types

In later chapters we will use entropy to give expressions for the asymptotic slice rank.

Definition 3.7.1. For a discrete random variable X taking values in A with probability distribution p the *entropy* of X is

$$H(X) := -\sum_{a \in A} p(a) \log p(a).$$

We may also write $H(p)$ to mean the same thing. In this expression we let $0 \log 0$ be equal to 0.

This captures the information contained in finding the value of an instance of X. The relevance of entropy for the slice rank and related notions arises through the idea of types. We write $\mathcal{P}(A)$ for the set of probability distributions on A.

Definition 3.7.2. Let A be a finite set and let $p \in \mathcal{P}(A)$ be some probability distribution. We say that some sequence $I \in A^n$ is of type p if $|\{i \in [n] : I_i = a\}| = p(a)n$ for all $a \in A$.

Lemma 3.7.3. *Fix a set A and pick a natural number n. The number of types $p \in \mathcal{P}(A)$ such that there are sequences $I \in A^n$ of type p is polynomial in n.*

Proof. If I is of type p, then $p(a)n$ must be an integer for all $a \in A$. We know that $\sum_{a \in A} p(a)n = n$, so $p(a)n$ must be in $\{0, 1, \dots, n\}$ for each a. There are only $n + 1$ choices for $p(a)n$. Because of this, the number of types is at most $(n + 1)^{|A|}$, which is polynomial in n. ☐

Entropy is connected to types through the following lemma.

Lemma 3.7.4. *Let $p \in \mathcal{P}(A)$ be a distribution whose values are all integer multiples of $\frac{1}{n}$. Then*

$$|\{I \in A^n : I \text{ is of type } p\}| = \binom{n}{[np(a)]_{a \in A}} \tag{3.2}$$

and this expression is $2^{(H(p)+o(1))n}$ as $n \to \infty$.

The following properties of entropy will be important to us.

Lemma 3.7.5 ([Zui18]). *Let p_1 and p_2 be some probability distributions on finite sets A_1 and A_2 then $H(p_1)+H(p_2) \geq H(p)$ for any probability distribution $p \in \mathcal{P}(A_1 \times A_2)$ with marginal distributions p_1 and p_2 and equality holds if $p(a_1, a_2) = p(a_1)p(a_2)$ for all $a_1 \in A_1$ and $a_2 \in A_2$.*

Theorem 3.7.6. *The entropy is concave in the probability mass function p, so*

$$H(\lambda p_1 + (1-\lambda)p_2) \geq \lambda H(p_1) + (1-\lambda)H(p_2)$$

for $0 \leq \lambda \leq 1$ and any probability mass functions p_1, p_2 on the same set.

Corollary 3.7.7. *Fix a finite set A. The uniform distribution is the unique probability mass function $p \in \mathcal{P}(A)$ that maximises the entropy.*

3.8 Laser method

The most successful method to obtain new upper bounds for ω is called the Laser method. In this method one starts with a certain tensor T, which we call the base tensor or start tensor. The asymptotic tensor rank of T should be known and as small as possible, but we also need that large powers of T contain many large matrix multiplication tensors. More precisely we need to find zeroing outs from $T^{\otimes n}$ to $\bigoplus_{i=1}^{m} \langle a_i, b_i, c_i \rangle$ with a_i, b_i, c_i large and m as large as possible. Intuitively this implies that you do not need a lot of multiplications to find $T^{\otimes n}$, but we can simulate many large matrix multiplications within T, thus we do not need a lot of multiplications for these matrix multiplications. The bound on ω is made precise by Schönhage's tau theorem, also called the asymptotic sum inequality.

Theorem 3.8.1 (Schönhage). *If there is a degeneration $T^{\otimes n} \trianglerighteq \bigoplus_i \langle a_i, b_i, c_i \rangle$, then $\sum_i (a_i b_i c_i)^{\omega/3} \leq \tilde{R}(T)^n$.*

Based on this theorem there is a specific value associated to a tensor which is extremely relevant for the laser method.

Definition 3.8.2. The *value of a 3-tensor T* is defined as

$$V_\rho(T) := \sup_{n,\ \text{degenerations}\ \trianglerighteq} \left\{ \left(\sum_i (a_i b_i c_i)^\rho \right)^{1/n} \mid T^{\otimes n} \trianglerighteq \bigoplus_i \langle a_i, b_i, c_i \rangle \right\}.$$

For each ρ this value is a kind of asymptotic rank notion and also has similar properties.

Lemma 3.8.3. *Let S, T be two 3-tensors then*

(i) $V_\rho(S \oplus T) \geq V_\rho(S) + V_\rho(T)$,

(ii) $V_\rho(S \otimes T) \geq V_\rho(S)V_\rho(T)$,

(iii) $V_\rho(S) \geq V_\rho(T)$ *if* $S \trianglerighteq T$ *and*

(iv) $V_\rho(S)^{\rho/\rho'} \geq V_\rho(S) \geq V_{\rho'}(S)$ *if* $\rho \geq \rho'$.

All these properties follow by direct analysis of the definition and the fact that Lemma 3.3.4 also holds for degenerations. The multiplicativity also implies that $V_\rho(T)$ could actually be defined as a limit as $n \to \infty$. One last important property is that there is no better degeneration than the identity in case T is a matrix multiplication tensor. Thus $V_\rho(\langle a, b, c \rangle) = (abc)^\rho$. The value allows us to write down a different version of the asymptotic sum inequality.

Theorem 3.8.4 ([DS13]). *Let T be a 3-tensor. Then $V_{\omega/3}(T) \leq \tilde{R}(T)$.*

Coppersmith and Winograd identified a class of useful tensors for the laser method and Alman called these laser-ready. Before these can be introduced we first need to introduce some other concepts. These concepts can be defined for general k-tensors, but will only be used for 3-tensors and so we restrict ourselves to these.

Definition 3.8.5. Consider a triple of partitions $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ where $X = \bigcup_{X_i \in \mathcal{X}} X_i$, $Y = \bigcup_{Y_i \in \mathcal{Y}} Y_i$, $Z = \bigcup_{Z_i \in \mathcal{Z}} Z_i$. We call each X_i an X*-block* and similarly define Y- and Z-blocks. We call a set $X_i \times Y_j \times Z_k$, or equivalently (i, j, k), a *block triple.* Given a tensor T on X, Y, Z and a triple of partitions of X, Y, Z we define *partition subtensors* T_{ijk} for each block triple (i, j, k). This subtensor T_{ijk} is the tensor $\sum_{x \in X_i, y \in Y_j, z \in Z_k} t(x, y, z)xyz$, which is T restricted to $X_i \times Y_j \times Z_k$. Some of these partition subtensors are potentially zero. We say that the *block support* of T are those block triples (i, j, k) for which $T_{ijk} \neq 0$.

Remark 3.8.6. Let T be a tensor with a triple of partitions $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ then any power $T^{\otimes n}$ inherits a triple of partitions. We take the partition $X^n = \bigcup_{X_{i_j} \in \mathcal{X}\ \forall j \in [n]} X_{i_1} \times \cdots \times X_{i_n}$ and Y^n, Z^n have similar partitions. We call these the power partitions.

In the laser method we specify a partition of T and take the power partitions of $T^{\otimes n}$. Next we specify some blocks and zero out all variables in those blocks. In the end we aim to be left with a direct sum of partition subtensors. This direct sum condition can also be translated to a property of the block support

Definition 3.8.7. A set $\Psi \subseteq X \times Y \times Z$ is *independent* if each $w \in X \cup Y \cup Z$ is in at most one element in Ψ.

Thus we zero out blocks until the block support becomes independent. If the block support contains almost all of $X \times Y \times Z$, then we would need to zero out almost all blocks. We impose some structure on the initial tensor to ensure that this does not happen.

Definition 3.8.8. A set $\Psi \subseteq X \times Y \times Z$ is *tight* if there are injective functions $\alpha_X : X \to \mathbb{Z}$, $\alpha_Y : Y \to \mathbb{Z}$ and $\alpha_Z : Z \to \mathbb{Z}$ such that $\alpha_X(x) + \alpha_Y(y) + \alpha_Z(z) = 0$ for all $(x, y, z) \in \Psi$.

Definition 3.8.9. A triple of partitions for a tensor is laser-ready if

1. Each partition subtensor in the block support is isomorphic to a matrix multiplication tensor whose dimensions match up with the sizes of the blocks.

2. The block support is a tight set.

We combine a lot of the new terminology in the following lemma.

Lemma 3.8.10. *A tensor with laser-ready partition is concise if each block is in at least one block triple in the block support.*

Proof. Suppose a tensor T over X, Y, Z is a counterexample. The tensor is laser-ready with each block used in a triple in the block support, but it is not concise. Without loss of generality we can say it is not 1-concise. There must be a block, say X_0, with $\bar{x} \in X_0$ such that $(y, z) \mapsto t(\bar{x}, y, z)$ is not linearly independent from the other $t_x : (y, z) \mapsto t(x, y, z)$. For all y, z we have the equality $t(\bar{x}, y, z) = \sum_{x \neq \bar{x}} \lambda_x t(x, y, z)$ for some $\lambda_x \in \mathbb{F}$. The block support is tight, so there are injective functions α_X, α_Y and α_Z such that $\alpha_X(i) + \alpha_Y(j) + \alpha_Z(k) = 0$ for all (i, j, k) in the block support. Any distinct i, i' cannot have $\alpha_X(i) = \alpha_X(i')$, so at most one of them can be in a block triple in the support with some fixed (j, k). We assumed that each block gets used in a block triple in the block support. Let j_0, k_0 be such that $(0, j_0, k_0)$ is in the block support. We have seen that for $i \neq 0$ the triple (i, j_0, k_0) is not in the block support. Therefore any xyz with $y \in Y_{j_0}$, $z \in Z_{k_0}$ and $x \in X_i$ for some $i \neq 0$ is not in the support of t. For any $y \in Y_{j_0}$ and $z \in Z_{k_0}$ this means that $t(\bar{x}, y, z) = \sum_{x \in X_0, x \neq \bar{x}} \lambda_x t(x, y, z)$. This would imply that T_{0jk} is not 1-concise, but it is isomorphic to a matrix multiplication tensor whose dimensions match the blocks sizes, so it must be concise. Contradiction. $\square$

3.9 Symmetry

The laser method has been applied to tensors which carry some symmetry. A lot of the analysis for the laser method or slice rank also gets easier if we restrict ourselves to tensors which have this sort of symmetry.

Definition 3.9.1. A set $\Phi \subseteq A \times \cdots \times A$ is in *symmetric form* if $(a_1, \ldots, a_k) \in \Phi$ implies that the rotated element $(a_2, \ldots, a_k, a_1) \in \Phi$. A set $\Phi \subseteq X_1 \times \cdots \times X_k$ is called *symmetric* if there is a set A and bijections $f_i : X_i \to A$ such that $(f_1 \times \cdots \times f_k)(\Phi) \subseteq A \times \cdots \times A$ is in symmetric form.

We can extend this definition about sets to a definition about tensors with respect to a certain basis.

Definition 3.9.2. For a tensor T on $X_1, \dots, X_k$ we define $\mathrm{rot}(T)$ to be the tensor on $X_2, \dots, X_k, X_1$ which just rotates the entries and as a result has coefficients $\mathrm{rot}(t)(x_2^{j_2}, \dots, x_k^{j_k}, x_1^{j_1}) = t(x_1^{j_1}, \dots, x_k^{j_k})$. We say that a tensor is symmetric if $T \cong \mathrm{rot}(T)$.

If T is symmetric, then it has symmetric support and hence it is isomorphic to a tensor in symmetric form. Additionally, note that rotation commutes with the tensor product, so $\mathrm{rot}(S \otimes T) \cong \mathrm{rot}(S) \otimes \mathrm{rot}(T)$. Write rot^i for the i times application of rot. Any k-tensor T satisfies $\mathrm{rot}^k(T) = T$. We can define the symmetrised tensor $T_{sym} := \bigotimes_{i=0}^{k-1} \mathrm{rot}^i(T)$ and this tensor is indeed symmetric as

$$\mathrm{rot}(T_{sym}) \cong \bigotimes_{i=0}^{k-1} \mathrm{rot}(\mathrm{rot}^i(T)) = \bigotimes_{i=0}^{k-1} \mathrm{rot}^{i+1}(T) = \left(\bigotimes_{i=1}^{k-1} \mathrm{rot}^i(T)\right) \otimes T \cong T_{sym}.$$

Observation 3.9.3. *For any tensor T we have $R(\mathrm{rot}(T)) = R(T)$ and $V_\rho(\mathrm{rot}(T)) = V_\rho(T)$*

This is true because any tensor rank decomposition of T can be rotated to give a rank decomposition of $\mathrm{rot}(T)$ and any degeneration can also be rotated. This observation implies by sub- and supermultiplicativity respectively that for 3-tensors $\tilde{R}(T_{sym}) \leq \tilde{R}(T)^3$ and $V_\rho(T_{sym}) \geq V_\rho(T)^3$. This gives a reason for preferring symmetric tensors for the laser method.

The probability distributions and partitions we defined can also be symmetric.

Definition 3.9.4. If $\Phi \subseteq X_1 \times \cdots \times X_k$ is some symmetric set, then $\mathcal{P}_{sym}(\Phi) := \{p \in \mathcal{P}(\Phi) : p(x) = p(x')$ if x' is the rotated element of x in $\Phi\}$ is the set of symmetric distributions.

Definition 3.9.5. A triple of partitions $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ for a symmetric tensor T is symmetric if the block support is in symmetric form and for (i,j,k) in the block support $T_{ijk} \cong T_{jki}$.

3.10 Slice rank for lower bounds

With all the terminology we have established we can now state the connection between slice rank and fast matrix multiplication. Observe the following:

$$V_{2/3}(\bigoplus_i \langle n_i, n_i, n_i \rangle) \geq \sum_i n_i^2 = SR(\bigoplus_i \langle n_i, n_i, n_i \rangle).$$

In fact equality holds. Using Schönhage's tau theorem to try and prove $\omega = 2$ will thus induce connections to the slice rank. In [Alm18] Alman found a way to use this connection. All current methods for bounding ω take a tensor T and construct a degeneration from $T^{\otimes n}$ to a direct sum of matrix multiplication tensors. Let $\omega_u(T)$ be the best bound on ω one could find by using a degeneration from a power of T. The fact that asymptotic slice rank is non-increasing under degeneration implies that $V_2(T)$ is close to $\tilde{R}(T)$ if and only if $\widetilde{\mathrm{SR}}(T)$ is close to $R(T)$. The exact connection is captured in the following lower bound.

Theorem 3.10.1 ([Alm18])**.** *For symmetric tensors T we have $\omega_u(T) \geq 2\frac{\log \tilde{R}(T)}{\log \widetilde{\mathrm{SR}}(T)}$.*

4 Combinatorial estimation of the slice rank

All expressions and bounds for the slice rank arise through an analysis of the support. In this chapter we shall mostly focus on results for the support and in the next chapter we translate these to results about the slice rank. One can regard the support as a hypergraph, which means that analysis of the support can be done in combinatorial context.

Definition 4.0.1. A *hypergraph* is a pair of sets (V, E) where $E \subseteq \mathcal{P}(V)$. The elements of V are called vertices and E is the set of edges. If each edge is a set of size k, then (V, E) is a *k-uniform* hypergraph. A hypergraph is called *k-partite* if its vertices can be partitioned into k classes such that each edge contains at most one vertex per partition class.

Any k-tensor T on $X_1, \dots, X_k$ defines a k-uniform k-partite hypergraph $\operatorname{supp}(T) = (V_T, E_T)$ through its support. Take $V_T = X_1 \cup \dots \cup X_k$ and for $x_i \in X_i$, let $\{x_1, \dots, x_k\} \in E_T$ if $x_1 x_2 \cdots x_k$ is in the support of T. This edge can be uniquely identified with $(x_1, \dots, x_k) \in X_1 \times \dots \times X_k$. We will often use this identification of the set of edges with a subset $E \subseteq X_1 \times \dots \times X_k$. Such a subset uniquely defines a k-uniform k-partite hypergraph and we use this bijection to speak of a hypergraph and its edge set interchangeably.

Remark 4.0.2. If a tensor is defined over $\mathbb{F}_2$, then it is uniquely determined by its support. Therefore there is a bijection between k-tensors over $\mathbb{F}_2$ and k-uniform k-partite hypergraphs.

In order to study the multiplicative behaviour of tensors through the support, we need to introduce a product on hypergraphs that mimics the Kronecker tensor product. We choose this product such that $\operatorname{supp}(S \otimes T) = \operatorname{supp}(S) \times \operatorname{supp}(T)$.

Definition 4.0.3. Let $G = (V_G, E_G), H = (V_H, E_H)$ be two k-uniform, k-partite hypergraphs with a fixed order on the partitions $V_{G,1}, \dots, V_{G,k}$ and $V_{H,1}, \dots, V_{H,k}$. Then the (Kronecker) product $G \times H$ of these hypergraphs is the hypergraph with vertex set $\bigcup_{i=1}^k V_{G,i} \times V_{H,i}$ and edge set consists of subsets $\{(v_1, v_1'), \dots, (v_k, v_k')\}$ of size k where both $\{v_1, \dots, v_k\} \in E_G$ and $\{v_1', \dots, v_k'\} \in E_H$.

Regarding the edge sets as subsets of a product, $E_{G \times H}$ is simply $E_G \times E_H$ regarded as a subset of $(V_{G,1} \times V_{H,1}) \times \dots \times (V_{G,k} \times V_{H,k})$.

From this we can also define the nth power of a hypergraph. For an edge set $E \subseteq X_1 \times \dots \times X_k$ the nth power E^n can be seen as a subset of $X_1^n \times \dots \times X_k^n \cong (X_1 \times \dots \times X_k)^n$. Recall that in such products we can talk about the type of elements.

4.1 Vertex cover numbers

The notion equivalent to slice rank in the hypergraph setting is the vertex cover number. This parameter is actively being studied in its own right [DeB+21; Dia18; LW22] and has some open conjectures itself. We will not study these, but only consider the aspects relevant for the slice rank.

Definition 4.1.1. A *vertex cover* of a hypergraph $H = (V, E)$ is a function $f : V \to \{0, 1\}$ such that for any edge $e \in E$ we have $\sum_{v \in e} f(v) \geq 1$. The *vertex cover number* $\tau(H)$ is the smallest integer r such that there exists a vertex cover f of H with $\sum_{v \in V} f(v) = r$. If f is a function $V \to [0, 1]$ satisfying the same conditions, then we call f a *fractional vertex cover* and the *fractional vertex cover number* $\tau^*(H)$ of a hypergraph is the infimum of all reals r for which there is a fractional vertex cover with $\sum_{v \in V} f(v) = r$.

Remark 4.1.2. The infimum that defines τ^* is attained. It is the infimum of a linear function over a compact set.

