A NEW FRAMEWORK FOR TENSOR PCA BASED ON TRACE INVARIANTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider the Principal Component Analysis (PCA) problem for tensors $\mathbf{T} \in (\mathbb{R}^n)^{\otimes k}$ of large dimension n and of arbitrary order $k \geq 3$. It consists in recovering a spike $\boldsymbol{v}_0^{\otimes k}$ (related to a signal vector $\boldsymbol{v}_0 \in \mathbb{R}^n$) corrupted by a Gaussian noise tensor $\mathbf{Z} \in (\mathbb{R}^n)^{\otimes k}$ such that $\mathbf{T} = \beta \boldsymbol{v}_0^{\otimes k} + \mathbf{Z}$ where β is the signal-to-noise ratio. In this paper, we propose a new framework based on tools developed by the theoretical physics community to address this important problem. They consist in trace invariants of tensors built by judicious contractions (extension of matrix product) of the indices of the tensor $\mathbf{T}$. Inspired by these tools, we introduce a new process that builds for each invariant a matrix whose top eigenvector is correlated to the signal for β sufficiently large. Then, we give examples of classes of invariants for which we demonstrate that this correlation happens above the best algorithmic threshold ($\beta \geq n^{k/4}$) known so far. This method has many algorithmic advantages: (i) it provides a detection algorithm linear in time and has only $\mathcal{O}(1)$ memory requirements (ii) the algorithms are very suitable for parallel architectures and have a lot of potential of optimization given the simplicity of the mathematical tools involved (iii) experimental results show an improvement of the state of the art for the symmetric tensor PCA. Furthermore, this framework allows more general applications by being able to theoretically study the recovery of a spike in the form of $\boldsymbol{v}_0 \otimes \cdots \otimes \boldsymbol{v}_k$ with different dimensions ($\mathbf{T} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k}$ with $n_1, \ldots, n_k \in \mathbb{N}$) as well as the recovery of a sum of different orthogonal spikes. We provide experimental results to these different cases that match well with our theoretical findings.

1 INTRODUCTION

Powerful computers and acquisition devices have made it possible to capture and store real-world multidimensional data. For practical applications (Kolda & Bader (2009)), analyzing and organizing these high dimensional arrays (formally called tensors) lead to the well known curse of dimensionality. Thus, dimensionality reduction is frequently employed to transform a high-dimensional data set by projecting it into a lower dimensional space while retaining most of the information and underlying structure. One of these techniques is Principal Component Analysis. PCA has made remarkable progress in a large number of areas thanks to its simplicity and adaptability (Jolliffe & Cadima (2016)).

In the Tensor PCA, as introduced by Richard & Montanari (2014), we consider a simple model where we attempt to detect and retrieve an unknown unit vector $\boldsymbol{v}_0$ from noise-corrupted multilinear measurements put in the form of a tensor $\mathbf{T}$. Using the notations found below, our model consist in:

$$\mathbf{T} = \beta \boldsymbol{v}_0^{\otimes k} + \mathbf{Z}, \tag{1}$$

with $\mathbf{Z}$ a pure Gaussian noise tensor of order k and dimension n with identically independent distributed (iid) standard Gaussian entries: $\mathbf{Z}_{i_1, i_2, \ldots, i_k} \sim \mathcal{N}(0, 1)$ and β the signal-to-noise ratio.

To solve this important problem, many methods have been proposed. However, practical applications require optimizable and parallelizable algorithms that are able to avoid the high computationally cost due to an unsatisfactory scalability of some of these methods. A summary of the time and space requirement of some existent methods can be found in Anandkumar et al. (2017).

One way to achieve this parallelizable algorithm is through methods based on tensor contractions (Kim et al. (2018)) which are extensions of the matrix product. These last years, tools based on tensor contractions have been developed by theoretical physicists where random tensors have emerged as a generalization of random matrices. We want to investigate the algorithmic threshold of tensor PCA and some of its variants using the theoretical physics approach in the hope that it leads to new insights and knowledge in tensor PCA.

Related work Tensor PCA was introduced by Richard & Montanari (2014) where the authors suggested and analyzed different methods to recover the signal vector like matrix unfolding and power iteration. Since then various other methods were proposed. Hopkins et al. (2015) introduced algorithms based on the sum of squares hierarchy with the first proven algorithmic threshold of $n^{k/4}$. However they generally require high computing resources and complex mathematics. Other studied methods have been inspired by different perspectives like homotopy in Anandkumar et al. (2017), statistical physics (Arous et al. (2020), Ros et al. (2019), Wein et al. (2019) and Biroli et al. (2020)), quantum computing (Hastings (2020)) as well as statistical query (Dudeja & Hsu (2020)).

Recently, a fundamentally different set of mathematical tools that have been developed for tensors in the context of high energy physics have been used to approach the problem. They consist in trace invariants of degree $d,\ d \in \mathbb{N}$, obtained by contracting pair of indices of d copies of the tensor $\mathbf{T}$. They have been used in Evnin (2020) to study the highest eigenvalue of a real symmetric Gaussian tensor. Subsequently, Gurau (2020) provided a theoretical study on a function based on an infinite sum of these invariants. Their results suggest a transition phase for the highest eigenvalue of a tensor for β around $n^{1/2}$ in a similar way to the BBP transition in the matrix case (Baik et al. (2005)). Thus, this function allows the detection of a spike. However evaluating the function involves an computing an integral over a n-dimensional space, which may not be possible in a polynomial time.

The contribution of this paper is the use of these invariant tools to build tractable algorithms with polynomial complexity. In contrast to Gurau (2020), instead of using a sum of an infinite number of invariants, we select one trace invariant with convenient properties to build our algorithms. It lets us detect the presence of the signal linearly in time and with a space requirement in $\mathcal{O}(1)$. Moreover, in order to recover the signal vector besides simply detecting it, we introduce new tools in the form of matrices associated to this specific invariant. Within this framework, we show as particular cases, that the two simpler graphs (of degree two) are equivalent to the tensor unfolding and to homotopy algorithm (which is equivalent to average gradient descent). These two algorithms are the main practical ones known from the point of view of space and time requirement (Anandkumar et al. (2017) provides a table comparison). Interestingly, our framework allows us to go further by using graphs of higher degrees and we can see in that case that the tetrahedral graph (degree four) performs better in the symmetric case. We should notice that the complexity of the algorithm does not necessarily increase with the degree of the graph (unlike sum of squares hierarchy).

Notations We use bold characters $\mathbf{T}, \boldsymbol{M}, \boldsymbol{v}$ for tensors, matrices and vectors. $[p]$ denotes the set $\{1, \dots, n\}$. A real $p-$th order tensor $\mathbf{T} \in \bigotimes_{i=1}^{p} \mathbb{R}^{n_i}$ is a member of the tensor product of Euclidean spaces $\mathbb{R}^{n_i}, i \in [p]$. It is symmetric if $\boldsymbol{T}_{i_1 \dots i_k} = \boldsymbol{T}_{\tau(i_1) \dots \tau(i_k)}\ \forall \tau \in \mathfrak{S}_k$. For a vector $\mathbf{v} \in \mathbb{R}^n$, we use $\boldsymbol{v}^{\otimes p} \equiv \boldsymbol{v} \otimes \mathbf{v} \otimes \dots \otimes \boldsymbol{v} \in \bigotimes^{p} \mathbb{R}^n$ to denotes its p-th tensor power.