Lemma 4.1.3. *Let $k \geq 2$. Any k-uniform hypergraph H has $\frac{2}{k}\tau(H) \leq \tau^*(H) \leq \tau(H)$.*

Proof. Any vertex cover is a fractional vertex cover, so that immediately gives $\tau^*(H) \leq \tau(H)$ for any H. The other bound requires a more complicated argument. See for example Proposition 4.9 from [Der22]. We can more easily show the bound $\frac{1}{k}\tau(H) \leq \tau^*(H)$ which is enough for the later statements in this thesis.
Let f be a fractional vertex cover of $H = (V, E)$. Define $g : V \to \{0, 1\}$ by setting $g(x) = 1$ if and only if $f(x) \geq \frac{1}{k}$. Then indeed we have $\sum_{v \in V} g(v) \leq \sum_{v \in V} k f(v) = k\tau^*(H)$, so we just need to show g is a vertex cover. For any edge e, we have $\sum_{v \in e} f(v) \geq 1$ and $|e| = k$, so there must be one v_0 for which $f(v_0) \geq \frac{1}{k}$. Thus $g(v_0) = 1$ and $\sum_{v \in e} g(v) \geq 1$, so g is a vertex cover. □

Example 4.1.4. Consider the capset tensor $T = 1 - 2xy - 2yz - 2zx - x^2 - y^2 - z^2$ with bases $\{1, x, x^2\}$, $\{1, y, y^2\}$ and $\{1, z, z^2\}$. There is a fractional vertex cover f of the support assigning $\frac{1}{2}$ to each of the ones, $\frac{1}{4}$ to each of the linear terms and 0 to each of the quadratic basis elements. This means $\tau^*(\operatorname{supp}(T)) \leq \frac{9}{4}$. For the vertex cover number we observe that the degrees of the vertices in $\operatorname{supp}(T)$ are $1, 2$ or 4. The number of edges is 7, so either we use two vertices of degree 4 or a vertex cover needs at least 3 vertices. However, the vertices of degree 4 are the ones and two of those do not cover all of the xy, yz, zx edges. We conclude that $\tau(\operatorname{supp}(T)) \geq 3$. This discrepancy between the vertex cover numbers explains some of the behaviour of the slice rank of the capset tensor that was mentioned in Chapter 2.

Corollary 4.1.5. *Let $H^{\times n}$ denote the nth Kronecker power of H, then $|\tau(H^{\times n})^{1/n} - \tau^*(H^{\times n})^{1/n}| \to 0$ as $n \to \infty$.*

Proof. For any n Lemma 4.1.3 implies that $\left(\frac{2}{k}\tau(H^{\times n})\right)^{1/n} \leq \tau^*(H^{\times n})^{1/n} \leq \tau(H^{\times n})^{1/n}$. Therefore $(1 - \left(\frac{2}{k}\right)^{1/n})\tau(H^{\times n})^{1/n} \geq \tau(H^{\times n})^{1/n} - \tau^*(H^{\times n})^{1/n} \geq 0$ and the upper bound goes to zero. □

The fractional vertex cover number behaves nicely with respect to the hypergraph product.

Lemma 4.1.6. *Let G, H be two k-uniform, k-partite hypergraphs then $\tau^*(G \times H) \geq \tau^*(G)\tau^*(H)$.*

Proof. The definition just says that τ^* is the solution of a linear program. We define its dual, the fractional matching number, and analyse that instead. For a hypergraph $H = (V, E)$ the fractional matching number ν^* is defined by the following linear program.

$$\begin{aligned} \nu^*(H) = \quad & \max \sum_{e \in E} f(e) \\ \text{s.t.} \quad & f : E \to [0,1], \\ & \sum_{e : v \in e} f(e) \leq 1 \quad \forall v \in V \end{aligned}$$

By the strong duality theorem we get $\nu^*(H) = \tau^*(H)$. Now we show that ν^* is supermultiplicative. Take two hypergraphs G, H with optimal functions g, h respectively. The edges in the hypergraph $G \times H$ can be considered as pairs (e, e') where e is an edge in G and e' is an edge in H. Then we define the function $f : E_{G\times H} \to [0,1]$ for which $f((e,e')) = g(e)h(e')$. For any (v, v') in the vertex set of $G \times H$, we have $\{\text{edges } \eta : (v, v') \in \eta\} = \{(e, e') : v \in e \text{ and } v' \in e'\}$. This implies that

$$\sum_{\eta\colon (v,v')\in\eta} f(\eta) = \sum_{(e,e')\colon v\in e, v'\in e'} f((e,e')) = \sum_{e\colon v\in e} g(e) \sum_{e'\colon v'\in e'} h(e')$$

Each of the factors on the right side is at most one, because both g and h are fractional matchings. This means that f is a fractional matching, which implies that

$$\nu^*(G \times H) \geq \sum_{e''\in E_{G\times H}} f(e'') = \sum_{(e,e')\in E_G \times E_H} g(e)h(e') = \nu^*(G)\nu^*(H).$$

Therefore ν^*, and also τ^*, is supermultiplicative. □

4.2 Oblique support

A vertex cover gives a slice rank decomposition and can thus be used for an upper bound.

Lemma 4.2.1 ([TS16])**.** *For any tensor T, we have $SR(T) \leq \tau(\text{supp}(T))$.*

Proof. If T is a k-tensor and f is a minimal vertex cover of $\text{supp}(T)$, then we can build subtensors $T_1, \ldots, T_k$ which give a slice decomposition. Start with each $T_j = 0$. For each non-zero term $t(i_1, \ldots, i_k)x^1_{i_1} \cdots x^k_{i_1}$ in T there is at least one j such that $f(x^j_{i_j}) = 1$. Add $t(i_1, \ldots, i_k)x^1_{i_1} \cdots x^k_{i_1}$ to T_j for one such j. After repeating this procedure for all terms in the support, we have $T = T_1 + \ldots + T_k$. Let a_j be the size of the subset of X_j-variables that appear in the support of T_j. Each of the a_j of these variables x^j_i must satisfy $f(x^j_i) = 1$ by our construction, so $\sum_{j=1}^k a_j \leq \tau(\text{supp}(T))$. We form slices of T_j by grouping all terms with the same X_j-variable together. Thus $SR(T_j) \leq a_j$. We conclude that $SR(T) \leq \sum_{i=1}^j SR(T_j) \leq \tau(\text{supp}(T))$. □

In their blogpost [TS16], Tao and Sawin also included a lower bound for the slice rank. Consider any $\Phi \subseteq X_1 \times \ldots \times X_k$ and define linear orders on each of $X_1, \ldots, X_k$. The product order of these is a partial order on $X_1 \times \ldots \times X_k$ and therefore it also defines a partial order on Φ. For such a partial order we write $\max(\Phi) \subseteq \Phi$ for the subset of maximal elements. This subset can itself also be considered as a hypergraph.

Lemma 4.2.2 ([TS16])**.** *For any k-tensor T and any product order on* $\text{supp}(T)$ *of the form above, we have $SR(T) \geq \tau(\max(\text{supp}(T)))$.*

Proof. We follow the proof in [Zui18]. Consider tensor $T \in V_1 \otimes \cdots \otimes V_k$ with bases $X_1, \ldots, X_k$. Let $<_i$ denote the ith linear order and let $<$ be the product order. Pick a slice rank decomposition of T and for any i, let $\{v_1^i, \ldots, v_{r_i}^i\} \subseteq V_i$ be the set of vectors used in the i-slices. Let $W_i = \operatorname{span}\{v_1^i, \ldots, v_{r_i}^i\} \subseteq V_i$. Let $W_i' \subseteq V_i^*$ be the elements in the dual space that vanish on W_i. Then pick a basis B_i for W_i' with the following property: with respect to the standard dual basis X_i^* (with elements ordered according to $<_i$), the matrix with elements B_i as columns is in reduced row echelon form. This means that each column is of the form $(* \cdots * 1 0 \cdots 0)^T$ and the pivot elements (the 1's) are all in different rows. Let $I_i \subseteq |X_i|$ be the indices of the pivot elements. Then consider the set $Y_i = \{x_j \in X_i : j \notin I_i\}$. Note that $|Y_i| = \dim W_i$, because there are $|B_i| = \dim W_i' = \dim V_i - \dim W_i$ elements in I_i. We show that $Y_1 \cup \cdots \cup Y_k$ is a vertex cover of $\max(\operatorname{supp}(T))$. This vertex cover has size $\sum_{i=1}^k \dim W_i = SR(T)$, so $\tau(\max(\operatorname{supp}(T))) \le SR(T)$.

Let $e = (x_{e_1}, \ldots, x_{e_k})$ be an edge in $\max(\operatorname{supp}(T))$. Suppose it is not covered by $Y_1 \cup \cdots \cup Y_k$. For all i, it must be the case that $e_i \in I_i$. Let $w_{e_i}^i \in B_i$ be the element whose pivot is at position e_i. The form of $w_{e_i}^i$ implies that $\ker(w_{e_i}^i)$ contains all $\{x_j : j >_i e_i\}$. This means that $w = \bigotimes_{i=1}^k w_{e_i}^i \in W_1' \otimes \cdots \otimes W_k'$ sends T to

$$\begin{aligned} w(T) &= \sum_{\eta \le e} t_\eta w(x_{\eta_1} \cdots x_{\eta_k}) \\ &= t_e x_{e_1} \cdots x_{e_k} + \sum_{\eta < e} t_\eta w(x_{\eta_1} \cdots x_{\eta_k}). \end{aligned}$$

This last sum is contained in $\operatorname{span}\{x_\eta : \eta < e\}$. Therefore the $t_e x_{e_1} \cdots x_{e_k}$ term does not get cancelled out and $w(T) \neq 0$. On the other hand, $w \in W_1' \otimes \cdots \otimes W_k'$ which is zero on each of the slices that form T. This is a contradiction, so $Y_1 \cup \cdots \cup Y_k$ is a vertex cover. □

In some cases the upper bound and lower bound agree, which means the support determines the slice rank. One such case is when $\max(\operatorname{supp}(T)) = \operatorname{supp}(T)$.

Definition 4.2.3. A set $\Phi \subseteq X_1 \times \cdots \times X_k$ is *oblique* if there exist linear orders on $X_1, \ldots, X_k$ for which all elements in Φ are maximal in Φ under the product order.

Remark 4.2.4. If a tensor T has oblique support, then $SR(T) = \tau(\operatorname{supp}(T))$. Note that the right hand side is independent of the field the tensor was defined over.

We record a few small facts about oblique sets.

Lemma 4.2.5. *If $\Phi \subseteq X_1 \times \cdots \times X_k$ and $\Phi' \subseteq X_1' \times \cdots \times X_k$ are oblique, then so is any $\Psi \subseteq \Phi$ and $\Phi \times \Phi'$.*

Proof. The linear orders that were used for Φ and Φ' can be used for Ψ and the lexicographic order these define can be used for $\Phi \times \Phi'$. □

Lemma 4.2.6. *All tight sets are oblique.*

Proof. Given an injective function $\alpha : A \to \mathbb{Z}$ we can define an order on A by saying $a \le b$ if and only if $\alpha(a) \le \alpha(b)$. If $\Phi \subseteq X \times Y \times Z$ is tight, then there are functions $\alpha_X, \alpha_Y, \alpha_Z$ that define orders on X, Y, Z in this way. For any (x, y, z) and (x', y', z') in Φ we have $\alpha_X(x) + \alpha_Y(y) + \alpha_Z(z) = 0$ and $\alpha_X(x') + \alpha_Y(y') + \alpha_Z(z') = 0$. Therefore one of $\alpha_X(x') > \alpha_X(x)$, $\alpha_Y(y') > \alpha_Y(y)$ and $\alpha_Z(z') > \alpha_Z(z)$ cannot be true, so $(x', y', z') \not> (x, y, z)$. We conclude that all elements in Φ are maximal under this order. □

Example 4.2.7. We show that the matrix multiplication tensor in its standard form has oblique support. In fact we can show that it is tight. Let $\langle a, b, c\rangle$ be a matrix multiplication tensor. We may regard the sets it is defined over as $X = [a] \times [b]$,$Y = [b] \times [c]$, $Z = [c] \times [a]$. Take $N = 2abc$ and consider the maps

$$\begin{aligned} i_X &: X \to \mathbb{Z}, \quad (i, j) \mapsto Ni - j \\ i_Y &: Y \to \mathbb{Z}, \quad (i, j) \mapsto i - N^2 j \\ i_Z &: Z \to \mathbb{Z}, \quad (i, j) \to N^2 i - Nj. \end{aligned}$$

These are injective and for $((i,j),(j',k),(k',i'))$ in the support we have $i=i', j=j'$ and $k=k'$, so

$$i_X((i,j))+i_Y(j',k)+i_Z((k',i'))=Ni-j+j'-N^2k+N^2k'-Ni'=0.$$

This proves that the support is tight. The degree of a vertex in the support hypergraph is at most $\max(a,b,c)$ and there are abc edges. Thus any vertex cover needs at least $\min(bc,ac,ab)$ vertices. This is also the upper bound by Lemma 3.4.10. Therefore $SR(\langle a,b,c\rangle)=\min(bc,ac,ab)$.

In their blogpost, Sawin and Tao [TS16] also found another family of tensors for which the upper and lower bound match. These tensors are studied for their connection to the capset problem where we deal with arithmetic progressions of length k, which are sequences of the form $a, a+b, a+2b, \ldots, a+(k-1)b$ for some a, b in a finite abelian group.

Proposition 4.2.8 ([TS16])**.** *Let T be a k-tensor for $k\geq 8$ where the sets of formal variables are some finite abelian group G. If supp(T) is contained in the set of all arithmetic progressions in T of length k and contains all constant sequences then $SR(T)=|G|$.*

In a bachelor thesis [Bor18], this statement was improved to $k\geq 7$. We improve this further to $k\geq 6$ with the use of the following lemma. Recall that a list of linear orders defines a partial order on the product of the sets, which we call the product order.

Lemma 4.2.9. *For all $n\in\mathbb{N}$ there is a list of 6 linear orders such that the constant sequences are maximal for the product order among all arithmetic sequences in $\mathbb{Z}/n\mathbb{Z}$.*

Proof. Take representatives $1,2,\ldots,n$ of the elements of $\mathbb{Z}/n\mathbb{Z}$. Let $\mathbf{c}$ be the constant sequence $(c,\ldots,c)$. Consider the following linear orders: $s:=1<2<\cdots<n$ and $f:=\lfloor\frac{n}{2}\rfloor<\lfloor\frac{n}{2}\rfloor-1<\cdots<1<n<n-1<\cdots<\lfloor\frac{n}{2}\rfloor+1$. We let $<_s$ be the symbol for the order of s, which agrees with the standard order on $\mathbb{Z}$ on its domain and $<_f$ for the order of f. Now, we may consider the following list $[s,f,s]$. We show that for any $c\geq\lfloor\frac{n}{2}\rfloor+1$, the sequence $\mathbf{c}$ is maximal under the partial order defined by this list. Fix such a c and suppose $\mathbf{c}$ is not maximal. Then there is an arithmetic sequence $a, a+b, a+2b, \ldots$ which is larger than $\mathbf{c}$. We can assume $a\in\{1,\ldots,n\}$ and $b\in\{1,\ldots,n\}$. Then we must have

$$\begin{aligned}c &\leq_s a\\ c &\leq_f a+b\\ c &\leq_s a+2b\end{aligned}$$

and one of these inequalities must be strict. Note that $\lfloor\frac{n}{2}\rfloor+1\leq c\leq n$ combined with $c\leq_f a+b$ implies that $\lfloor\frac{n}{2}\rfloor+1\leq_s a+b\leq_s c$. In the integers the inequality $c\leq_s a$ implies $c\leq a<a+b\leq 2n$. Combine these two inequalities to get $n+\lfloor\frac{n}{2}\rfloor+1\leq a+b\leq n+c$ in the integers.
We move on to analyse $a+2b$. As integers we know that

$$3n\geq a+2b=2(a+b)-a\geq 2(n+\lfloor\frac{n}{2}\rfloor+1)-n=n+2+2\lfloor\frac{n}{2}\rfloor\geq 2n+1.$$

The $c\leq_s a+2b$ condition says that $2n+c\leq a+2b\leq 3n$ in $\mathbb{Z}$. Combining this with $a+2b=2(a+b)-a\leq 2(n+c)-c=2n+c$ implies $a+2b=2n+c$ in $\mathbb{Z}$. This immediately implies that

$$b=(a+2b)-(a+b)\geq 2n+c-(n+c)=n.$$

As a result, b must be exactly n and $a=c$, so $a, a+b, a+2b$ is the constant c sequence, which is not strictly larger than $\mathbf{c}$. Hence, $\mathbf{c}$ is maximal.

The map $\phi: x\mapsto 1-x$ defines a bijection from $\mathbb{Z}/n\mathbb{Z}$ to itself. It is linear, so it also defines a bijection from arithmetic progressions to arithmetic progressions. This map defines two new orders on $\mathbb{Z}/n\mathbb{Z}$. These are $a<_{\phi(s)}b$ if and only if $\phi(a)<_s\phi(b)$ and a similarly defined $<_{\phi(f)}$. For a $c\leq_s\lfloor\frac{n}{2}\rfloor$ we have $\phi(c)\geq_{\phi(s)}\lfloor\frac{n}{2}\rfloor+1$, so the same analysis as above shows that any constant sequence $\mathbf{c}$ with $1\leq_s c\leq_s\lfloor\frac{n}{2}\rfloor$ is maximal under the list $[\phi(s),\phi(f),\phi(s)]$. Therefore any constant sequence is maximal under $[s,f,s,\phi(s),\phi(f),\phi(s)]$. ∎

We will quickly record how this implies the statement below. However, this implication is essentially the same as in [TS16].