Let's define the operator norm, which is equivalent to the highest eigenvalue of a tensor of any rank: $\|\mathbf{X}\|_{\text{op}} \equiv \max \{\mathbf{X}_{i_1,\dots,i_k} (\boldsymbol{w}_1)_{i_1} \dots (\boldsymbol{w}_k)_{i_k}, \forall i \in \{1, \dots, n\}, \|\boldsymbol{w}_i\| \leq 1\}$ Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ be a matrix and $\lambda_1, \dots, \lambda_n$ its (real or complex) eigenvalues. The spectral radius of $\boldsymbol{A}$ is the highest eigenvalue in absolute value. In particular, when a matrix N is normal (i.e., it commutes with its conjugate transpose), we have the result $\rho(\boldsymbol{N}) = \|\boldsymbol{N}\|_{\text{op}}$. Thus, in order to compute the operator norm of a given matrix $\boldsymbol{A}$, we can determine the spectral radius of the matrix $\boldsymbol{A}^* \boldsymbol{A}$ and take its square root since $\|\boldsymbol{A}^* \boldsymbol{A}\|_{\text{op}} = \|\boldsymbol{A}\|_{\text{op}}^2$.

We will use the Einstein summation convention, when an index variable appears twice in a single term and is not otherwise defined, it implies summation of that term over all the values of the index. We denote the expectation of a variable X by $\mathbb{E}(X)$ and its variance by $\sigma(X)$.

2 GENERAL FORMALISM FOR DETECTION AND RECOVERY OF THE SIGNAL IN THE LARGE N LIMIT

2.1 WHAT DO WE USE TO STUDY THE SIGNAL?

An important concept in problems involving matrices is the spectral theory. It refers to the study of eigenvalues and eigenvectors of a matrix. It is of fundamental importance in many areas. In machine learning, the matrix PCA computes the eigenvectors and eigenvalues of the covariance matrix of the features to perform a dimensional reduction while ensuring most of the key information is maintained. In this case, the eigenvalues is a very efficient tool to describe data variability. In the case of signal processing, eigenvalue can contain information about the intensity of the signal, while the eigenvector points out to its direction. Lastly, a more theoretical example involves quantum physics where the spectrum of the matrix operator is used to calculate the energy levels and the state associated.

In all of these examples, an important property of the eigenvalues of a matrix $\boldsymbol{M}$ is its invariance under orthogonal transformations $\boldsymbol{M} \to \boldsymbol{O}\boldsymbol{M}\boldsymbol{O}^{-1}$. Since these transformations essentially just rotate the basis to define the coordinate system, they must not affect intrinsic information like data variability, signal intensity or the energy of a system. The eigenvalues are able to capture some of these inherent information, but recovering the complete general information requires computing their respective eigenvectors (for example to find the principal component, the direction of the signal or the physical state). There is more such invariants than eigenvalues. Another important set worth mentioning are the traces of the n first matrix powers $\mathrm{Tr}(A), \mathrm{Tr}\big(A^2\big), \ldots, \mathrm{Tr}(A^n)$. Obtaining them is slightly uses slightly different methods than eigenvalues, but they contain the same information since each set can be inferred from the other through some basic algebraic operations.

On the basis of the matrix case, we expect that tensor invariants under orthogonal transformations should capture similar intrinsic information like the intensity of the signal, and conceivably, related objects able to indicate direction of the signal. However the concept of eigenvalue and eigenvector is ill defined in the tensor case and not practical giving that the number of eigenvalues is exponential with the dimension n (Qi (2005), Cartwright & Sturmfels (2013)) and the complexity of computing them. In contrast, we have a very convenient generalization of the traces of the power matrices for the tensors that we call trace invariants, obtained by contracting (equivalent to a matrix multiplication) an even number of copies of the tensor to obtain a scalar. They have been extensively studied during the last years in the context of high energy physics and many important properties have been proven. An invariant is called of degree d if it consists in contraction of d copies of $\mathbf{T}$. We present here two different classes of invariants:

Invariants under $\mathrm{O}(n)^k$: A tensor T transforms under the group $\bigotimes_{i=1}^{k} \mathrm{O}(n_i)$ acts on

$$\mathbf{T}_{a_j^1 \ldots a_j^k} \to O^{(1)}_{a_j^1 b_j^1} \ldots O^{(k)}_{a_j^k b_j^k} \mathbf{T}_{b_j^1 \ldots b_j^k} \quad \text{for } \boldsymbol{O}^{(i)} \in \mathrm{O}(n_i) \;\; \forall i \in [k]. \tag{2}$$