Proposition 4.2.10. *Let T be a k-tensor with $k \geq 6$ on $G \times \cdots \times G$ where G is some finite abelian group. If supp(T) is contained in the set of all arithmetic progressions in T of length k and contains all constant sequences then $SR(T) = |G|$.*

Proof. For any such tensor T the bound $SR(T) \leq |G|$ is immediate by Lemma 3.4.10. The structure theorem for finite abelian groups says that $G \cong \prod_{i=1}^{m} \mathbb{Z}/n_i\mathbb{Z}$ for some $n_i, m \in \mathbb{N}$. For each n_i we have the orders $[s_{n_i}, f_{n_i}, s_{n_i}, \phi(s)_{n_i}, \phi(f)_{n_i}, \phi(s)_{n_i}]$. Taking the lexicographic order defined by $(s_{n_1}, \ldots, s_{n_m})$ gives an order s on G through the isomorphism. Similarly we also get orders $f, \phi(s), \phi(f)$ on G. For $k \geq 6$ we may use any orders on the last $k-6$ factors G and use $[s, f, s, \phi(s), \phi(f), \phi(s)]$ on the first six. Any arithmetic progression in G is also an arithmetic progression in each of the n_i. In the previous lemma we showed that all constant sequences in $\mathbb{Z}/n_i\mathbb{Z}$ are maximal. Thus each constant sequence in G is maximal per n_i and therefore maximal for $[s, f, s, \phi(s), \phi(f), \phi(s)]$. The extra orders do not matter, so all constant sequences are in max(supp(T)). Let $G_c \subseteq G^k$ be the set of constant sequences. Any vertex cover of max(supp(T)) is a vertex cover of G_c, so $\tau(\max(\text{supp}(T))) \geq \tau(G_c)$. In G_c each vertex is only in one edge, so $\tau(G_c) \geq |G_c| = |G|$. Lemma 4.2.2 shows that $SR(T) \geq \tau(\max(\text{supp}(T))) \geq |G|$. The upper and lower bound agree, so we get $SR(T) = |G|$. ☐

Remark 4.2.11. The question whether Proposition 4.2.10 is true remains open for $k = 4$ and $k = 5$, but for $k \leq 3$ it is not true for all appropiate tensors. The capset tensor gives a counterexample. For large powers $T^{\otimes n}$ of this tensor the slice rank is less than 3^n. The capset tensor was based on a function $\mathbb{F}_3 \times \mathbb{F}_3 \times F_3 \to F_3$ and thus we can also write it with bases $\{0, 1, 2\}$. Then the support of $T^{\otimes n}$ is the set of all arithmetic progressions of length 3 in $\mathbb{F}_3^n$. We can also show that Lemma 4.2.9 is not true when replacing the 6 by a four. This can be shown by considering $\mathbb{Z}/3\mathbb{Z}$. In this case all arithmetic progressions are of the form x, y, z, x for some $x, y, z \in \mathbb{Z}/3\mathbb{Z}$ and all such sequences with x, y, z distinct are arithmetic progressions. Therefore if x, x, x, x is maximal, then $x >_2 y$ or $x >_3 z$. Suppose without loss of generality that $0 <_2 1 <_2 2$, then we must have that $0 >_3 1$ and $0 >_3 2$. Now $1, 2, 0, 1$ is greater than $1, 1, 1, 1$. We conclude that not all constant sequences can be maximal.

4.3 Asymptotic vertex cover numbers

These slice rank bounds also translate to the asymptotic world. A lot of results in this section also appear in a different form in [CVZ21]. However, we avoid the explicit mention of the asymptotic spectrum. Furthermore, Proposition 4.3.5 is new. Firstly, we define the asymptotic vertex cover numbers.

Definition 4.3.1. For any k-partite k-uniform hypergraph H we define the asymptotic vertex cover number by

$$\tilde{\tau}(H) := \lim_{n\to\infty} \tau(H^{\times n})^{1/n}$$
$$\tilde{\tau}^*(H) := \lim_{n\to\infty} \tau^*(H^{\times n})^{1/n}$$

The latter limit exists by Fekete's lemma and the supermultiplicativity from Lemma 4.1.6 and the first limit is equal to it by Corollary 4.1.5.

Lemmas 4.2.1 and 4.2.2 imply similar bounds in the asymptotic case.

Corollary 4.3.2. *Let T be a k-tensor and let some product order be defined on the support then $\tilde{\tau}(\text{supp}(T)) \geq \widetilde{\text{SR}}(T) \geq \tilde{\tau}(\max(\text{supp}(T)))$.*

Proof. Note that for any natural n we have $\text{supp}(T^{\otimes n}) = \text{supp}(T)^{\times n}$. This means that for any $n \in \mathbb{N}$ we can use Lemma 4.2.1 to say $SR(T^{\otimes n})^{1/n} \leq \tau(\text{supp}(T^{\otimes n}))^{1/n} = \tau(\text{supp}(T)^{\times n})^{1/n}$ and thus taking the lim sup on both sides gives $\widetilde{\text{SR}}(T) \leq \tilde{\tau}(\text{supp}(T))$. The same argument using Lemma 4.2.2 shows $\tilde{\tau}(\max(\text{supp}(T))) \leq \widetilde{\text{SR}}(T)$. ☐

This corollary shows that $\widetilde{\mathrm{SR}}(T) = \tilde{\tau}(\mathrm{supp}(T))$ for tensors with oblique support. It would be extremely interesting if we could compute $\tilde{\tau}(H)$ for k-uniform, k-partite hypergraphs. This problem can be turned into a continuous optimisation problem. Let $\Phi \subseteq X_1 \times \cdots \times X_k$ be a set and recall that we write $\mathcal{P}(\Phi)$ for the set of all probability distribution on Φ. If $p \in \mathcal{P}(\Phi)$ we write p_{X_i} for the marginal distributions of p on X_i.

Proposition 4.3.3 ([TS16])**.** *Let $\Phi \subseteq X_1 \times \cdots \times X_k$ be some set. If $p \in \mathcal{P}(\Phi)$ then we have*

$$\log \tilde{\tau}(\Phi) = \sup_{p \in \mathcal{P}(\Phi)} \min_i H(p_{X_i}).$$

Proof. For ease of notation we write the proof for $k = 3$. For any $n \in \mathbb{N}$ we can pick a distribution $q(n)$ on Φ such that $q(n)$ is close to p and $q(n)$ takes values which are multiples of n. Here 'close to' means that $|q(n) - p| < 1/n$. Now consider the set $\Psi_{q(n)} \subseteq \Phi^n$ of all elements which are of type $q(n)$. This has $\tau(\Psi_{q(n)}) \leq \tau(\Phi^n)$, as it is a subset. Take a minimal vertex cover f of $\Psi_{q(n)}$. The vertices in one of X^n, Y^n, Z^n must cover at least $\frac{1}{3}$ of the edges, first we treat the case this is X. The degree of each element $x \in X^n$ is the number of elements $(y, z) \in Y^n \times Z^n$ such that (x, y, z) is of type $q(n)$. Let the permutation group S_n act on X^n by permuting $x_1, \dots, x_n$ in $x = (x_1, \dots, x_n)$. Each $x \in X^n \cap \Psi_{q(n)}$ must have type $q(n)_X$, so for any two $x = (x_1, \dots, x_n), x' = (x'_1, \dots, x'_n) \in X^n \subseteq \Psi_{q(n)}$ there is a $\sigma \in S_n$ which sends x to x'. Then let σ act on y and z for some $(x, y, z) \in \Psi_{q(n)}$. This gives y', z' and then (x', y', z') also has type $q(n)$. Using the existence of inverses we have now established a bijection between the edges x and x' are in. Thus each $x \in X^n \cap \Psi_{q(n)}$ has the same degree. The total degree of elements in $X^n \cap \Psi_{q(n)}$ is $|\Psi_{q(n)}|$, hence each element has degree $d = |\Psi_{q(n)}|/|X^n \cap \Psi_{q(n)}|$. In order to cover at least $\frac{1}{3}$ of the edges with vertices in X^n we need to use at least $\frac{1}{3}|\Psi_{q(n)}|/d = \frac{1}{3}|X^n \cap \Psi_{q(n)}|$ of the X-vertices. Lemma 3.7.4 says that this is $\frac{1}{3} \cdot 2^{(H(q(n)_X)+o(1))n}$. In the cases that the Y or Z vertices cover at least $\frac{1}{3}$ of the edges, we get the same expression but with $q(n)_Y$ and $q(n)_Z$ respectively. The minimum of these expressions bounds $\tau(\Psi_{q(n)})$ from below and in the limit we get

$$\begin{aligned}
\log \tilde{\tau}(\Phi) \geq \lim_{n\to\infty} \frac{1}{n} \log \tau(\Psi_{q(n)}) &\geq \lim_{n\to\infty} \min\{H(q(n)_X), H(q(n)_Y), H(q(n)_Z)\} + o(1) \\
&= \min\{H(p_X), H(p_Y), H(p_Z)\}.
\end{aligned}$$

For the upper bound we note that all elements in Φ^n are of some type. If $q(n)$ is any type then the subset $\Psi_{q(n)}$ of all edges of type $q(n)$ can be covered by $2^{(\min\{H(q(n)_X), H(q(n)_Y), H(q(n)_Z)\}+o(1))n}$ many vertices, because one of $X^n \cap \Psi_{q(n)}, Y^n \cap \Psi_{q(n)}, Z^n \cap \Psi_{q(n)}$ is of such size. In the limit this becomes

$$\begin{aligned}
\log \tilde{\tau}(\Phi) &\leq \lim_{n\to\infty} \frac{1}{n} \log \left(\sum_{\text{types } q(n)} \tilde{\tau}(\Psi_{q(n)}) \right) \\
&\leq \lim_{n\to\infty} \frac{1}{n} \log \left(\mathrm{poly}(n) \max_{\text{types } q(n)} \tilde{\tau}(\Psi_{q(n)}) \right) \\
&= \sup_{p \in \mathcal{P}(\Phi)} \min\{H(p_X), H(p_Y), H(p_Z)\}.
\end{aligned}$$

Combining the upper and lower bound shows the equality. □

The expression simplifies if we introduce symmetry.

Corollary 4.3.4. *Let $\Phi \subseteq X_1 \times \cdots \times X_k$ be symmetric, then*

$$\log \tilde{\tau}(\Phi) = \sup_{p \in \mathcal{P}_{sym}(\Phi)} H(p_{X_1}).$$

Proof. The expressions all stay the same under bijections of the X_i, so we may assume Φ is in symmetric form. For symmetric distributions all marginal distributions are the same, so $H(p_{X_i}) = H(p_{X_j})$ for all i, j. The minimum in Proposition 4.3.3 is obtained for each i, so

$$\log \tilde{\tau}(\Phi) = \sup_{p \in \mathcal{P}(\Phi)} H(p_{X_1}) \geq \sup_{p \in \mathcal{P}_{sym}(\Phi)} H(p_{X_1}).$$

Now we prove the reverse inequality. The entropy function is concave as stated in Theorem 3.7.6, which means that for any $p \in \mathcal{P}(\Phi)$, the symmetrised distribution q with

$$q(a_{i_1}, \ldots, a_{i_k}) := \frac{1}{k}(p(a_{i_1}, \ldots, a_{i_k}) + p(a_{i_2}, \ldots, a_{i_k}, a_{i_1}) + \ldots + p(a_{i_k}, \ldots, a_{i_{k-2}}, a_{i_{k-1}}))$$

has

$$\min_i H(q_{X_i}) = H(q_{X_1}) \geq \frac{1}{k}\sum_i H(p_{X_i}) \geq \min_i H(p_{X_i}).$$

This shows that $\sup_{p\in\mathcal{P}(\Phi)} \min_i H(p_{X_i}) \leq \sup_{p\in\mathcal{P}_{sym}(\Phi)} H(p_{X_1})$, which finishes the proof. $\square$

Using this expression it can be shown that $\tilde{\tau}$ behaves well with respect to the Kronecker product for symmetric sets.

Proposition 4.3.5. *If $\Phi \subseteq X_1 \times \cdots \times X_k$ and $\Psi \subseteq X'_1 \times \cdots \times X'_k$ are symmetric then*

$$\tilde{\tau}(\Phi)\tilde{\tau}(\Psi) = \tilde{\tau}(\Phi \times \Psi).$$

Proof. Again, we take $k = 3$ for ease of notation. Observe that $\Phi \times \Psi$ is also symmetric and the bijections that put Φ and Ψ into symmetric form also put $\Phi \times \Psi$ into symmetric form. The asymptotic vertex cover number does not change under bijections, so we can assume all sets are in symmetric form. For any $r \in \mathcal{P}_{sym}(\Phi \times \Psi)$ we can define $p \in \mathcal{P}_{sym}(\Phi)$ and $q \in \mathcal{P}_{sym}(\Psi)$ such that $H(r_X) \leq H(p_X) + H(q_X)$. Take

$$p(x, y, z) := \sum_{x',y',z'} r((x, x'), (y, y'), (z, z'))$$
$$q(x', y', z') := \sum_{x,y,z} r((x, x'), (y, y'), (z, z')).$$

It is easily checked that these are symmetric probability distributions. In this case r_X is a probability distribution on $X \times X'$ and has marginals p_X, $q_{X'}$. Lemma 3.7.5 implies that $H(r_X) \leq H(p_X) + H(q_X)$. Thus applying sup over all symmetric probability distributions first on the right hand side and then on the left hand side yields

$$\sup_{r\in\mathcal{P}_{sym}(\Phi\times\Psi)} H(r_X) \leq \sup_{p\in\mathcal{P}_{sym}(\Phi)} H(p_X) + \sup_{q\in\mathcal{P}_{sym}(\Psi)} H(q_X).$$

Thus $\tilde{\tau}$ is submultiplicative for symmetric sets. On the other hand, $\tau^\star$ is supermultiplicative, which means that

$$\tau^*(\Phi^{\times n} \times \Psi^{\times n}) \geq \tau^*(\Phi^{\times n})\tau^*(\Psi^{\times n}).$$

Taking the limit implies that $\tilde{\tau}^*$ is supermultiplicative as well. Hence $\tilde{\tau} = \tilde{\tau}^*$ is multiplicative. $\square$

4.4 Partitions and asymptotic vertex cover numbers

The asymptotic cover number of a hypergraph can now be found using optimisation software. The supremum of the entropies over all distributions is a concave optimisation problem. This is solvable in polynomial time, but for large sets, which means a high-dimensional $\mathcal{P}(\Phi)$, this still takes very long. For symmetric tensors we can reduce the dimension by restricting to symmetric distributions. In order to reduce the problem further, we try to group certain variables together and give the same probability mass to all variables in one group. If the groups are chosen appropiately, then the supremum does not change too much under this extra condition. We write this down formally for 3-tensors.

We can use the same terminology about partitions of tensors for partitions of edge sets. A triple of partitions $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ of X, Y and Z gives a partition $\{X_i \times Y_j \times Z_k : X_i \in \mathcal{X}, Y_j \in \mathcal{Y}, Z_k \in \mathcal{Z}\}$ of $X \times Y \times Z$. For $\Phi \subseteq X \times Y \times Z$ and some triple of partitions Λ the block support of Φ with respect to Λ is the set of block triples (i, j, k) such that $\Phi_{ijk} := (X_i \times Y_j \times Z_k) \cap \Phi \neq \emptyset$. Such a block support L is a subset of $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ and thus it can be regarded as the edge set of a 3-uniform 3-partite hypergraph.
Consider $p \in \mathcal{P}(L)$. This carries less information than an element of $\mathcal{P}(\Phi)$, but we would like to use it to say something about $\tilde{\tau}(\Phi)$. It turns out that the size of the blocks is enough information, thus we introduce weighted entropy.

Definition 4.4.1. For a set of sets $\mathcal{A}$ and a probability distribution p on $\mathcal{A}$ we define the *weighted entropy* as

$$H_{\mathcal{A}}(p) := -\sum_{A \in \mathcal{A}} p(A) \log \frac{p(A)}{|A|}.$$

For any triple of partitions $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ of X, Y, Z and $\Phi \subseteq X \times Y \times Z$ we are particularly interested in the expression

$$h_\Lambda(\Phi) := \sup_{p \in \mathcal{P}(L)} \min\{H_{\mathcal{X}}(p_X), H_{\mathcal{Y}}(p_Y), H_{\mathcal{Z}}(p_Z)\}.$$

Proposition 4.4.2. *Let $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ be a triple of partitions of X, Y, Z and let $\Lambda' = (\mathcal{X}', \mathcal{Y}', \mathcal{Z}')$ be a refinement of Λ in the sense that $\mathcal{X}'$ is a refinement of $\mathcal{X}$, $\mathcal{Y}'$ is a refinement of $\mathcal{Y}$ and $\mathcal{Z}'$ is a refinement of $\mathcal{Z}$. Consider a set $\Phi \subseteq X \times Y \times Z$. Then, $h_\Lambda(\Phi) \geq h_{\Lambda'}(\Phi)$.*

Proof. We show the statement if Λ' is the same as $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ except for $\mathcal{X}'$ in which one set in $\mathcal{X}$ has been split into two. This implies the statement for any refinement. By symmetry, the statement also holds for splitting the Y- or Z-variables. By repeating this splitting step, we can get any refinement and the inequality still holds. Let $X_1, \ldots, X_r$ form the partition of X in Λ. Suppose without loss of generality that X_1 gets split into X'_0 and X'_1 and all other sets remain the same for Λ'. Write L and L' for the block support of Φ with respect to Λ and Λ' respectively. Given $p' \in \mathcal{P}(L')$, we define $p \in \mathcal{P}(L)$ on any block $X_i \times Y_j \times Z_k$ via

$$p(X_i \times Y_j \times Z_k) = \begin{cases} p'(X'_0 \times Y_j \times Z_k) + p'(X'_1 \times Y_j \times Z_k) & \text{if } i = 1, \\ p'(X_i \times Y_j \times Z_k) & \text{otherwise.} \end{cases}$$

We assumed Y had the same partition in Λ and Λ' and for any $Y_j \in \mathcal{Y}$, we defined p such that

$$\begin{aligned} p(Y_j) &= \sum_{i=1}^{r} \sum_{Z_k \in \mathcal{Z}} p(X_i \times Y_j \times Z_k) \\ &= \sum_{Z_k \in \mathcal{Z}} \left(p'(X'_0 \times Y_j \times Z_k) + p'(X'_1 \times Y_j \times Z_k) + \sum_{i=2}^{r} p'(X_i \times Y_j \times Z_k) \right) = p'(Y_j). \end{aligned}$$

This shows that $H_{\mathcal{Y}}(p_Y) = H_{\mathcal{Y}}(p'_Y)$. Analogously it is true that $H_{\mathcal{Z}}(p_Z) = H_{\mathcal{Z}}(p'_Z)$. For any $i > 1$ we also get that $p(X_i) = p'(X_i)$, so the term $C := -\sum_{i=2}^{r} p(X_i) \log \frac{p(X_i)}{|X_i|}$ is the same in both $H_{\mathcal{X}}(p_X)$ and $H_{\mathcal{X}'}(p'_X)$.

Now apply the inequality of the weighted arithmetic and geometric means. In general this gives for weights $w_i \geq 0$, not all zero, and reals $a_i \geq 0$ that

$$\left(\frac{w_1 a_1 + \cdots + w_n a_n}{w_1 + \cdots + w_n} \right)^{w_1 + \cdots + w_n} \geq a_1^{w_1} \cdots a_n^{w_n}$$

with equality if and only if $a_i = a_j$ for all i, j with $w_i > 0$ and $w_j > 0$.
Applying this to $\frac{|X'_0|}{p'(X'_0)}$ and $\frac{|X'_1|}{p'(X'_1)}$ with weights $p'(X'_0)$ and $p'(X'_1)$ gives

$$\left(\frac{|X_1|}{p(X_1)} \right)^{p(X_1)} = \left(\frac{|X'_0| + |X'_1|}{p'(X'_0) + p'(X'_1)} \right)^{p'(X'_0) + p'(X'_1)} \geq \left(\frac{|X'_0|}{p'(X'_0)} \right)^{p'(X'_0)} \left(\frac{|X'_1|}{p'(X'_1)} \right)^{p'(X'_1)}$$

Taking the logarithm on both sides preserves the inequality and thus

$$H_{\mathcal{X}}(p_X) = C - p(X_1) \log \frac{p(X_1)}{|X_1|} \geq C - p'(X'_0) \log \frac{p'(X'_0)}{|X'_0|} - p'(X'_1) \log \frac{p'(X'_1)}{|X'_1|} = H_{\mathcal{X}'}(p'_X)$$

This shows that for any $p' \in \mathcal{P}(L')$, there is a $p \in \mathcal{P}(L)$ such that

$$\min\{H_{\mathcal{X}'}(p'_X), H_{\mathcal{Y}'}(p'_Y), H_{\mathcal{Z}'}(p'_Z)\} \leq \min\{H_{\mathcal{X}}(p_X), H_{\mathcal{Y}}(p_Y), H_{\mathcal{Z}}(p_Z)\}.$$

This inequality must also be true for the supremum and therefore $h_{\Lambda'}(\Phi) \leq h_\Lambda(\Phi)$. □

In the particular case that each partition in Λ' consists of singletons we have $h_{\Lambda'}(\Phi) = \log \tilde{\tau}(\Phi)$. This singleton partition is a refinement for any triple of partitions Λ, so $\log \tilde{\tau}(\Phi) \le h_\Lambda(\Phi)$. Combining this with Lemma 4.3.2 shows Theorem 4.4 from [Alm18]. In this paper Alman also showed that equality is obtained for laser-ready tensors. Based on our proof we identify another type of partition for which $\log \tilde{\tau}(\Phi) = h_\Lambda(\Phi)$.