Invariants under $\mathrm{O}(n)$: A tensor T transforms under the group $\mathrm{O}(n)$ (Carrozza & Tanasa (2016)) acts on

$$\mathbf{T}_{a_j^1 \ldots a_j^k} \to O_{a_j^1 b_j^1} \ldots O_{a_j^k b_j^k} \mathbf{T}_{b_j^1 \ldots b_j^k} \quad \text{for } \boldsymbol{O} \in \mathrm{O}(n). \tag{3}$$

A trace invariant of degree d of a tensor $\mathbf{T}$ of order k admit a practical graphical representation as edge colored graphs. We draw d vertices representing the d different copies of $\mathbf{T}$. The indices of each copy is represented by k half-edges with a different color for each index position as shown in Figure 1a. When two different indices are contracted, we connect their corresponding half-edges. Figure 1b and 1c illustrate $\mathrm{O}(n)^k$ invariants while Figure 1d is an example of an $\mathrm{O}(n)$ invariant. Avohou et al. (2019) provides a throrough study about their number.

Gaussian expectation of trace invariants:

A very useful asset of these invariants is that for tensors whose components are Gaussian, we are able to compute their expectation using simple combinatory (Gurau (2017)). In the case of a $\mathcal{O}(n)$ invariant, it reduces the following expression where $F(\mathcal{G})$ denotes the number of connected subgraphs of only two colors of $\mathcal{G}$.

$$\langle \mathrm{Tr}_{\mathcal{B}}(\mathbf{T}, \bar{\mathbf{T}}) \rangle = \sum_{\mathcal{G}, \mathcal{G} \backslash \mathcal{E}^0(\mathcal{G}) = \mathcal{B}} n^{F(\mathcal{G}) - F(\mathcal{B})} = \sum_{\mathcal{G}, \mathcal{G} \backslash \mathcal{E}^0(\mathcal{G}) = \mathcal{B}} n^{\sum_c F^{0,c}(\mathcal{G})}. \tag{4}$$

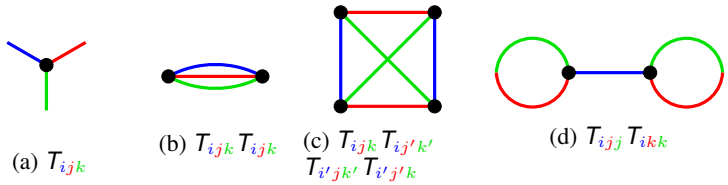


Figure 1: Example of graphs and their associated invariants

2.2 What do we use to recover the signal?

As previously mentioned in Section 2.1, an invariant should be able to detect a signal. But in order to recover it, we should find mathematical objects that are able to provide a vector. To this effect, we introduce in this paper a new set of tools in the form of matrices. They are obtained by cutting an edge e of the graph, which amounts to not summing over the two indices associated to the two half-edges of the open edge and using them instead to index the matrix. This results on a matrix that we denote $\boldsymbol{M}_{\mathcal{G},e}$. We will drop the index of the matrix when the context is clear. An example of such a matrix is given in Figure 2 in the case of a melon.

2.3 Phase Transition within this Formalism

We can represent the tensor from which we hope to extract the signal represented graphically as:

$$T_{ij_1\dots j_{k-1}} \quad = \quad \sqrt{N}\beta v_i v_{j_1}\dots v_{j_{k-1}} \quad + \quad Z_{ij_1\dots j_{k-1}}$$

This decomposition leads us naturally to decompose in a similar way a tensor trace invariant $I_{\mathcal{G}}(\mathbf{T})$ into two parts, separating a first contribution $I_{\mathcal{G}}^{(N)}(\mathbf{T})$ associated to the pure noise tensor and a second part $I_{\mathcal{G}}^{(S)}(\mathbf{T})$ enclosing the additional contributions resulting from the addition of a signal tensor.