Definition 4.4.3. A set $\Phi \subseteq X \times Y \times Z$ is *regular* if there is a weight function $w : \Phi \to [0,1]$ such that

$$\sum_{y\in Y, z\in Z} w(x,y,z) = \frac{1}{|X|} \quad \forall x \in X$$
$$\sum_{x\in X, z\in Z} w(x,y,z) = \frac{1}{|Y|} \quad \forall y \in Y$$
$$\sum_{y\in Y, x\in X} w(x,y,z) = \frac{1}{|Z|} \quad \forall z \in Z.$$

An example of a regular set is the support of a matrix multiplication tensor. In the support of $\langle a,b,c\rangle$ each X-variable x_{ij} appears exactly in the c terms $x_{ij}y_{jk}z_{ki}$ where k varies. Similarly each Y-variable appears in a terms and each Z-variable appears in b terms. Thus $w(x,y,z) = \frac{1}{abc}$ gives an appropiate weight function.

Proposition 4.4.4. *Let $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ be some triple of partitions for $\Phi \subseteq X \times Y \times Z$. If Φ_{ijk} is a regular set for each block triple (i,j,k) in the block support then $h_\Lambda(\Phi) = \log \tilde{\tau}(\Phi)$.*

Proof. We have already seen that $h_\Lambda(\Phi) \ge \log \tilde{\tau}(\Phi)$. Suppose all Φ_{ijk} are regular, so to each block triple B in the block support L we can associate a weight function w_B. Take any $p \in \mathcal{P}(\Phi)$. For any xyz in Φ there is a unique block triple $B = X_i \times Y_j \times Z_k$ in which it lies. Then define $q(xyz) := p(B)w_B(x,y,z)$. This defines a probability distribution q on Φ, as $\sum_{xyz\in B} w_B(x,y,z) = 1$ for all block triples B. Each block triple is regular, so for all $x \in X_i$ we have

$$\begin{aligned} q(x) &= \sum_{j,k} \sum_{\substack{y,z: \\ xyz\in X_i\times Y_j\times Z_k}} q(xyz) \\ &= \sum_{j,k} p(X_i \times Y_j \times Z_k) \sum_{\substack{y,z: \\ xyz\in X_i\times Y_j\times Z_k}} w_{X_i\times Y_j\times Z_k}(x,y,z) \\ &= \sum_{j,k} \frac{p(X_i \times Y_j \times Z_k)}{|X_i|} = \frac{p(X_i)}{|X_i|}. \end{aligned}$$

We can now compute the entropy

$$\begin{aligned} H(q_X) &= -\sum_{x\in X} q(x)\log q(x) \\ &= -\sum_i \sum_{x\in X_i} \frac{p(X_i)}{|X_i|} \log \frac{p(X_i)}{|X_i|} \\ &= -\sum_i p(X_i) \log \frac{p(X_i)}{|X_i|} = H_{\mathcal{X}}(p_X). \end{aligned}$$

Similarly we also get $H(q_Y) = H_{\mathcal{Y}}(p_Y)$ and $H(q_Z) = H_{\mathcal{Z}}(p_Z)$. The distribution q on Φ is such that

$$\min\{H_{\mathcal{X}}(p_X), H_{\mathcal{Y}}(p_Y), H_{\mathcal{Z}}(p_Z)\} = \min\{H(q_X), H(q_Y), H(q_Z)\} \le \log \tilde{\tau}(\Phi).$$

This can be done for any $p \in \mathcal{P}(\Phi)$, so the inequality holds for the supremum and hence $h_\Lambda(\Phi) \le \log \tilde{\tau}(\Phi)$. It follows that $h_\Lambda(\Phi) = \log \tilde{\tau}(\Phi)$.

□

We can combine this result with Corollary 4.3.2.

Corollary 4.4.5. *For any tensor T with oblique support and any triple of partitions Λ for which all partition subtensors are regular, we have* $\log \widetilde{\mathrm{SR}}(T) = h_\Lambda(\mathrm{supp}(T))$.

In order to find $h_\Lambda(\mathrm{supp}(T))$, one just needs to consider all probability distributions on the block support. This is an optimisation problem that has much fewer degrees of freedom and can therefore be solved for a larger class of tensors. In [Alm18] this same equality was shown for tensors with laser-ready partitions. These have tight and thus oblique block support. Each block is isomorphic to a matrix multiplication tensor and is therefore both regular and oblique. This does not imply that each tensor with laser-ready partitions is oblique. We say that the isomorphisms are *consistent* if the variables in each block can be named in such a way that each block triple is exactly $\sum_{i=1}\sum_{j=1}\sum_{k=1} x_{ij}y_{jk}z_{ki}$. If the isomorphisms are consistent, then the laser-ready tensor is oblique and we can deduce that $\log \widetilde{\mathrm{SR}}(T) = h_\Lambda(\mathrm{supp}(T))$ from Corollary 4.4.5.

Remark 4.4.6. We can ask ourselves for which tensors T with oblique support and triple of partitions Λ we get $\log \widetilde{\mathrm{SR}}(T) = h_\Lambda(\mathrm{supp}(T))$. If we can guarantee that each step in the proof of Proposition 4.4.2 must be an equality, then we need each partition subtensor to be regular. This is the case in the proposition below. In general this does not need to be true. Some block triples can get $p(B) = 0$ in the maximising distribution, which either means that there is no restriction on $\mathrm{supp}(T)_B$, or that we need not have that a block is regular in all variables. For example, if the minimum of $\min(H_\mathcal{X}(p_X), H_\mathcal{Y}(p_Y), H_\mathcal{Z}(p_Z))$ is not obtained at the Y term, then we do not require regularity in the Y-variables.

Proposition 4.4.7. *Any symmetric tensor T with oblique support has maximal asymptotic slice rank if and only if* $\mathrm{supp}(T)$ *is regular.*

Proof. For symmetric tensors with oblique support we know that $\log \widetilde{\mathrm{SR}}(T) = \log \tilde{\tau}(\mathrm{supp}(T))$ and that $\log \tilde{\tau}(\mathrm{supp}(T)) = \sup_{p\in\mathcal{P}_{sym}(\mathrm{supp}(T))} H(p_X)$. Now consider the triple of partitions $\Lambda = (\{X\}, \{Y\}, \{Z\})$. In this case there is only one distribution and $h_\Lambda(\mathrm{supp}(T)) = \log |X|$ by symmetry. If $\mathrm{supp}(T)$ is regular, then $\sup_{p\in\mathcal{P}_{sym}(\Phi)} H(p_X) = h_\Lambda(\mathrm{supp}(T))$, so the asymptotic slice rank is equal to the dimension bound, so it is maximal. If the asymptotic slice rank is maximal, then $\sup_{p\in\mathcal{P}_{sym}(\mathrm{supp}(T))} H(p_X) = \log |X|$ must hold. We know that for a probability distribution on X the entropy only gets maximised for a uniform distribution. If q is the uniform distribution, then $H(q) = \log |X|$. Thus p_X must be the uniform distribution. By symmetry the same must hold for Y and Z and therefore p is a weight function, which shows that $\mathrm{supp}(T)$ is regular. □

5 Multiplicativity of the asymptotic slice rank

In the previous chapter we compared the slice rank and the vertex cover number of the support. We know the slice rank is the same for equivalent tensors. Thus we can compare the slice rank of T to the vertex cover number of the support of all equivalent tensors. For a tensor T we let G_T be the group of all equivalences on T. If T is a tensor on $X_1, \ldots, X_k$ over $\mathbb{F}$ this is the group $GL(X_1, \mathbb{F}) \times \cdots \times GL(X_k, \mathbb{F})$.

Theorem 5.0.1. *Let T be a tensor. Then $SR(T) = \min_{g\in G_T} \tau(\mathrm{supp}(g \cdot T))$.*

Proof. We give a sketch of the proof of this fact. Let X_i be a basis of V_i for each i and $T \in V_1 \otimes \cdots \otimes V_k$. For any $g \in G$ we have $SR(T) = SR(g \cdot T) \leq \tau(\mathrm{supp}(g \cdot T))$, so $SR(T) \geq \min_{g\in G_T} \tau(\mathrm{supp}(g \cdot T))$. For the reverse inequality we note the following. Fix a slice rank decomposition for T. Each i-slice is an element of $V_1 \otimes \cdots \otimes \{v_i\} \otimes \cdots \otimes V_k$. The different i-slice vectors v_i for a fixed i are linearly independent by minimality of the slice rank decomposition. Therefore there is a basis of each V_i which contains all the i-slice vectors. Writing T in this basis corresponds to picking a certain $g \in G_T$. The support in this basis can be covered by all the slice vectors v_i. There are $SR(T)$ many of these, so there is a $g \in G_T$ such that $\tau(\mathrm{supp}(g \cdot T)) \leq SR(T)$. □

In general it is hard to compute the slice rank from this expression as there are potentially infinitely many elements in G_T. If T has oblique support, then the minimum is obtained at $g = \mathbf{1}_{G_T}$. This extra information can still be used and therefore it is easier to consider tensors with oblique support.

Definition 5.0.2. A tensor T is *oblique* if it is equivalent to a tensor with oblique support.

5.1 The G-stable rank

This characterisation of the slice rank as the minimum over vertex cover numbers leads to the question if we can do the same with the fractional vertex cover number.

Definition 5.1.1. The *G-stable rank* $rk^G(T)$ of a tensor T is $\min_{g\in G_T} \tau^*(\mathrm{supp}(g\cdot T))$.

This is a valid notion of rank in the sense that it is subadditive for sums, decreasing under restriction and satisfies $rk^G(\langle n\rangle_k) = n$ for $k \geq 2$. It has been studied by Derksen [Der22]. In his paper he adopted a different definition, but also showed that our definition and his are equivalent. We may hope that this rank notion inherits the supermultiplicativity of τ^*. It is not known whether this is generally true. However, surprisingly it has been shown to hold for tensors over $\mathbb{C}$.

Lemma 5.1.2 ([Der22]). *Let S and T be two k-tensors over $\mathbb{C}$ then $rk^G(S\otimes T) \geq rk^G(S)rk^G(T)$.*

Lemma 4.1.3 implies that the notion of slice rank and G-stable rank should be close together.

Corollary 5.1.3. *For any k-tensor T we have $\frac{2}{k}SR(T) \leq rk^G(T) \leq SR(T)$.*

From these two statements we deduce the following.

Corollary 5.1.4. *Let $k \geq 2$. The inequality*

$$\frac{4}{k^2}SR(S)SR(T) \leq rk^G(S)rk^G(T) \leq rk^G(S\otimes T) \leq SR(S\otimes T)$$

holds for any k-tensors S and T over $\mathbb{C}$.

5.2 Supermultiplicativity for oblique tensors and complex tensors

In Chapter 2 it was established that the slice rank is neither sub- nor supermultiplicative. This implied that the asymptotic slice rank had to be defined as a limsup. Christandl, Vrana and Zuiddam found two cases in which the limit exists [CVZ21]; through a connection between asymptotic slice rank and the general framework of the asymptotic spectrum they showed that this limit exists for oblique tensors and for tensors over $\mathbb{C}$. This connection also immediately implied that asymptotic slice rank is supermultiplicative and superadditive in those cases. In this section we record an alternative proof of these facts. Specifically the following lemma is new.

Lemma 5.2.1. *Let S and T be oblique k-tensors with $k \geq 2$. Then*

$$\frac{8}{k^3}SR(S)SR(T) \leq SR(S\otimes T).$$

Proof. There are invertible restrictions g_s, g_t such that $g_s\cdot S$ and $g_t\cdot T$ have oblique support. The restriction $(g_s\otimes g_t)\cdot(S\otimes T) = (g_s\cdot S)\otimes(g_T\cdot T)$ is equivalent to $S\otimes T$ and has oblique support by Lemma 4.2.5. Slice rank and G-stable rank are constant on equivalent tensors, so we may assume that S and T have oblique support. In this case $SR(S\otimes T) = \tau(\mathrm{supp}(S\otimes T))$, so Corollary 5.1.3 implies that

$$\frac{2}{k}\tau(\mathrm{supp}(S\otimes T)) \leq rk^G(S\otimes T) \leq SR(S\otimes T). \tag{5.1}$$

Additionally, we know that $\mathrm{supp}(S\otimes T) = \mathrm{supp}(S)\times\mathrm{supp}(T)$, so

$$\tau^*(\mathrm{supp}(S))\tau^*(\mathrm{supp}(T)) \leq \tau^*(\mathrm{supp}(S)\times\mathrm{supp}(T)) \leq \tau(\mathrm{supp}(S)\times\mathrm{supp}(T))$$

by Lemmas 4.1.6 and 4.1.3. The G-stable rank is defined as a minimum, so we get $rk^G(T) \leq \tau^*(\mathrm{supp}(T))$ for all tensors T. Lemma 5.1.3 can now be used to get the string of inequalities

$$\frac{4}{k^2}SR(S)SR(T) \leq rk^G(S)rk^G(T) \leq \tau^*(\mathrm{supp}(S))\tau^*(\mathrm{supp}(T)) \leq \tau(\mathrm{supp}(S)\times\mathrm{supp}(T)). \tag{5.2}$$

From Inequalities 5.1 and 5.2 we deduce the final inequality

$$\frac{8}{k^3}SR(S)SR(T) \leq \frac{2}{k}\tau(\mathrm{supp}(S)\times\mathrm{supp}(T)) \leq rk^G(S\otimes T) \leq SR(S\otimes T).$$

□

This lemma and Corollary 5.1.4 show that the slice rank is supermultiplicative up to a constant for the family of oblique tensors and for the family of tensors over $\mathbb{C}$. We will now show this is enough to give supermultiplicativity and superadditivity for the asymptotic slice rank.

Proposition 5.2.2. *Let $\mathcal{T}$ be a family of k-tensors which is closed under the Kronecker product and addition and for there is a constant $c > 0$ such that any $S, T \in \mathcal{T}$ satisfy $SR(S \otimes T) \geq c \cdot SR(S)SR(T)$. Then the asymptotic slice rank is supermultiplicative for tensors in $\mathcal{T}$.*

Proof. First of all note that in this case $f(T) = c \cdot SR(T)$ is supermultiplicative on $\mathcal{T}$. If $T \in \mathcal{T}$, then $T^{\otimes n} \in \mathcal{T}$ for all $n \geq 1$, so the limit of $f(T^{\otimes n})^{1/n}$ exists by Fekete's lemma. The asymptotic slice rank is equal to this limit and it follows that $SR(T^{\otimes n})^{1/n}$ also converges. Let S, T be two tensors in $\mathcal{T}$. We know that $(T \otimes S)^{\otimes n} \cong T^{\otimes n} \otimes S^{\otimes n}$ and $SR(T^{\otimes n} \otimes S^{\otimes n}) \geq c \cdot SR(T^{\otimes n})SR(S^{\otimes n})$, hence

$$\begin{aligned}\widetilde{\mathrm{SR}}(T \otimes S) &= \lim_{n\to\infty} SR((T \otimes S)^{\otimes n})^{\frac{1}{n}} \\ &\geq \lim_{n\to\infty} \left(c \cdot SR(T^{\otimes n})SR(S^{\otimes n})\right)^{\frac{1}{n}} = \lim_{n\to\infty} (SR(T^{\otimes n})SR(S^{\otimes n}))^{\frac{1}{n}} = \widetilde{\mathrm{SR}}(T)\widetilde{\mathrm{SR}}(S).\end{aligned}$$

□

Remark 5.2.3. By induction we also get that $\left(\sqrt{c} \cdot SR(T)\right)^n \leq SR(T^{\otimes n})$ for every positive integer n. This shows that $\widetilde{\mathrm{SR}}(T) \geq \sqrt{c} \cdot SR(T)$ for any tensor in $\mathcal{T}$.

Asymptotic rank notions that are supermultiplicative are often also superadditive, as can for example be seen in [Sch22]. Asymptotic slice rank follows this rule of thumb.

Proposition 5.2.4. *If $\mathcal{T}$ is a family as in the previous proposition, then asymptotic slice rank is superadditive under direct sums for tensors in $\mathcal{T}$.*

Proof. We show that the asymptotic slice rank is superadditive along the same lines as it was shown for the Shannon capacity in [Sch22]. Let S and T be any two k-tensors in $\mathcal{T}$. For any $n \in \mathbb{N}$, we can write out $(S \oplus T)^{\otimes n}$ as a direct sum. A lot of the tensors in this sum will be isomorphic according to Lemma 3.3.3 and if we group them accoding to isomorphism class, we get

$$(S \oplus T)^{\otimes n} \cong \bigoplus_{i=1}^{n} \bigoplus_{j=1}^{\binom{n}{i}} (S^{\otimes i} \otimes T^{\otimes n-i}).$$

Then the additivity from Lemma 3.4.4 and the fact that SR has the same value on equivalent tensors gives the following equality

$$SR((S \oplus T)^{\otimes n}) = \sum_{i=0}^{n} \binom{n}{i} SR(S^{\otimes i} \otimes T^{\otimes n-i}).$$

For any $t, m, i \geq 1$ and any tensor T we note that $SR(T^{\otimes tm+i}) \geq SR(T^{\otimes mt})$ by Corollary 3.4.9, and thus $SR(T^{\otimes tm+i}) \geq \left(\sqrt{c}SR(T^{\otimes t})\right)^m$. We use this inequality, with $m = \lfloor \frac{j}{t} \rfloor$ for some j, in the following calculation. Fix $t \geq 1$. In case S, T are tensors in $\mathcal{T}$, we have

$$\begin{aligned}SR((S \oplus T)^{\otimes n}) &\geq \sum_{i=0}^{n} \binom{n}{i} c \cdot SR(S^{\otimes i})SR(T^{\otimes n-i}) \\ &\geq \sum_{i=0}^{n} \binom{n}{i} c \left(\sqrt{c}\right)^{\lfloor \frac{i}{t} \rfloor + \lfloor \frac{n-i}{t} \rfloor} SR(S^{\otimes t})^{\lfloor \frac{i}{t} \rfloor} SR(T^{\otimes t})^{\lfloor \frac{n-i}{t} \rfloor} \\ &\geq c \left(\sqrt{c}\right)^{n/t} \sum_{i=0}^{n} \binom{n}{i} SR(S^{\otimes t})^{\frac{i}{t}-1} SR(T^{\otimes t})^{\frac{n-i}{t}-1} \\ &= c^{(n+2t)/(2t)} \left(SR(S^{\otimes t})^{1/t} + SR(T^{\otimes t})^{1/t}\right)^n SR(S^{\otimes t})^{-1} SR(T^{\otimes t})^{-1}.\end{aligned}$$

This means that for any $t \geq 1$

$$\begin{aligned}\widetilde{\mathrm{SR}}(S \oplus T) &= \limsup_{n\to\infty} SR((S \oplus T)^{\otimes n})^{1/n} \\ &\geq \lim_{n\to\infty} \left(SR(S^{\otimes t})^{1/t} + SR(T^{\otimes t})^{1/t}\right) SR(S^{\otimes t})^{-1/n} SR(T^{\otimes t})^{-1/n} \cdot c^{(n+2t)/2nt} \\ &= \left(SR(S^{\otimes t})^{1/t} + SR(T^{\otimes t})^{1/t}\right) c^{1/2t}.\end{aligned}$$

Letting $t \to \infty$ in this expression shows that $\widetilde{\mathrm{SR}}(S \oplus T) \geq \widetilde{\mathrm{SR}}(S) + \widetilde{\mathrm{SR}}(T)$.

□

Corollary 5.2.5. *Asymptotic slice rank is supermultiplicative and superadditive for oblique tensors and for tensors over* $\mathbb{C}$.

Proof. Corollary 5.1.4 and Lemma 5.2.1 show that the family of oblique tensors and the family of tensors over $\mathbb{C}$ satisfy the conditions of Propositions 5.2.2 and 5.2.4. The propositions then imply that the asymptotic slice rank is supermultiplicative and superadditive for these families of tensors. □

It is not known whether supermultiplicativity holds for all pairs of tensors over fields other than $\mathbb{C}$. We can however show that submultiplicativity cannot hold for all tensors.

Example 5.2.6. Let n be some positive integer. We know that

$$SR(\langle a, b, c\rangle^{\otimes n}) = SR(\langle a^n, b^n, c^n\rangle) = \min\{a^n b^n, b^n c^n, c^n a^n\} = SR(\langle a, b, c\rangle)^n.$$

This means that $\widetilde{\mathrm{SR}}(\langle a, b, c\rangle) = SR(\langle a, b, c\rangle)$. Therefore, the asymptotic slice rank is strictly supermultiplicative for the same tensors $\langle 1, 1, k\rangle$ and $\langle 1, k, 1\rangle$ as used in Example 3.4.6. This strict supermultiplicativity must imply strict superadditivity, see [Sch22]. We can also show this explicitly. The tensor product is commutative up to isomorphism, so we get that

$$(\langle 1, 1, k\rangle \oplus \langle 1, k, 1\rangle)^{\otimes n} \cong \bigoplus_{i=0}^{n} \bigoplus_{j=1}^{\binom{n}{i}} \langle 1, k^i, k^{n-i}\rangle.$$

If we apply the slice rank on both sides, we see that

$$SR((\langle 1, 1, k\rangle \oplus \langle 1, k, 1\rangle)^{\otimes n}) = \sum_{i=0}^{n} \binom{n}{i} SR(\langle 1, k^i, k^{n-i}\rangle) = \sum_{i=0}^{n} \binom{n}{i} k^{\min\{i, n-i\}} \geq 2 \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n}{i} k^i.$$

This last sum can be bounded from below by $2(1+k)^{\lfloor \frac{n-1}{2} \rfloor}$. This shows for $k > 3$ that

$$\widetilde{\mathrm{SR}}(\langle 1, 1, k\rangle \oplus \langle 1, k, 1\rangle) \geq \sqrt{1+k} > 2 = \widetilde{\mathrm{SR}}(\langle 1, 1, k\rangle) + \widetilde{\mathrm{SR}}(\langle 1, k, 1\rangle).$$

It may be noted that the tensors used in this example are highly asymmetric. We have not found a symmetric example that shows strict supermultiplicativity.

Conjecture The asymptotic slice rank is submultiplicative for symmetric tensors.

5.3 The multiplicativity of asymptotic slice rank for symmetric and oblique tensors

In this section we prove the above conjecture in the case of oblique tensors.

Theorem 5.3.1. *Let S and T be two symmetric and oblique k-tensors. Then* $\widetilde{\mathrm{SR}}(S \otimes T) = \widetilde{\mathrm{SR}}(S)\widetilde{\mathrm{SR}}(T)$ *and* $\widetilde{\mathrm{SR}}(S \oplus T) = \widetilde{\mathrm{SR}}(S) + \widetilde{\mathrm{SR}}(T)$.

First, we need to be able to get a tensor which has symmetric and oblique support at the same time.

Lemma 5.3.2. *If a k-tensor T is symmetric and oblique, then $T^{\otimes k}$ is equivalent to a symmetric tensor with oblique support.*

Proof. The tensor T is symmetric, so it is equivalent to a tensor with support in symmetric form, say S. Because S is equivalent to T, it is also oblique, so there are invertible matrices $(A_1, \ldots, A_k)$ for which $U = (A_1 \otimes \cdots \otimes A_k) \cdot S$ is a tensor with oblique support. For any cycle $\sigma \in S_k$ the restriction $(A_2, \ldots, A_k, A_1) \cdot S$ is well-defined, because S is in symmetric form. The restriction is equal to $\mathrm{rot}(U)$. The restriction can be rotated even further. The product of these rotated restrictions

$$(A_1 \otimes \cdots \otimes A_k) \otimes (A_2 \otimes \cdots \otimes A_k \otimes A_1) \otimes \cdots \otimes (A_k \otimes A_1 \otimes \cdots \otimes A_{k-1})$$

defines a restriction from $S^{\otimes k}$ to the tensor product $S' = U \otimes \mathrm{rot}(U) \otimes \cdots \otimes \mathrm{rot}^{k-1}(U)$, which is symmetric. Rotating the order of the linear orders for $\mathrm{supp}(U)$ shows that $\mathrm{supp}(\mathrm{rot}^i(U))$ is oblique. Hence S' is a product of tensors with oblique support and so it has oblique support by Lemma 4.2.5. Additionally, all A_i were invertible, so $A_1 \otimes \cdots \otimes A_k$ is invertible. This means that $S^{\otimes k}$ is equivalent to S'. Therefore $T^{\otimes k}$ is also equivalent to S', a symmetric tensor with oblique support. □

From this lemma and Lemma 4.3.5 the theorem follows rather quickly.

Proof of Theorem 5.3.1. The previous lemma implies that there are S' and T' which are symmetric, have oblique support and are equivalent to $S^{\otimes k}$ and $T^{\otimes k}$ respectively. As S', T' and $S' \otimes T'$ have oblique support, we can establish $\widetilde{\mathrm{SR}}$ by looking at $\tilde{\tau}$ of the support. The support is symmetric, so Proposition 4.3.5 implies

$$\widetilde{\mathrm{SR}}(S')\widetilde{\mathrm{SR}}(T') = \tilde{\tau}(\mathrm{supp}(S'))\tilde{\tau}(\mathrm{supp}(T')) = \tilde{\tau}(\mathrm{supp}(S' \otimes T')) = \widetilde{\mathrm{SR}}(S' \otimes T').$$

The fact that $S' \otimes T'$ is equivalent to $S^{\otimes k} \otimes T^{\otimes k}$, which is isomorphic to $(S \otimes T)^{\otimes k}$, means that

$$\widetilde{\mathrm{SR}}(S)^k \widetilde{\mathrm{SR}}(T)^k = \widetilde{\mathrm{SR}}(S^{\otimes k})\widetilde{\mathrm{SR}}(T^{\otimes k}) = \widetilde{\mathrm{SR}}((S \otimes T)^{\otimes k}) = \widetilde{\mathrm{SR}}(S \otimes T)^k.$$

We conclude that asymptotic slice rank is indeed multiplicative in this case.

From multiplicativity we can deduce additivity. Take oblique and symmetric k-tensors S and T and any positive integer n. We know from Proposition 5.2.4 that $\widetilde{\mathrm{SR}}(S \oplus T) \geq \widetilde{\mathrm{SR}}(S) + \widetilde{\mathrm{SR}}(T)$, because $SR(S \otimes T) \geq \frac{8}{k^3} SR(S)SR(T)$. For subadditivity we use Remark 5.2.3 to get

$$\begin{aligned} SR((S+T)^{\otimes n}) &= \sum_{i=1}^{n} \binom{n}{i} SR(S^{\otimes i} \otimes T^{\otimes n-i}) \\ &\leq \sum_{i=1}^{n} \binom{n}{i} \sqrt{\frac{k^3}{8}} \widetilde{\mathrm{SR}}(S^{\otimes i} \otimes T^{\otimes n-i}) \\ &= \sum_{i=1}^{n} \binom{n}{i} \tfrac{k3/2}{4} \widetilde{\mathrm{SR}}(S)^i \widetilde{\mathrm{SR}}(T)^{n-i} = \tfrac{k^{3/2}}{4} (\widetilde{\mathrm{SR}}(S) + \widetilde{\mathrm{SR}}(T))^n. \end{aligned}$$

If we raise this to the power $\frac{1}{n}$ and let $n \to \infty$, we get $\widetilde{\mathrm{SR}}(S \oplus T) = \widetilde{\mathrm{SR}}(S) + \widetilde{\mathrm{SR}}(T)$. □

6 The laser method

We now properly describe the laser method as developed by Coppersmith and Winograd. Recall the definition of the value and the generalised Schönhage's tau theorem.

Definition 3.8.2. The *value of a 3-tensor* T is defined as

$$V_\rho(T) := \sup_{n,\ \text{degenerations}\ \trianglerighteq} \left\{ \left(\sum_i (a_i b_i c_i)^\rho \right)^{1/n} \mid T^{\otimes n} \trianglerighteq \bigoplus_i \langle a_i, b_i, c_i \rangle \right\}.$$

Theorem 3.8.4 ([DS13]). *Let T be a 3-tensor. Then $V_{\omega/3}(T) \leq \tilde{R}(T)$.*

A tensor T can be used to bound ω if we can lower bound the value and upper bound the asymptotic slice rank. In the next section it is shown that choosing a tensor with laser-ready partition allows us to a lower bound for the value. These laser-ready tensors are concise by Lemma 3.8.10 and therefore the asymptotic rank is bounded below by the dimension. Among the laser-ready tensors, two families were identified for which the border rank is close to this dimension minimum. The asymptotic rank must then also be close to minimal by Lemma 3.6.4. These families are the following two.

For an integer $q \geq 1$, the small Coppersmith-Winograd tensor cw_q is the tensor

$$\sum_{i=1}^{q} x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0.$$

If we partition the variable sets as $X_0 = \{x_0\}, X_1 = \{x_1, \ldots, x_q\}, Y_0 = \{y_0\}, Y_1 = \{y_1, \ldots, y_q\}$ and $Z_0 = \{z_0\}, Z_1 = \{z_1, \ldots, z_q\}$, then the block support becomes $\{(0,1,1),(1,0,1),(1,1,0)\}$, which is a tight set. Each of the partition subtensors is a rotation of the matrix multiplication tensor $\langle 1, 1, q\rangle$.

The large Coppersmith-Winograd tensor CW_q is also defined for every $q \geq 1$ and is the same except for some extra corner terms:

$$CW_q := x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + z_{q+1} y_0 z_0 + \sum_{i=1}^{q} x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0.$$

Using the same partitions as for cw_q and putting $x_{q+1}, y_{q+1}, z_{q+1}$ each in its own class gives the block support

$$\{(0,1,1),(1,0,1),(1,1,0),(0,0,2),(0,2,0),(2,0,0)\}$$

which is once again a tight set. Each of the additional block triples contains a single term, so it will be isomorphic to $\langle 1,1,1\rangle$.

These tensors have the following border rank decompositions

$$\begin{aligned}\varepsilon^3 cw_q + O(\varepsilon^4) = \varepsilon &\sum_{i=1}^{q} (x_0 + \varepsilon x_i)(y_0 + \varepsilon y_i)(z_0 + \varepsilon z_i) \\ &- \left(x_0 + \varepsilon^2 \sum_{i=1}^{q} x_i\right)\left(y_0 + \varepsilon^2 \sum_{i=1}^{q} y_i\right)\left(z_0 + \varepsilon^2 \sum_{i=1}^{q} z_i\right) \\ &+ (1 - q\varepsilon) x_0 y_0 z_0 = \varepsilon^3 cw_q + O(\varepsilon^4)\end{aligned}$$

and

$$\begin{aligned}\varepsilon^3 CW_q + O(\varepsilon^4) = \varepsilon &\sum_{i=1}^{q} (x_0 + \varepsilon x_i)(y_0 + \varepsilon y_i)(z_0 + \varepsilon z_i) \\ &- \left(x_0 + \varepsilon^2 \sum_{i=1}^{q} x_i\right)\left(y_0 + \varepsilon^2 \sum_{i=1}^{q} y_i\right)\left(z_0 + \varepsilon^2 \sum_{i=1}^{q} z_i\right) \\ &+ (1 - q\varepsilon)(x_0 + \varepsilon^3 x_{q+1})(y_0 + \varepsilon^3 y_{q+1})(z_0 + \varepsilon^3 z_{q+1}).\end{aligned}$$

These decompositions show that $\underline{R}(cw_q) \leq q + 2$ and $\underline{R}(CW_q) \leq q + 2$, whereas conciseness gives the lower bound $\tilde{R}(cw_q) \geq q + 1$ and $\tilde{R}(CW_q) \geq q + 2$.

6.1 Lower bounding the value

Definition 6.1.1. For any probability distribution $p \in \mathcal{P}(\Phi)$ where $\Phi \subseteq X \times Y \times Z$ we let $\mathrm{Marg}(p)$ be the set of probability distributions on Φ which have the same marginals as p. We let $\Gamma(p) := \max_{q \in \mathrm{Marg}(p)} H(q) - H(p)$.

The lower bound for the value that arises from the laser method is provided by the following theorem.

Theorem 6.1.2 ([AW20])**.** *Let T be a tensor with block support L. For any $p \in \mathcal{P}(L)$ we have*

$$\log V_\rho(T) \geq \min\{H(p_X) + H(p_Y) + H(p_Z)\} + \sum_{B \in L} p(B) \log V_\rho(T_B) - \frac{1}{2}\Gamma(p)$$

The proof of this theorem goes through the steps in the laser method and analyses them. We shall first describe these steps in more detail. We restrict ourselves to the case in which T is a symmetric tensor with symmetric laser-ready partitions.

Write $X_1, \ldots, X_r$ for the partition classes of X, also called blocks, and use similar notation for the partitions of Y and Z. The tensor T has a tight block support so we may as well choose the labels of the blocks such that $X_iY_jZ_k$ is in the block support only if $i + j + k = c$ for some fixed c. Take a very large power $T^{\otimes n}$ and use the power partition. Each block in this partition is identified with an element of $[r]^n$. The block support of $T^{\otimes n}$ thus consists of block triples $X_IY_JZ_K$ where $I, J, K \in [r]^n$ are considered sequences of length n. As the base tensor T had tight support, each non-zero triple must satisfy $I_t + J_t + Z_t = c$ for all $t \in [n]$.

The method considers $T^{\otimes n}$ and then uses multiple steps of zeroing out to reach an independent set of partition subtensors T_{IJK}. Each subtensor T_{IJK} is the tensor product of partition subtensors T_{ijk} from T. If the value $V_\rho(T_{ijk})$ of each partition subtensor T_{ijk} is known, then Lemma 3.8.3(ii) implies a lower bound for the value of T_{IJK}. Lemma 3.8.3(i) and (iii) show that this gives a lower bound for the value of T. The zeroing out steps will be chosen such that all subtensors that remain at the end will have the same optimised value. This eases the analysis, but also ensures that we do not zero out all high-value subtensors in favor of low-value ones

The algorithm will only perform zeroing outs, so all the changes can be described through the support. We can thus analyse this support directly in which case $I, J, K \in [r]^n$ are our blocks and we let $S \subseteq [r]^n \times [r]^n \times [r]^n$ be the block support of $T^{\otimes n}$.

Define a symmetric distribution p on the block support of T. The algorithm aims to only keep block triples of $T^{\otimes n}$ which are approximately of type p. Fix n. We will now describe the zeroing out from $T^{\otimes n}$ in the laser method.

Remark 6.1.3. Note that for the fixed distribution p the expression $p(a, b, c) \cdot n$, which appears in the definition of types, see Definition 3.7.2, might not always be an integer. Instead, we pick a distribution α whose probabilities are all multiples of $\frac{1}{n}$. We choose α such that these distribution approach p as n grows large. The quantities and values in Theorem 6.1.2 are continuous in the chosen probability distribution, so the p dependence in the lower bound is achieved in the limit. The rigorous treatment of this issue can be found in [AW20].

Step 1

Pick a distribution α as described in the remark. We wish to only keep block triples of type α. However, we cannot pick triples to zero out and instead are restricted to zeroing out variables. If (I, J, K) is a block triple of type α then that does give information about the types of I, J, K. Thus we only want to keep X-blocks I that are of the type α_X, the X-marginal. Therefore we zero out any I that is of a different type and do the same for Y- and Z- blocks which are not of type α_Y and α_Z respectively. By symmetry these marginals are all the same.

Remark 6.1.4. At this point all blocks have the right marginal type. However, this does not mean that all block triples are of type α. In the initial analysis by Strassen [Str87] and Coppersmith and Winograd [CW90] this was not an issue, because there is only one distribution on the block support of CW_q and $CW_q^{\otimes 2}$ given the marginals. For example, let β be a probability distribution on the block support of CW_q with fixed marginals. Then $\beta(0, 0, 2)$ is determined by $\beta_Z(2)$. After establishing $\beta(0, 0, 2)$ and its rotations, we can use $\beta(0, 1, 1) = \beta_X(0) - \beta(0, 0, 2) - \beta(0, 2, 0)$ to find β on the rest of the block support. However for higher powers of CW_q there are multiple options and this does become an important point.