$$I_{\mathcal{G}} = I_{\mathcal{G}}^{(N)} + I_{\mathcal{G}}^{(S)}. \tag{5}$$

We consider a graph $\mathcal{G}$ and its tensor trace invariant $I_{\mathcal{G}}$. Let's denote $I'_{\mathcal{G}}$ the invariant associated to the subgraph obtained by removing from $\mathcal{G}$ the edge e and its two vertices. We can distinguish three kind of contributions:

$$\begin{aligned}
\mathbf{M}_{\mathbf{i}_1\mathbf{i}_2} &= \mathbf{M}^{(N)}_{\mathbf{i}_1\mathbf{i}_2} + \mathbf{M}^{(D)}_{\mathbf{i}_1\mathbf{i}_2} + \mathbf{M}^{(R)}_{\mathbf{i}_1\mathbf{i}_2} \\
&= \mathbf{Z}_{\mathbf{i}_1..}\mathbf{Z}_{\mathbf{i}_2..}\dots + \beta\,\mathbf{Z}_{\mathbf{i}_1..}\mathbf{Z}_{\mathbf{i}_2..}\dots + (\mathbf{v}_{\mathbf{i}_1}\mathbf{Z}_{\mathbf{i}_2\dots..} + \mathbf{v}_{\mathbf{i}_2}\mathbf{Z}_{\mathbf{i}_1\dots..}) + \mathbf{v}_{\mathbf{i}_1}\mathbf{v}_{\mathbf{i}_2}
\end{aligned}$$

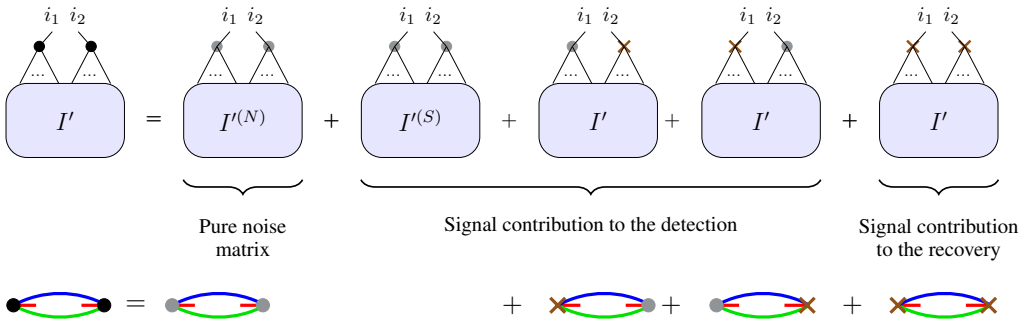


Figure 2: Decomposition of a matrix graph and the melon exemple

Lemma 1. $\mathbb{E}(\boldsymbol{M}^{(N)}) = \frac{\mathbb{E}(I_{\mathcal{G}}^{(N)})}{n}\boldsymbol{I}_n$.

Calculating the operational norm of a matrix The trace of the power r of a matrix $\text{Trace}(M^r)$ can be represented as the graph gluing open edges of $2r$ subgraphs $\mathcal{G}_i$ representing one matrix A where $M = A^\top A$, we denote this graph by $\mathcal{G}$. In case we remove the expectation of the matrix, it comes back to the fact that no subgraph $\mathcal{G}_i$ can have only internal propagators.