Step 2

The next zeroing out step uses the probabilistic method. Choose an appropiately large prime Q and define random hash functions h_X, h_Y, h_Z from $[r]^n \to \mathbb{Z}/Q\mathbb{Z}$. We use the following random hashing. Let weights $w_0, \ldots, w_n$ and h_0 be chosen independently and uniformly at random in $\mathbb{Z}/Q\mathbb{Z}$. Define

$$\begin{aligned} h_X(I) &= h_0 + \sum_{t=1}^{n} w_t \cdot I_t \\ h_Y(J) &= h_0 + 2w_0 + \sum_{t=1}^{n} w_t \cdot J_t \\ h_Z(K) &= h_0 + w_0 + 2^{-1} \sum_{t=1}^{n} w_t (c - K_t). \end{aligned} \tag{6.1}$$

Remember that $I_t + J_t + K_t = c$ for all $t \in [r]^n$ and all triples (I, J, K) in the block support, so for any (I, J, K) in the block support

1. $h_X(I) + h_Y(J) = 2h_Z(K)$ always holds independent of the instances of $w_0, \ldots, w_n$ and h_0.
2. h_X is uniformly distributed because h_0 is. The same holds for h_Y and h_Z. Additionally, h_Y is independent of h_X as we can condition on h_0 and $w_1, \ldots, w_n$ and then h_Y is still uniformly distributed because w_0 is. A similar argument shows independence for the other pairs.
3. For any two unequal I, I', the hash values $h_X(I)$ and $h_X(I')$ are independent as I, I' must differ in a coordinate t and then w_t is independent and uniformly distributed.

Pick a large Salem-Spencer set $U \subseteq \mathbb{Z}/Q\mathbb{Z}$. A Salem-Spencer set is a set U which contains no non-trivial arithmetic progressions of size 3. Thus if $a, b, c \in U$ and $a + b = 2c$ then $a = b = c$. It was shown by Behrend [Beh46] that such sets can be of size $Q^{1-o(1)}$. Now we zero out any block that does not hash to a value in U. We show that there is a choice of $w_0, \ldots, w_n$ and h_0 for which we keep close to $\frac{1}{Q}$ of block triples in this step. The choice of weights w_i and h_0 will also be important for later steps. We want to pick these weights, so that the final value bound is maximised. At this point the block support consists of triples (I, J, K) such that they have the right marginal types and $h_X(I), h_Y(J), h_Z(K) \in U$. Item 1. in the list above implies that these hashed values form an arithmetic progression, which means that $h_X(I) = h_Y(J) = h_Z(K) \in U$ as U is a Salem-Spencer set.

Step 3

In the next step we greedily zero out blocks until the set of remaining triples of type α is independent. We repeatedly takes a triple of type α and zero out one of its blocks if it is shared with other triples of type α. We show that there is also a choice of $h_0, w_0, \ldots, w_n$ such that we are still left with a large number of triples of type α after this step.

Step 4

The final step gets rid off all triples which are not of type α, after which we must be left with an independent set. This step was originally also done greedily, but in [AW20] the authors found an improvement through another probabilistic argument. First we zero out any block that is not in a triple of type α. Then pick random subsets of the remaining X-, Y- and Z-blocks and zero out any blocks not in the random subset. The random subset is chosen in such a way that all remaining triples must be of types α. Then use the independence of triples of type α to find a lower bound for the expectation of the number of remaining triples. Now there must be a choice of random hashing and random subset for which the remaining number of independent triples is at least this lower bound.

6.2 Motivation of choices in the laser method

In this algorithm we assumed that our tensor and partitions are symmetric. As mentioned before this is a reasonable assumption. Given any tensor T with a triple of partitions $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ we know that $T_{sym} = T \otimes \mathrm{rot}(T) \otimes \mathrm{rot}(\mathrm{rot}(T))$ is symmetric. It can be given partitions $(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}, \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}, \mathcal{X} \times \mathcal{Y} \times \mathcal{Z})$. This is a symmetric triple of partitions. Thus T_{sym} fits into our analysis. Lemma 3.8.3 implies that $V_\rho(T_{sym}) \geq V_\rho(T)^3$. More specifically the lower bound for the value found by the laser method also follows this supermultiplicativity. This means that it is always best to analyse symmetric tensors with symmetric partitions. Thus we can restrict our analysis to these tensors. On the other hand the statement is true in general [AW20].
It also seems reasonable to restrict to symmetric distributions. The entropy function is concave, thus $\min\{H(p_X), H(p_Y), H(p_Z)\}$ is maximal for a symmetric distribution just as in Corollary 4.3.4. This heuristic does not exclude the possibility that there are tensors where the Γ-term could affect the value in such a way that an asymmetric distribution is actually the maximiser for the lower bound.

Some of the steps might seem arbitrary and we now aim to give some explanation why these steps are reasonable to get a good lower bound for the value. The first idea, which is captured in step 1, is that in the zeroing out of $T^{\otimes n}$, we wish to only leave blocks of some type α. This does mean we might throw away some block triples preemptively that could have been a part of the final independent set. However, there are only polynomially many types, because of which throwing away all these other types only loses us a polynomial factor of other block triples. This factor is $o(1)$ in the final lower bound. On the other hand, we can now distinguish between 'good block triples', those block triples of type α, and other block triples. The other steps will try to keep as many good block triples as possible. If we choose p and thus α such that the good block triples have high values, then this looks like an effective strategy to keep a large set of high value subtensors. This step also helps with making our block support independent as there can only be relatively few block triples that all have a block in common.

In step 2 we effectively lose a factor of $\frac{1}{Q}$ of all good block triples. In the final lower bound this comes down to having a $H(p_X)$ term instead of a $H(p)$ term. Although this step loses some triples, it is currently the most efficient way to get a nearly independent set of good block triples. It is nearly independent in the sense that the number of pairs that have a block in common is smaller than the number of good block triples as can be seen in Lemma 6.3.4. Step 3 is efficient enough that it does not impact the final lower bound. Finally, step 4 is necessary as there can still be terms which are not of type α. This step is the reason for the $\frac{1}{2}\Gamma(p)$ term in the lower bound. It is argued in [AW20] that this term cannot be improved for all possible starting tensors.

6.3 Proof of Theorem 6.1.2

In this section we work towards the proof of Theorem 6.1.2. We largely follow the proof in [AW20]. The proof of this theorem consists of going through the laser method algorithm and analysing the tensor as it goes through the steps. At the end of the algorithm all partition subtensors are of type α. The value of such a subtensor can be lower bounded.

Lemma 6.3.1. *Let T be a symmetric partitioned tensor with r parts in the partition. Let $T_{IJK} \subseteq T^{\otimes n}$ for $I, J, K \in [r]^n$ be a subtensor of type α. Then*

$$\log V_\rho(T_{IJK}) \geq n \sum_{i,j,k \in [r]} \alpha(i,j,k) \log V_\rho(T_{ijk})$$

Proof. The value is supermultiplicative, see Lemma 3.8.3ii. By definition, $T_{IJK} = \bigotimes_{t=1}^n T_{I_t J_t K_t}$. Each of these $T_{I_t J_t K_t}$ is one of T_{ijk} with $i, j, k \in [r]$ and each T_{ijk} appears $\alpha(i,j,k)n$ times in the tensor product. Thus indeed

$$V_\rho(T_{IJK}) = V_\rho(\prod_{ijk} T_{ijk}^{\otimes \alpha(i,j,k)n}) \geq \prod_{i,j,k \in [r]} V_\rho(T_{ijk})^{\alpha(i,j,k)n}.$$

Taking the logarithm on both sides proves the lemma. □

Now we analyse the number of partition subtensors left at the end of the algorithm. We start with a symmetric set $S \subseteq [r]^n \times [r]^n \times [r]+^n$. Note that in step 2 we introduce randomness. We only fix our choice of random hashings at the end. Define S_1 as the support after step 1 and let S_2, S_3, S_4 be the random variables which represent the supports that remain after step 2, 3 and 4 respectively. Let $A_i \subseteq S_i$ be the (random) set of triples of type α. Note that $A_4 = S_4$ and that we want to choose a random hashing that makes $|A_4|$ as large as possible. We bound each of the $|S_i|$ and $|A_i|$ in consecutive lemmas.

Lemma 6.3.2. *For any symmetric distribution α we have $\log |S_1| \leq (\Gamma(\alpha) + H(\alpha))n + o(n)$ and $\log |A_1| = H(\alpha)n + o(n)$.*

Proof. In step 1 we zero out all blocks that do not match the marginal distribution of α. Any $(I, J, K) \in S$ of type α is also in S_1 as it has the right marginals. α was defined on the support of T, so any triple in $[r]^n \times [r]^n \times [r]^n$ of type α is also in S. Lemma 3.7.4 says that the number of such triples is $2^{H(\alpha)n+o(n)}$ and thus $\log |A_1| = H(\alpha)n + o(n)$. Any triple (I, J, K) that remains is of some type β such that the marginals of β are $\alpha_X, \alpha_Y, \alpha_Z$, thus $\beta \in \text{Marg}(\alpha)$. Group all remaining triples based on which type they have and then we know

$$|S_1| \leq \sum_{\beta \in \text{Marg}(\alpha)} 2^{H(\beta)n+o(n)} \tag{6.2}$$

By definition of $\Gamma(\alpha)$ we know that $H(\beta) \leq \Gamma(\alpha) + H(\alpha)$ for all $\beta \in M_\alpha$. Furthermore any triple must be of a type whose entries are all multiples of $\frac{1}{n}$. There are only polynomially many of such distributions 3.7.3. This means that there is a polynomial f such that the sum in 6.2 consists of $\leq f(n)$ many terms. We conclude that $\log |S_1| \leq \log(f(n)) + (\Gamma(\alpha) + H(\alpha))n + o(n)$ and $\log(f(n))$ is also $o(n)$. □

In order to do the analysis for step 2 we also need the following information.

Lemma 6.3.3. *After step 1 there are $N := 2^{H(\alpha_X)n+o(n)}$ X-blocks that are not zeroed out and each remaining block is in exactly $R := \frac{|A_1|}{N}$ triples of type α.*

Proof. The number of elements of $[r]^n$ that are of type α_X is $2^{H(\alpha_X)n+o(n)}$, so there are indeed N X-blocks left. We can use the same argument using permutations as in the proof of Proposition 4.3.3 to argue that all X-blocks are in the same number of elements of A_1. Thus each block is indeed in R triples. By symmetry the same holds for Y-blocks and Z-blocks. □

In step 2 we pick a prime Q. We require that this prime is in $[10R, 20R]$. Such a prime exists by Bertrand's postulate as $R \geq 1$. Then we pick a Salem-Spencer set $U \subseteq \mathbb{Z}/Q\mathbb{Z}$ and use a probabilistic analysis to establish that there is a hashing that has the properties we want. For this analysis we shall first establish the expected value of some relevant random variables.

Lemma 6.3.4. *For a random hashing as defined in Equation 6.1 let C be the number of pairs of triples $(I, J, K), (I', J', K') \in A_2$ such that $I = I'$, $J = J'$ or $K = K'$. Then we have*

- $\mathbb{E}(|A_2|) = \frac{|A_1||U|}{Q^2}$.
- $\mathbb{E}(|S_2|) = \frac{|S_1||U|}{Q^2}$.
- $\mathbb{E}(C) \leq \frac{3|A_1||U|}{20Q^2}$

Proof. The first two expected values are straightforward to establish. Recall that the property of the Salem-Spencer set implies that any triple $(I, J, K) \in \mathcal{S}$ satisfies $h_X(I) = h_Y(J) = h_Z(K) \in U$. The hashing functions h_X, h_Y are independent and determine h_Z. Thus $\mathbb{P}(h_X(I) = h_Y(J) = h_Z(K) \in U) = \mathbb{P}(h_X(I) \in U)\mathbb{P}(h_Y(J) = b | h_X(I) = b) = \frac{|U|}{Q^2}$. Summing over all relevant triples gives us the first two expectations.

For $\mathbb{E}(C)$ we note that two distinct $(I, J, K), (I, J', K') \in A_1$ can have at most one block in common. For example, if $I = I'$ and $J = J'$ then by tightness we also get $K = K'$. An X-block I is in R triples, so there are $\binom{R}{2}$ triples that share I. This is the same for all X-blocks, so there are $N\binom{R}{2}$ triples that share an X-block. Take any such pair (I, J, K) and (I, J', K') both in A_1. In order for both of these to be in A_2 we require $h_X(I) = u$ for some $u \in U$ and then also $h_Y(J) = h_Y(J') = u$. All these events are independent, so we get

$\mathbb{P}((I,J,K),(I,J',K') \in A_2) = \frac{|U|}{Q^3}$. The same holds for Y-blocks and Z-blocks, so summing over all blocks this gives $\mathbb{E}(C) = 3N\binom{R}{2}\frac{|U|}{Q^3} \leq \frac{3NR^2|U|}{2Q^3}$. By definition of R and because $Q \geq 10R$ we get $\mathbb{E}(C) \leq \frac{3|A_1||U|}{20Q^2}$. □

Lemma 6.3.5. $\mathbb{E}(|S_3|) \leq \frac{|S_1||U|}{Q^2}$ *and* $\mathbb{E}(|A_3|) \geq \frac{7|A_1||U|}{10Q^2}$.

Proof. At the very least $S_3 \subseteq S_2$ as more triples are zeroed out in step 3. Thus $\mathbb{E}(|S_3|) \leq \mathbb{E}(|S_2|) \leq \frac{|S_1||U|}{Q^2}$. For $|A_3|$ we analyse step 3 a bit more. In step 3 we pick out one block I at a time. If this block is shared by $r \geq 2$ triples in A_2 then it is zeroed out. For each zeroing out we lose $\binom{r}{2} \geq r/2$ pairs of triples that share a block and r triples in A_2. If we lose more than $2C$ triples in A_2 then we would have also lost more than C pairs which is impossible. Thus at the end of step 3 there are at least $|A_2| - 2C$ triples of type α left. Therefore

$$\mathbb{E}(|A_3|) \geq \mathbb{E}(|A_2| - 2C) \geq \frac{|A_1||U|}{Q^2} - 2\frac{3|A_1||U|}{20Q^2} = \frac{7|A_1||U|}{10Q^2}$$

□

In step 4 we start with the random sets A_3 and S_3 and zero out the remaining triples which are not of type α. In order to analyse $|A_4|$ we need the following lemma.

Lemma 6.3.6 ([AW20]). *Given a set a and $S \subseteq a^3$ such that*

- *$(i,i,i) \in S$ for all $i \in a$*
- *for all other $(i,j,k) \in S$ the three values i,j,k are distinct.*

Then there is a subset $b \subseteq a$ of size $\geq \frac{2|a|^{3/2}}{3(3|S|)^{1/2}}$ such that $S|_{b\times b\times b} = \{(i,i,i) : i \in b\}$.

Proof. We follow the proof in [AW20]. We define a random subset $A \subseteq a$. Each element is included in A independently with probability $p = \frac{|a|^{1/2}}{\sqrt{3(|S|-|a|)}}$. Consider the set $S' \subseteq S|_{A\times A\times A}$ of all off-diagonal elements. For a specific $(i,j,k) \in S$ with $i \neq j \neq k \neq i$ each of i,j and k is included in A with probability p. By independence we have $(i,j,k) \in S'$ with probability p^3. It follows that $\mathbb{E}(|S'|) = (|S| - |a|)p^3$.

We remove the set $A' = \{i \mid (i,j,k) \in S'\}$ from A. This gives a set B with

$$\mathbb{E}(|B|) = \mathbb{E}(|A|) - \mathbb{E}(|A'|) \geq \mathbb{E}(|A|) - \mathbb{E}(|S'|) = p|a| - (|S| - |a|)p^3 = \frac{2|a|^{3/2}}{3^{3/2}(|S| - |a|)^{1/2}} \geq \frac{2|a|^{3/2}}{3(3|S|)^{1/2}}$$

There is a choice of randomness such that the set B is larger than this expectation. We choose this set as b. If there is $(i,j,k) \in S|_{b\times b\times b}$ with $i \neq j$ or $i \neq k$, then none of i,j,k are the same. This implies that (i,j,k) would have been in S', but then $i \in A'$ and thus $i \notin b$. We conclude that $S|_{b\times b\times b} = \{(i,i,i) : i \in b\}$. □

Corollary 6.3.7. *Given A_3 and S_3, there is a choice of subsets in step 4 such that $|A_4| \geq \frac{2}{3\sqrt{3}}|A_3|^{3/2}|S_3|^{-1/2}$.*

Proof. In this step we first zero out all blocks that are not in a triple of type α. Suppose $X', Y', Z' \subseteq [r]^n$ are the sets of remaining X-, Y- and Z-blocks after this zeroing out and $S_3' \subseteq X' \times Y' \times Z'$ is the set of remaining triples. Note that $A_3 \subseteq S_3'$. The triples of type α form an independent set at this point, so each block is in a unique triple of type α. We define a bijection $f_X : X' \to A_3$ by sending $I \in X'$ to the unique triple $(I,J,K) \in A_3$ which it is a part of. We similarly define bijections $f_Y : Y' \to A_3$ and $f_Z : Z' \to A_3$. The function $f := f_X \times f_Y \times f_Z : X' \times Y' \times Z' \to A_3 \times A_3 \times A_3$ is a bijection and therefore is injectively on S_3'. The set $f(S_3') \subseteq A_3 \times A_3 \times A_3$ of size $|S_3'|$ satisfies the conditions in Lemma 6.3.6.

- For any $B = (I,J,K) \in A_3$ we have $f(I,J,K) = (B,B,B)$.
- Suppose without loss of generality that for some block triples B_1, B_2 we have $(B_1, B_1, B_2) \in f(S_3')$. Then there is a $B = (I,J,K) \in S_3' \setminus A_3$ with $f_X(I) = f_Y(J) = B_1$, but then $B_1 = (I,J,K')$ for $K' \neq K$. This is impossible because S_3' is tight.

Thus Lemma 6.3.6 provides us with a set $b \subseteq A_3$. Let $X'' = f_X^{-1}(b)$ and define Y'', Z'' analogously. We know $f(S_3')|_{b\times b\times b} = \{(i,i,i) : i \in b\}$ which must all come from triples in A_3. In step 4 we zero out all blocks in $X' \setminus X''$, $Y' \setminus Y''$ and $Z' \setminus Z''$. Then all remaining triples are of type α and these are all independent. Each block is in a triple, so $|A_4| = |X''| = |b|$ and thus we have found subsets for which $|A_4| \geq \frac{2}{3\sqrt{3}}|A_3|^{3/2}|S_3'|^{-1/2} \geq \frac{2}{3\sqrt{3}}|A_3|^{3/2}|S_3|^{-1/2}$. □

It is now clear that we want to pick the random hashings such that $|A_3|^{3/2}|S_3|^{-1/2}$ is as large as possible. We can get at least as good as the expectation.

Lemma 6.3.8. *If X, Y are random variables with values in $[0,\infty)$ on the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$, then there must be an instance $\sigma \in \Omega$ such that $X(\sigma)^{3/2}Y(\sigma)^{-1/2} \geq \mathbb{E}(X)^{3/2}\mathbb{E}(Y)^{-1/2}$.*

Proof. We prove this by contradiction. Suppose this is not the case, so $X(\sigma)^{3/2}Y(\sigma)^{-1/2} < \mathbb{E}(X)^{3/2}\mathbb{E}(Y)^{-1/2}$ for all σ. All terms are non-negative, so this can be squared to give $X(\sigma)^3 < \mathbb{E}(X)^3\mathbb{E}(Y)^{-1}Y(\sigma)$ for all σ. It is a standard measure theoretic fact that the strict inequality is preserved if we take the expectation, which means that

$$\mathbb{E}(X^3) < \mathbb{E}(X)^3\mathbb{E}(Y)^{-1}\mathbb{E}(Y) = \mathbb{E}(X)^3.$$

However, the function $x \mapsto x^3$ is convex on the non-negative reals, so Jensen's inequality says that $\mathbb{E}(X^3) \leq \mathbb{E}(X)^3$. Contradiction. □

We are now ready to combine all these lemmas into a proof of Theorem 6.1.2.