For a general graph, we identify three possible phases in function of which matrix operational norm dominates:

- **No recovery, no detection:** $\boldsymbol{M}^{(N)} - \mathbb{E}(\boldsymbol{M}^{(N)})$ dominating.
  We can't distinguish if there is a signal. It is for example the phase for $\beta \to 0$. We removed the expectation of $\boldsymbol{M}^{(N)}$ since it is proportional to the identity.
- **Detection but no recovery:** $\boldsymbol{M}^{(D)}$ dominating
  We can detect the presence of the signal (thanks to the highest eigenvalue) but we can't recover the signal vector since the eigenvector associated to the maximal eigenvalue is not correlated to the signal vector
- **Detection and recovery:** $\boldsymbol{M}^{(R)}$ dominating
  We recover the signal vector. It is for example the phase for $\beta \to \infty$

2.4 Algorithmic Threshold for a General Graph

Algorithm 1: Detection algorithm associated to the graph $\mathcal{G}$ and edge e

Input: The tensor $\mathbf{T} = \beta \boldsymbol{v}^{\otimes 3} + \mathbf{Z}$
Goal: Detection of $\boldsymbol{v}$
1 Calculate the scalar $\alpha = I_{\mathcal{G}}(\mathbf{T})$;
2 Compare α with the variance $\sigma(I_{\mathcal{G}}^{(N)}(\mathbf{T}))$.
Result: Detecting the presence of a spike

Theorem 2. *Let $\mathcal{G}$ be a graph of degree d, $\exists\ \beta_{det} > 0$ so that Algorithm 1 detects the presence of a signal for $\beta \geq \beta_{det}$.*

Algorithm 2: Recovery algorithm associated to the graph $\mathcal{G}$ and edge e

Input: The tensor $\mathbf{T} = \beta \boldsymbol{v}^{\otimes 3} + \mathbf{Z}$
Goal: Estimate $\boldsymbol{v}$
1 Calculate the matrix $\boldsymbol{M}_{\mathcal{G},e}(\mathbf{T})$;
2 Compute its top eigenvector by matrix power iteration.
Result: Obtaining an estimated vector $\mathbf{v}$

Theorem 3. *Let $\mathcal{G}$ be a graph of degree d, $\exists\ \beta_{rec} > 0$ so that Algorithm 2 gives an estimator $\boldsymbol{v}$ so that **Loss**$(\boldsymbol{v}, \boldsymbol{v}_0) = o(1)$ for $\beta \geq \beta_{rec}$.*

Theorem 4. *Let $k \geq 3$. It is impossible to detect or recover the signal below the threshold $\beta \leq n^{k/4}$ which is the minimal variance of any graph.*

3 Some important examples of graphs

3.1 Tensor invariant based on the perfect one-factorizations graph

We use this example because it is simple combinatorially speaking and we believe that the method used for the demonstration could be standard to many others. It is a generalization of the tetrahedral graph.

Lemma 5. *If $\beta_{det} = \mathcal{O}(n^{3/4})$, the intermediate matrices and the intermediate graphs are never dominant.*

Theorem 6. *The complete graph is able to recover the signal vector for $\beta \geq n^{k/4}$.*

3.2 THE MELON GRAPH BETTER KNOWN AS TENSOR UNFOLDING

Let's consider the invariant $T_{i_1 \dots i_n} T_{i_1 \dots i_n}$ which is illustrated by the graph in Fig. 1b when $k = 3$. Studying the matrix obtained by cutting any of its edges corresponds to the tensor unfolding method where we use a matrix $n \times n^{k-1}$ that is denoted $\text{Mat}_1(\mathbf{T})$ using Richard & Montanari (2014) notations. We can prove a conjuncture that this algorithm works also for the symmetric case.

Theorem 7. *The algorithms 1 and 2 works for the melon with $\beta_{det} = \beta_{det} = n^{k/4}$ in linear time and respectively $\mathcal{O}(n^2)$ and $\mathcal{O}(1)$ (computing an invariant which is a scalar) memory requirement.*

3.3 THE TADPOLE GRAPH

Figure 1d has a special characteristic: we can obtain two disconnected parts by cutting only one line. These disconnected parts are called by physicists tadpole diagram. This characteristic has as a consequence the fact that cutting that edge build a matrix of rank one in the form of $\boldsymbol{v}\boldsymbol{v}^T$. The obtained vector is not completely correlated to the signal vector however it allows tensor power iteration (conjecturally) to recover it. It is a variant of algorithms introduced from different perspectives: homotopy and statistical physics.