Proof of Theorem 6.1.2. The algorithm gives us an independent set of size $|A_4| \geq \frac{2}{3\sqrt{3}}|A_3|^{3/2}|S_3|^{-1/2}$. By the previous lemma there is a choice of random hashing such that $|A_3|^{3/2}|S_3|^{-1/2} \geq \mathbb{E}(|A_3|)^{3/2}\mathbb{E}(|S_3|)^{-1/2}$. Pick such a choice. Then we get

$$\begin{aligned}
|A_4| &\geq \frac{2}{3\sqrt{3}}\mathbb{E}(|A_3|)^{3/2}\mathbb{E}(|S_3|)^{-1/2} \\
&\geq \frac{2}{3\sqrt{3}}\left(\frac{7|A_1||U|}{10Q^2}\right)^{3/2}\left(\frac{|S_1||U|}{Q^2}\right)^{-1/2} \\
&= \frac{14\sqrt{7}}{30\sqrt{30}}\frac{|A_1|^{3/2}|U|}{Q^2|S_1|^{1/2}}.
\end{aligned}$$

Now we use $|U| = Q^{1-o(1)}$, $Q \leq 20R$ and the bounds for $|A_1|$ and $|S_1|$ to say

$$\begin{aligned}
\log|A_4| &\geq -(1+o(1))\log Q + \frac{3}{2}\log|A_1| - \frac{1}{2}\log|S_1| + \log 0.01 \\
&= -\log R + \frac{3}{2}H(\alpha)n - \frac{1}{2}(\Gamma(\alpha) + H(\alpha))n + o(n) \\
&= \log N - \frac{1}{2}\Gamma(\alpha)n + o(n)
\end{aligned}$$

Thus after step 4 we have found a zeroing out from $T^{\otimes n}$ to a direct sum of $|A_4|$ subtensors T_B which are all of type α. The value is superadditive for direct sums, so for the tensor T we get

$$V_\rho(T) = V_\rho(T^{\otimes n})^{1/n} \geq V_\rho\left(\bigoplus_{B\in A_4} T_B\right)^{1/n} \geq \left(\sum_{B\in A_4} V_\rho(T_B)\right)^{1/n} = (|A_4|V_\rho(T_B))^{1/n}$$

and by Lemma 6.3.1

$$\begin{aligned}\log V_\rho(T) &\geq \frac{1}{n}\left(\log|A_4| + \log V_\rho(T_B)\right)\\ &\geq \frac{1}{n}\left(\log N - \frac{1}{2}\Gamma(\alpha)n + o(n) + n\sum_{i,j,k\in[r]}\alpha(i,j,k)V_\rho(T_{ijk})\right)\\ &= H(\alpha_X) - \frac{1}{2}\Gamma(\alpha) + \sum_{i,j,k\in[r]}\alpha(i,j,k)V_\rho(T_{ijk}) + o(1).\end{aligned}$$

As $n \to \infty$ the $o(1)$ term goes to zero, so we recover the intended expression. □

Most bounds on ω are based on Theorem 6.1.2. For example, choosing $T = CW_6$ and p symmetric such that $p(2,0,0) \approx 0.016$ and $p(1,1,0) \approx 0.317$, gives the bound $\omega < 2.39$.

6.4 Advanced laser methods

After the initial paper about the laser method by Strassen [Str87], there has been further improvement on the exponent of matrix multiplication. All of these have been based on the lower bound on the value from Theorem 6.1.2. Each improvement has used a power of CW_q as the base tensor, so the asymptotic rank did not improve. However the lower bounds on the values of the partition subtensor were relatively better. The improved lower bounds came through application of merging. The initial start tensor was $T = CW_q$ with the standard partition. Each of the partition subtensors is a matrix multiplication tensor. When moving our attention to $CW_q^{\otimes 2}$ we could just use the power partition and get the same value, but we can also combine some block triples to improve the value. In particular

$$T_{0,1,1}\otimes T_{0,1,1} + T_{0,2,0}\otimes T_{0,0,2} + T_{0,0,2}\otimes T_{0,2,0} \cong \sum_{i,i'=1}^{q} x_{(0,0)}y_{(i,i')}z_{(i,i')} + x_{(0,0)}y_{(q+1,0)}z_{(0,q+1)} + x_{(0,0)}y_{(0,q+1)}z_{(q+1,0)}$$

which is isomorphic to $\langle 1,1,q^2+2\rangle$. This has a better value than the sum of the values of the three individual tensors. This idea of merging blocks together can thus improve the lower bound for the value of some subtensors. In $CW_q^{\otimes 2^k}$ we merge blocks X_I with $I \subseteq \{0,1,2\}^{2^k}$ together if they have the same value of $\sum_{t=1}^{2^k} I_t$. This merging also creates partition subtensors which are not matrix multiplication tensors, vastly increasing the complexity of the analysis. By recursively lower bounding the values of the subtensors and taking advantage of the merging behaviour one can improve the bound on ω.

Table 1 Improvements using the laser method through the years.

Authors	year	Bound on ω	Method
Coppersmith and Winograd	1987	2.376	Laser method applied to $CW_q^{\otimes 2}$
Stothers	2010	2.374	Starting tensor $CW_q^{\otimes 4}$
Vassilevska Williams	2011	2.37288	Starting tensor $CW_q^{\otimes 8}$
Le Gall	2014	2.37287	Starting tensor $CW_q^{\otimes 32}$
Alman and Vassilevska Williams	2020	2.37286	Probabilistic argument for step 4

In 2014 Ambainis, Filmus and Le Gall [AFG15] showed that 2.3725 is a lower bound on the value of ω that you can find through applying the laser method to higher and higher powers of CW_q when using the canonical method of merging described above. Each of the above algorithms uses this canonical method and thus much further improvement in this same line is not possible. In their recent paper Duan, Wu and Zhou [DWZ22] broke through this barrier and showed that $\omega \leq 2.37187$ by performing a less wasteful analysis of the recursive steps in the value lower bounding.

Their idea is the following. Suppose you are trying to bound the value of T via the laser method. In order to do this you need to also lower bound the value of some subtensor T_{ijk} which is not a matrix multiplication tensor.

Therefore we also perform the laser method on T_{ijk}. For this the partition classes X_i, Y_j, Z_k are themselves partitioned and then we describe a zeroing out of $T_{ijk}^{\otimes n}$ based on these partitions. If $\alpha(i,j,k)m = n$ then this essentially describes a further zeroing out of $T^{\otimes m}$. However, this further zeroing out might have helped in making the block triples more independent. Performing the zeroing out of T_{ijk} before performing step 3 for T, might make the support more independent, therefore needing less zeroing out in step 3 and 4. Ultimately, this gives a larger A_4 and thus improves the lower bound on $V_\rho(T)$. The complexity of this method is much higher, not the least because they treat Z-variables asymmetrically from X- and Y-variables.

7 Alternative base tensors

The current laser method is limited to showing $\omega \leq 2.30$ [AFG15]. However, it is conjectured that $\omega = 2$, because of which a lot of effort is put into improving the upper bound. There are no known ways to construct good degenerations which are not zeroing outs, so it seems that we are limited to the laser method. One potential way the bound can be improvement is through better bounds for $\tilde{R}(cw_q)$. Currently, we only know $q + 1 \leq \tilde{R}(cw_q) \leq q + 2$ given by the conciseness and border rank bounds. Alternatively, we could use other starting tensors than the Coppersmith-Winograd tensors to give better bounds for ω. In this chapter we look at some alternative starting tensors and perform some computations to judge whether they could be good starting tensors. This does not lead to any new breakthroughs, but we record our efforts in the hope they can be continued by others. Our search for alternative base tensors looks at a family of tensors that contains the Coppersmith-Winograd tensors. Our search was still restricted to very specific tensors. This was mostly to ease the analysis, but we give some motivation for doing so.

First of all, we restrict to tensors with a laser-ready partition. These have a tight block support and each subtensor T_{ijk} is a matrix multiplication tensor. This means we can easily apply the laser method to this tensor. In earlier chapters we have seen that symmetric tensors give the best value and best asymptotic rank, therefore we further restrict to symmetric laser-ready tensors. We generate these by picking a symmetric tight set and then assigning matrix multiplication tensors to each block triple. These assignments are restricted by the sizes of blocks. Let us consider the following example with the support of the Coppersmith-Winograd tensor.

Suppose we have decided to construct symmetric laser-ready tensors T with block support

$$\{(0,1,1),(1,1,0),(1,0,1),(0,0,2),(0,2,0),(2,0,0)\}.$$

Let $T_{011} \cong \langle m,n,p\rangle$ with $m,n,p \geq 1$. Then we get $|X_0| = mn, |Y_1| = np$ and $|Z_1| = pm$. By symmetry we must have that $|X_1| = |Y_1| = |Z_1|$. This means that $np = pm$, so $m = n$. If $T_{0,0,2} \cong \langle m',n',p'\rangle$, then $m^2 = |X_0| = m'n'$ and by symmetry this must also be $n'p'$. We conclude that $p' = m' = \frac{m^2}{n'}$. Thus the laser-ready tensor is determined by m, p and n'.

The other important feature of a good starting tensor is a low asymptotic rank. If $m,n,p > 1$ then the asymptotic rank of T_{011} is $(mnp)^{\omega/3}$. There is a zeroing out from T to T_{011}, so $\tilde{R}(T) \geq (mnp)^{\omega/3}$. Intuitively this makes it harder to use such a tensor for an upper bound on ω. It might be possible to use partition subtensors of the form $\langle m,n,1\rangle$ with $m,n > 1$, but note that the Coppersmith-Winograd tensor only consists of subtensors of the form $\langle m,1,1\rangle$. A potential reason for this is that it was only possible to find a good border rank decomposition, because of this feature. We follow this practice and only allow partition subtensors where two out of three of m,n and p are equal to 1.

We return to our example tensor T and enforce each subtensor to have two out of three of m,n and p equal to 1. The previously found condition $m = n$ implies that these must both be equal to 1. Then we must also get $m' = n' = p' = 1$ and thus we must get that $T_{011} = \langle 1,1,q\rangle$ and $T_{002} = \langle 1,1,1\rangle$ for some $q \geq 1$. This means that we can only construct Coppersmith-Winograd tensors CW_q. We show that the restrictions we have enforced so far imply uniqueness for other supports as well. We will write $[n]_0$ to mean the set $\{0,1,\ldots,n\}$ which we use to follow standard notation for CW_q.

Definition 7.0.1. A set $\Phi \subseteq [d]_0 \times \cdots \times [d]_0$ is *fully tight* if $\Phi = \{(i_1,\ldots,i_k) : i_1 + \cdots + i_k = d\}$.

Lemma 7.0.2. *Let T be a symmetric tensor with symmetric partition such that each of its partition subtensors is of the form $\langle 1,1,m\rangle, \langle 1,m,1\rangle$ or $\langle m,1,1\rangle$ for some positive integer m which may differ per subtensor. If the block support of T is fully tight, then there can only be one $q>1$ such that there are partition subtensors isomorphic to $\langle q,1,1\rangle$, $\langle 1,q,1\rangle$ and $\langle 1,1,q\rangle$.*

Proof. Let d be the value such that the block support of T is equal to $\{(i,j,k)\in[d]_0\times[d]_0\times[d]_0 : i+j+k=d\}$. We will now determine the size of each X_i. This uniquely determines each partition subtensor up to isomorphism as we know these are all isomorphic to matrix multiplication tensors $\langle 1,1,m\rangle$ or a rotation of it. This subtensor must be defined on blocks of sizes $1,m,m$ in some order. For any $\ell\in\{0,1,\dots,\lfloor\frac{d}{2}\rfloor\}$, there is a triple $(d-2\ell,\ell,\ell)$ in the support. We know that $X_{d-2\ell}, Y_\ell, Z_\ell$ must be $1,m,m$ in some order. By symmetry we have $|Y_\ell|=|Z_\ell|$ and therefore $|X_{d-2\ell}|$ has to be 1. We assume that there is an index i such that there are j,k for which $T_{ijk}=\langle q,1,1\rangle$ with $q>1$. Then $|X_i|=q$, so i and d cannot have the same parity. If d is even then $d-i-1\equiv d \mod 2$, so the triple $(i,d-i-1,1)$ shows that $|X_1|=q$, because $|X_i|=q$, $|Y_{d-i-1}|=1$, so $|Z_1|=q$ and there is symmetry. This argument can now be used to say that any j with $j\not\equiv d \mod 2$ has $|X_j|=q$, because $(1,j,d-j-1)$ is a triple with $|X_1|=q>1$ and $|Y_{d-j-1}|=1$. If d is odd then $d-i\equiv d \mod 2$, so the triple $(i,d-i,0)$ shows that $|X_0|=q$. From here we show any $j\not\equiv d \mod 2$ has $|X_j|=q$ with the triple $(0,d-j,j)$.

The size of each block is uniquely determined and thus each partition subtensor is uniquely determined up to isomorphism and is $\langle 1,1,q\rangle$ or a rotation of it. □

Any tight set is isomorphic to the subset of a fully tight set. If a block support is not fully tight, then it is possible to have $p,q>1$ such that both $\langle p,1,1\rangle$ and $\langle q,1,1\rangle$ are subtensors. We do not consider these now.

Remark 7.0.3. For each d and q there are tensors that satisfy the lemma above. We have found that such tensors have block sizes $(|X_0|,\dots,|X_d|)$ equal to $(1,q,1,q,\dots,q,1)$ if d is even and equal to $(q,1,q,\dots,q,1)$ if d is odd. One example of such a tensor is

$$\mathrm{Full}(d,q) := \sum_{a+b+c=d}\sum_{i=1}^{|X_a|}\sum_{j=1}^{|Y_b|}\sum_{k=1}^{|Z_c|} x_i^a y_j^b z_k^c,$$

where $x_i^a\in X_a$, $y_j^b\in Y_b$, $z_k^c\in Z_c$. This family of tensors includes the small coppersmith-winograd tensors, $cw_q=\mathrm{Full}(1,q)$ and the big Coppersmith-Winograd tensors, $CW_q=\mathrm{Full}(2,q)$.

The family of tensors $\mathrm{Full}(d,q)$ consists of laser-ready tensors whose subtensors are all equal to matrix multiplication tensors. In Chapter 3 we showed that $h_\Lambda(\Phi)$ is an upper bound for the asymptotic slice rank. We can compute this upper bound.

The tensors $\mathrm{Full}(d,q)$ are laser-ready tensors in which each block is in a triple in the block support, so they are concise by Lemma 3.8.10. Therefore $\sum_{a=0}^{d}|X_a|$ is a lower bound for the asymptotic rank. The tensors are symmetric, so we can give lower bounds $f_u(d,q)$ for $\omega_u(\mathrm{Full}(d,q))$ based on Theorem 3.10.1. The python code that performs the necessary optimisations can be found in Appendix A. Note that all optimisations are over convex functions, so the code should find the appropiate optimum up to certain precision.

Table 2 The (rounded) lower bound $f_u(d,q)$ for $\omega_u(\mathrm{Full}(d,q))$ for the first few values of d and q.

$f_u(d,q)$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=7$	$q=8$	$q=9$
$d=1$	2.17795	2	2.02538	2.06244	2.09627	2.12549	2.15064	2.17245	2.19155
$d=2$	2.16805	2.17795	2.19146	2.20551	2.21913	2.23201	2.24405	2.25525	2.26567
$d=3$	2.15949	2.06528	2.0738	2.0959	2.11843	2.13902	2.1574	2.17377	2.18839
$d=4$	2.15237	2.14713	2.1482	2.15865	2.17158	2.18467	2.19718	2.20887	2.21971
$d=5$	2.14641	2.08064	2.08607	2.10402	2.12299	2.14066	2.15663	2.17098	2.1839
$d=6$	2.14135	2.13257	2.1323	2.1426	2.15547	2.16845	2.18079	2.19228	2.20289
$d=7$	2.137	2.08595	2.09029	2.10625	2.12339	2.13951	2.15417	2.16742	2.17939
$d=8$	2.13321	2.12335	2.12314	2.1335	2.14626	2.15905	2.17115	2.18237	2.19272
$d=9$	2.12987	2.08784	2.09168	2.10645	2.12245	2.13758	2.1514	2.16391	2.17526
$d=10$	2.1269	2.11675	2.11683	2.12724	2.13987	2.15245	2.16431	2.17529	2.1854
$d=11$	2.12423	2.0883	2.09188	2.10584	2.12104	2.13547	2.14868	2.16067	2.17156
$d=12$	2.12182	2.11168	2.11208	2.1225	2.13499	2.14737	2.15901	2.16978	2.17968
$d=13$	2.11963	2.0881	2.09153	2.1049	2.1195	2.13339	2.14614	2.15773	2.16827
$d=14$	2.11762	2.10761	2.10831	2.11871	2.13106	2.14326	2.1547	2.16528	2.175
$d=15$	2.11577	2.08757	2.09091	2.10382	2.11795	2.13142	2.1438	2.15507	2.16532
$d=16$	2.11407	2.10425	2.10519	2.11557	2.12778	2.13981	2.15108	2.16148	2.17104

We make a few observations about this table.

1. The lower bounds for $d=1$ and $d=2$ match the bounds for cw_q and CW_q found by Alman [Alm18].
2. Each of the lower bounds we have found is below 2.37, so each of these tensors could potentially be used in the universal method to give a better upper bound on ω. All of the lower bounds are above 2 except for the bound for $\mathrm{Full}(1,2) = cw_2$. Therefore we have not found a candidate besides cw_2 for establishing $\omega = 2$ via the universal method.
3. Each row is increasing from $q=2$. Assuming this holds for all q none of the $\mathrm{Full}(d,q)$ with $d \leq 7$ can establish $\omega = 2$. The function $f_u(d,q)$ was shown to be increasing in q for the big Coppersmith-Winograd tensor [Alm18].
4. Each displayed column with $q > 1$ is increasing in the odd rows and decreasing in the even rows. If this pattern holds then $\mathrm{Full}(d,q)$ cannot establish $\omega = 2$ for odd d. We also suspect that the values in the even rows do not approach two. In fact the table suggests that $f_u(d,q)$ is at least $f_u(2d+1,q)$.
5. In each row $d \geq 2$ in the table we have $f_u(d,1) \geq f_u(d,2)$.

The observations made in this table suggest that $\mathrm{Full}(d,q)$ can be used in the universal method for better upper bounds on ω. This only works with our current estimates of the asymptotic slice rank and asymptotic tensor rank. This does not mean that the current version of the laser method can obtain such an upper bound. The main obstruction is the fact that we do not have good upper bounds on the asymptotic tensor rank of these tensors. These are hard to find, but we can give a rough indication of what the asymptotic tensor rank has to be in order to give new upper bounds on ω.