Theorem 8. *The tadpole graph allows to recover the signal vector by using local algorithms to enhance the signal contribution to the vector.*

3.4 HANDLING A SPIKE WITH DIFFERENT DIMENSIONS

We take $k = 3$ and we consider the more general case where the tensor has axes of different dimensions n_i where $n_i \in \mathbb{N}$. We can assume without any loss of generality that $n_1 \geq n_2 \geq n_3$.

$$\mathbf{T} = \beta \mathbf{v_1} \otimes \mathbf{v_2} \otimes \mathbf{v_3} + \mathbf{Z} \quad \text{where} \quad \mathbf{v_i} \in \mathbb{R}^{n_i}, \quad n_i \in \mathbb{N}. \tag{6}$$

Reiterating the previous demonstration, we obtain that the threshold in this case is $\max\left((n_1 n_2 n_3)^{1/4}, n_1^{1/2}\right)$.

Theorem 9. *Assume $n_1 \geq n_2 \geq n_3$ without any loss of generality. Using the melon graph, the threshold for $\boldsymbol{v}_1$ is given by $\max\left((n_1 n_2 n_3)^{1/4}, n_1^{1/2}\right)$ while the thresholds for $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ are equal to $(n_1 n_2 n_3)^{1/4}$.*

3.5 HANDLING MULTIPLE ORTHOGONAL SPIKES

We take $k = 3$ and we consider a symmetric tensor with multiple orthogonal spikes.

$$\mathbf{T} = \sum_{i=1}^{p} \beta_i \mathbf{v_i}^{\otimes k} + \mathbf{Z} \quad \text{where} \quad \langle \mathbf{v_i}, \mathbf{v_j} \rangle = 0 \quad \forall i \neq j. \tag{7}$$

Theorem 10. *If p is constant compared to n, we can recover all the spikes using the melonic diagram followed by power iteration and deflation. These last two steps are needed in the case that the intensity of the spikes are similar which will prevent us from directly recovering the signal vectors. The experimental results suggest that it works for p up to n.*

4 NUMERICAL EXPERIMENTS

4.1 COMPARISON OF RECOVERY METHODS

We begin by comparing our results to the state of art in both the symmetric case and the asymmetric one. We take $n = 100$ for different values of β. As in Richard & Montanari (2014), for each algorithm we use two variants: the simple algorithm and the algorithm called "with warm start", this last one consists in taking the result of the simple algorithm and performing tensor power iteration on it. The figure 3 shows that the comparison is non trivially related to the symmetry of the tensor. Both algorithms perform equivalently asymptotically but our algorithm gives better results for β close to the threshold in the symmetric case.

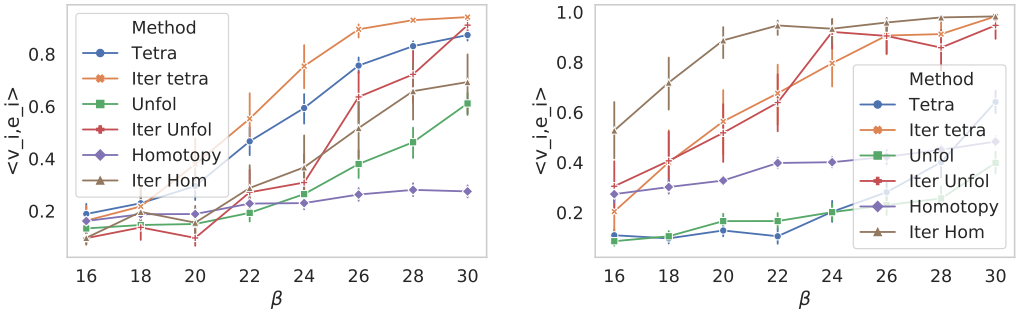
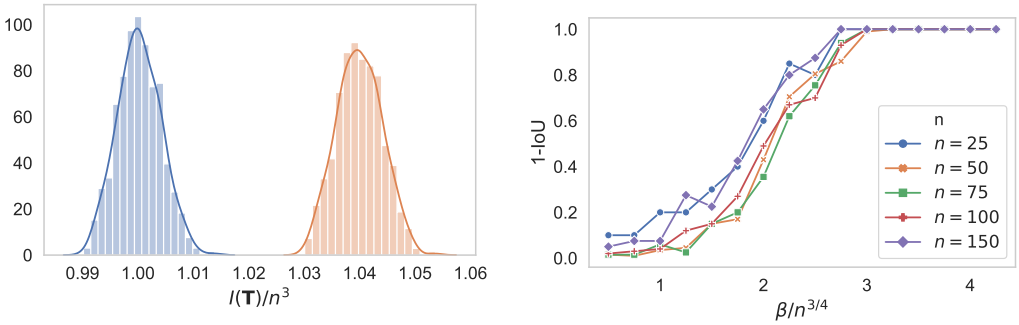