Appendix A also has code that optimises the lower bound for the value from Theorem 6.1.2. From this we deduce which upper bound $r(d,q)$ of $\tilde{R}(\mathrm{Full(d,q)})$ is needed to prove $\omega \leq 2.37$ using the laser method as described in Chapter 5.

Table 3 Upper bound $r(d,q)$ on $\tilde{R}(\text{Full(d,q)})$ needed to prove $\omega \leq 2.37$.

$r(d,q)$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=7$	$q=8$	$q=9$
$d=1$	1.88988	3.26775	4.50153	5.65018	6.73939	7.78348	8.79148	9.76958	10.7223
$d=2$	2.7551	3.7872	4.84392	5.89158	6.91871	7.92224	8.90237	9.8605	10.7984
$d=3$	3.61072	6.17565	8.50736	10.6782	12.7367	14.7099	16.6149	18.4633	20.2638
$d=4$	4.46158	6.57696	8.75218	10.8431	12.8559	14.8007	16.6867	18.5218	20.3125
$d=5$	5.30973	9.00298	12.4022	15.5668	18.5677	21.4443	24.2214	26.9162	29.5409
$d=6$	6.13224	8.97685	11.6623	14.8419	17.7995	20.5308	23.4774	26.9627	29.5797
$d=7$	7.00129	11.2275	14.8018	18.1612	21.225	14.3365	26.9253	29.6681	32.1378
$d=8$	7.84129	11.6008	15.0347	20.519	24.418	28.1673	31.7934	35.3157	38.7487

We also make a few observations about this table.

1. In each row $r(d,q)$ is increasing in q. If q increases the value will also increase and thus $\tilde{R}(\text{Full}(d,q))$ can be larger. There is one exception: $r(7,6)$ is much lower. For some unknown reason the optimisation does not find an optimal value in this case. We have checked that we can indeed get a better value. With this better value $r(7,6)$ is around 24.1. We are unsure why this happened for only this value specifically.
2. We have mentioned that CW_6 gives the bound $\omega < 2.39$. In this table one can see why CW_6 gave such a good bound on ω. The bound $r(2,6)$ is very close to the asymptotic rank of CW_6, which is 8.
3. We can compare $r(d,q)$ to the lower bound given by conciseness. The only values in the table for which $r(d,q)$ is larger than this lower bound are

$$\begin{gathered}(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),(1,9),\\(3,2),(3,3),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9),\\(5,2),(5,3),(5,4),(5,5),(5,6),(5,7)\end{gathered}$$

The row with $d=1$ corresponds to cw_q. The other two rows, $d=3$ and $d=5$, contain promising alternative base tensors.
4. For $d \geq 6$ all $r(d,q)$ are below the lower bound from conciseness. We conjecture this is also true for values outside this table. It is therefore not possible to use Theorem 6.1.2 to find a new upper bound on ω using these tensors.
5. We also checked the value of Γ for each of these tensors. It seemed to be zero for all optimal distributions. We are unsure whether this is true in general for $\text{Full}(d,q)$.

8 Outlook

This thesis collected some results on the slice rank and the asymptotic slice rank. Some new results were added, but throughout this year a lot more questions were raised than answered. There are a lot of open conjectures in the field, but the new results in this thesis raise some more specific questions that we want to address.

In Chapter 3 Proposition 4.2.10 on the slice rank of k-tensors with support in arithmetic progressions was proved for $k \geq 6$. We remarked that it is unknown whether the proposition holds for $k=5$ or $k=4$. This question is related to the question of the maximal size of sets without length k arithmetic progressions. In particular it answers the question whether a straight-forward application of the slice rank can lead to exponential improvement on the upper bound. We also remarked that the relatively elementary Lemma 4.2.9 was not known for $k=5$. This author is aware of some unsuccessful efforts to prove the lemma for $k=5$. In these efforts linear orders that make constant sequences maximal in $\mathbb{Z}/n\mathbb{Z}$ for $k=5$ were found for some small values of n.

In Proposition 4.4.7 we showed that symmetric tensors with oblique support have maximal slice rank if and only if they are regular. It seems as though symmetric, oblique and regular support gives quite a lot of constraints on the tensor. Thus we wonder if it is possible to classify all tensors or all supports that are symmetric, oblique and regular. We also remarked that there are some difficulties in establishing an if and only if condition for the more general situation of Proposition 4.4.4, but this might still be possible. The laser method finds a large direct sum of matrix multiplication tensors in high powers $T^{\otimes n}$ of T if T has tight block support. Such a direct sum of matrix multiplication tensors has a regular support. Thus more generally, one can ask if it is always possible to find a zeroing out from $T^{\otimes n}$ to a large tensor T' with regular support. The size of T' is limited by the asymptotic slice rank of T, but we wonder how close it can get to this asymptotic slice rank for different starting tensors T.

Theorem 5.3.1 proved that asymptotic slice rank is multiplicative for oblique and symmetric tensors. We found this result after conjecturing that asymptotic slice rank is in fact submultiplicative for all symmetric tensors. This conjecture is still open. We believe this conjecture to be true, because we feel that the supermultiplicative behaviour of the slice rank arises from multiplying an i-slice with a j-slice with $i \neq j$. For symmetric tensors this supermultiplicativity has already taken place in high powers $T^{\otimes n}$ and is thus already accounted for in the asymptotic slice rank. This way of thinking also leads us to wonder if $SR(T^{\otimes k})$ is close to $\widetilde{\mathrm{SR}}(T)^k$ for all symmetric k-tensors T. More specifically, we ask whether there is a constant c, possibly dependent on k, such that $\widetilde{\mathrm{SR}}(T)^k \leq c \cdot SR(T^{\otimes k})$.

In Chapter 6 we explored some new base tensors. This chapter left a lot of room for further exploration. We list some options that the author considered, but could not go into. First of all, we chose tensors where all subtensors were of the form $\langle 1, 1, m\rangle$, but it might very well be possible to obtain bounds on ω via the laser method by allowing $\langle 1, m, n\rangle$. With the tensors $\mathrm{Full}(d, q)$ we established which asymptotic rank is needed to obtain new upper bounds on ω. The border rank of $\mathrm{Full}(d, q)$ is not known for $d > 2$ and this could be a way to find a new upper bound on ω. Additionally, our lower bound on the value of $\mathrm{Full}(d, q)$ need not be optimal. The best versions of the laser method use powers $CW_q^{2^k}$ where merging can be exploited. Such merging should also occur for $\mathrm{Full}(d, q)$ with $d > 2$. This could potentially also be exploited to get bounds on ω. Lastly, we observed a few patterns in Tables 2 and 3. It would be interesting to see if it can be proven that these patterns hold for all d and q.

References

[Alm18] Josh Alman. *Limits on the Universal Method for Matrix Multiplication.* 2018. DOI: 10.48550/ARXIV.1812.08731. URL: https://arxiv.org/abs/1812.08731.

[AW20] Josh Alman and Virginia Vassilevska Williams. *A Refined Laser Method and Faster Matrix Multiplication.* 2020. arXiv: 2010.05846 [cs.DS].

[AFG15] Andris Ambainis, Yuval Filmus and François Le Gall. 'Fast Matrix Multiplication'. In: *Proceedings of the forty-seventh annual ACM Symposium on Theory of Computing.* ACM, June 2015. DOI: 10.1145/2746539.2746554. URL: https://doi.org/10.1145%2F2746539.2746554.

[Beh46] Felix A. Behrend. 'On Sets of Integers Which Contain No Three Terms in Arithmetical Progression.' In: *Proceedings of the National Academy of Sciences of the United States of America* 32 12 (1946), pp. 331–2.

[Bin80] Dario Bini. 'Relations between exact and approximate bilinear algorithms. Applications'. In: *Calcolo* 17 (Jan. 1980), pp. 87–97. DOI: 10.1007/BF02575865.

[BLR80] Dario Bini, Grazia Lotti and Francesco Romani. 'Approximate Solutions for the Bilinear Form Computational Problem'. In: *SIAM J. Comput.* 9 (Nov. 1980), pp. 692–697. DOI: 10.1137/0209053.

[Blä+19] Markus Bläser et al. *Variety Membership Testing, Algebraic Natural Proofs, and Geometric Complexity Theory.* 2019. arXiv: 1911.02534 [cs.CC].

[Bor18] Sander J. Borst. 'Using the slice rank for finding upper bounds on the size of cap sets'. Delft University of Technology, 2018.

[BCS96] Peter Bürgisser, Michael Clausen and M. Amin Shokrollahi. *Algebraic Complexity Theory.* Springer Berlin, Heidelberg, 1996. ISBN: 978-3-540-60582-9. DOI: https://doi.org/10.1007/978-3-662-03338-8.

[CVZ21] Matthias Christandl, Péter Vrana and Jeroen Zuiddam. 'Universal points in the asymptotic spectrum of tensors'. In: *Journal of the American Mathematical Society* 36.1 (Nov. 2021), pp. 31–79. DOI: 10.1090/jams/996. URL: https://doi.org/10.1090%2Fjams%2F996.

[CW90] Don Coppersmith and Shmuel Winograd. 'Matrix multiplication via arithmetic progressions'. In: *Journal of Symbolic Computation* 9.3 (1990). Computational algebraic complexity editorial, pp. 251–280. ISSN: 0747-7171. DOI: https://doi.org/10.1016/S0747-7171(08)80013-2. URL: https://www.sciencedirect.com/science/article/pii/S0747717108800132.

[DN05] Paolo D'Alberto and Alexandru Nicolau. 'Using Recursion to Boost ATLAS's Performance.' In: Jan. 2005, pp. 142–151.

[DS13] Alexander M. Davie and Andrew J. Stothers. 'Improved bound for complexity of matrix multiplication'. In: *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 143 (Apr. 2013). DOI: 10.1017/S0308210511001648.

[DeB+21] Louis DeBiasio et al. *Generalizations and strengthenings of Ryser's conjecture.* 2021. arXiv: 2009.07239 [math.CO].

[Der22] Harm Derksen. 'The G-stable rank for tensors and the cap set problem'. In: *Algebra & Number Theory* 16.5 (Aug. 2022), pp. 1071–1097. DOI: 10.2140/ant.2022.16.1071. URL: https://doi.org/10.2140%2Fant.2022.16.1071.

[Dia18] Zhuo Diao. *On the vertex cover number of 3 uniform hypergraph.* 2018. arXiv: 1807.00473 [math.CO].

[DWZ22] Ran Duan, Hongxun Wu and Renfei Zhou. *Faster Matrix Multiplication via Asymmetric Hashing.* 2022. DOI: 10.48550/ARXIV.2210.10173. URL: https://arxiv.org/abs/2210.10173.

[EG16] Jordan Ellenberg and Dion Gijswijt. 'On large subsets of F_q^n with no three-term arithmetic progression'. In: *Annals of Mathematics* 185 (May 2016). DOI: 10.4007/annals.2017.185.1.8.

[Gal14] François Le Gall. 'Algebraic complexity theory and matrix multiplication'. In: *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*. ACM, July 2014. DOI: 10.1145/2608628.2627493. URL: https://doi.org/10.1145%2F2608628.2627493.

[Gow21] William T. Gowers. *The slice rank of a direct sum*. 2021. arXiv: 2105.08394 [math.CO].

[HL13] Christopher Hillar and Lek-Heng Lim. *Most tensor problems are NP-hard*. 2013. arXiv: 0911.1393 [cs.CC].

[LW22] Yun-Shan Lu and Hung-Lung Wang. *A note on the Tuza constant c_k for small k*. 2022. arXiv: 2206.14376 [math.CO].

[NS17] Eric Naslund and Will Sawin. 'Upper bounds for sunflower-free sets'. In: *Forum of Mathematics, Sigma* 5 (2017). DOI: 10.1017/fms.2017.12. URL: https://doi.org/10.1017%2Ffms.2017.12.

[Sch22] Alexander Schrijver. *On the Shannon capacity of sums and products of graphs*. 2022. DOI: 10.48550/ARXIV.2204.06853. URL: https://arxiv.org/abs/2204.06853.

[Shi17] Yaroslav Shitov. *A counterexample to Strassen's direct sum conjecture*. 2017. arXiv: 1712.08660 [math.CO].

[Str87] Volker Strassen. In: 1987.375-376 (1987), pp. 406–443. DOI: doi:10.1515/crll.1987.375-376.406. URL: https://doi.org/10.1515/crll.1987.375-376.406.

[Str69] Volker Strassen. 'Gaussian elimination is not optimal'. In: *Numerische Mathematik* 13 (Aug. 1969), pp. 354–356. DOI: 10.1007/BF02165411.

[Tao16] Terence Tao. *A symmetric formulation of the Croot-Lev-Pach-Ellenberg-Gijswijt capset bound*. May 2016. URL: https://terrytao.wordpress.com/2016/05/18/a-symmetric-formulation-of-the-croot-lev-pach-ellenberg-gijswijt-capset-bound/.

[TS16] Terence Tao and Will Sawin. *Notes on the "slice rank" of tensors*. Aug. 2016. URL: https://terrytao.wordpress.com/2016/08/24/notes-on-the-slice-rank-of-tensors/.

[Wil12] Virginia Vassilevska Williams. 'Multiplying matrices faster than coppersmith-winograd'. In: *Symposium on the Theory of Computing*. 2012.

[Zui18] Jeroen Zuiddam. 'Algebraic complexity, asymptotic spectra and entanglement polytopes'. PhD thesis. University of Amsterdam, 2018.

A Code

Code needed for both programs

```
import numpy as np
from scipy.optimize import minimize, NonlinearConstraint
from tabulate import tabulate

#Define the Full(d,q) tensors.
#The information we need is the block support, the value of each subtensor
#and the sizes of all the blocks
def fullsupport(n,q):
  support = []
  for i in range(n+1):
    for j in range(n+1):
      k = n-i-j
      if (i <= j and j <= k) or (i < k and j > k):
        if (j-i)%2 == 0 and (k-j)%2 == 0:
          support.append([i,j,k,1])
        else:
          support.append([i,j,k,q])

  sizes = [q,1]*int((n+1)/2)
  if n%2 == 0:
    sizes.insert(0,1)
  return (support,n), sizes

#Given a symmetric distribution on the symmetric block triples
#this function returns the marginal distribution
def marginals(x, distr, max):
  marginal = np.zeros(max+1)
  for triple in range(len(distr)):
    for block in range(3):
      marginal[distr[triple][block]] += x[triple]/3

  return marginal


#Function that computes the entropy given a distribution
def Entropy(marginaldistr):
  H = 0
  for p in marginaldistr:
    if p != 0:
      H -= p*np.log2(p)
  return H
```

Compute lower bounds $f_u(d,q)$

```
import numpy as np


# Compute the weighted entropy given a distribution on the support given by x
def weightedentropy(x, support):
  xdistr = marginals(x,support[0][0],support[0][1])
  return Entropy(xdistr) + sum([xdistr[i]*np.log2(support[1][i]) for i in range(len(xdistr))])

# Compute the log asymptotic slice rank by optimising the distribution on the block support
def logasr(support):
  distr = support[0]
```

```
  constraint = [{'type': 'ineq', 'fun': lambda x: x},
               {'type': 'eq', 'fun': lambda x: 1-sum(x)}]
  x0 = [1/len(distr[0])]*len(distr[0])

  result = minimize(lambda x : -weightedentropy(x, support),
                  x0, constraints=constraint, bounds = [(0,1)]*len(distr[0]))

  return (-result.fun, result.x)

#Compute the lower bound for omega based on the log asymptotic slice rank
#and the estimate for the border rank.
def omegau(rank, support):
  return 2*np.log2(rank)/logasr(support)[0]

#Allows us to output our results as a LaTeX table.
def outputlatextable(rows):
  print('Tabulate Table:')
  print(tabulate(rows, headers='firstrow', tablefmt='latex'))
  return

#Define the header row for our table.
rows = [['']]
for q in range(1,10):
  rows[0].append('q = ' + str(q) )

#Compute the table one row at a time.
for d in range(1,17):
  lowerbounds = ['d = '+ str(d) ]
  for q in range(1,10):
    supp = fullsupport(d,q)
    lowerbounds.append(omegau(sum(supp[1]),supp))

  rows.append(lowerbounds)

outputlatextable(rows)
```

Compute upper bounds $r(d, q)$

```

#Function that computes the Gamma term given a distribution.
def Gamma(x,distr,max):
  marg = marginals(x,distr,max)
  constraint3 = [{'type': 'ineq', 'fun': lambda z: z},
               {'type': 'eq', 'fun': lambda z: marginals(z,distr,max)-marginals(x,distr,max)}]
  z0 = x.copy()
  gamma = minimize(lambda z : -Entropy(z), z0, constraints=constraint3, bounds = [(0,1)]*len(distr))

  return -gamma.fun - Entropy(x)

#Function that computes the log of the value given the distribution, rho and
#the logvalues of the subtensors.
def logvalueproblem(x, distr, max, rho):
    z = []
    for block in range(len(distr)):
      if distr[block][0] == distr[block][1] and distr[block][1] == distr[block][2]:
        z.append(3*x[block])
      else:
        z.append(x[block])

```

```
    return Entropy(marginals(x, distr,max)) \
            + sum([z[i]*np.log2(distr[i][3])*rho/3 for i in range(len(distr))]) \
            - Gamma(x,distr,max)/2



# Compute the log of V_{3*rho}(T) given rho and a distribution.
def value(rho, distr):
  constraint = [{'type': 'ineq', 'fun': lambda x: x}, {'type': 'eq', 'fun': lambda x: 1-sum(x)}]
  x0 = [1/len(distr[0])]*len(distr[0])

  result = minimize(lambda x : -logvalueproblem(x, distr[0], distr[1], rho),
                    x0, constraints=constraint, bounds = [(0,1)]*len(distr[0]))

  return -result.fun




# The asymptotic rank for which we can establish $\omega \leq a$.
# is exactly equal to 2^{value(a,distr)}
def lasermethod(a, distr):
  return np.exp2(value(a,distr))


def fullsupport(n,q):
  support = []
  for i in range(n+1):
    for j in range(n+1):
      k = n-i-j
      if (i <= j and j <= k) or (i < k and j > k):
        if (j-i)%2 == 0 and (k-j)%2 == 0:
          support.append([i,j,k,1])
        else:
          support.append([i,j,k,q])

  sizes = [q,1]*int((n+1)/2)
  if n%2 == 0:
    sizes.insert(0,1)
  return (support,n), sizes



#Allows us to output our results as a LaTeX table.
def outputlatextable(rows):
  print(tabulate(rows, headers='firstrow', tablefmt='latex'))
  return

#Define the header row for our table.
rows = [['']]
for q in range(1,10):
  rows[0].append('q = ' + str(q) )

#Compute the table one row at a time.
for d in range(1,9):
  rank = ['d = '+ str(d) ]
  for q in range(1,10):
    supp = fullsupport(d,q)
    rank.append(lasermethod(2.37,supp[0]))
```

```

    rows.append(rank)

outputlatextable(rows)
```