Figure 3: Comparison of different methods ($n = 100$). Left: symmetric case, right: asymmetric.

4.2 Detection

In Figure 4a, we plot the distribution of the melonic invariant with and without a signal. We see that for a β large enough, the two distributions become disjoint. In Figure 4b we use the Intersection over Union (IoU, the cardinal of the intersection of the distributions over their unions) metric and plot 1-IoU in function of $\beta/n^{3/4}$ which proves that it is the right asymptotic behavior.



(a) Melon distribution for detection (n=50). Blue without signal and orange with signal

(b) Detection with the melon graph.

Figure 4: Detection using the melon graph

4.3 A spike with different dimensions

We see in the left figure that the threshold $(n_1 n_2 n_3)^{1/4}$ matches perfectly with the experiences when $n_3 < n_1.n_2$. We also see in the right figure that when $n_3 > n_1.n_2$ the recovery of v_3 has a different asymptotic behavior than v_1 and v_2 (it becomes $n_3^{1/2}$ while the threshold for v_1 and v_2 stays at $(n_1 n_2 n_3)^{1/4}$), corresponding to what our theoretical study predicted.

4.4 Multiple orthogonal spikes

In this subsection, we take $n = 50$ and we try to recover the sum of 50 orthogonal spikes. The figure 6 shows that it works well. We performed two experiments. In the first we fix n and the number of spikes n_{spik} and vary β. We note the correlation found for every spike. In the second experiment we fixed n and β and varied the method and the number of spikes. We can notice two phases, for small number of spikes the iterative tetrahedral performs as well as the tetrahedral but then behaves slightly the random power iteration when the number of spikes increase. We obtain an empirical threshold $\beta = n^{3/4}$ for any number of spikes.

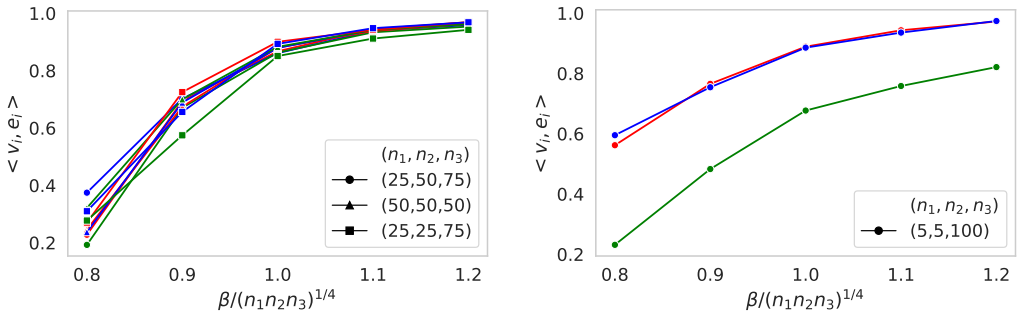
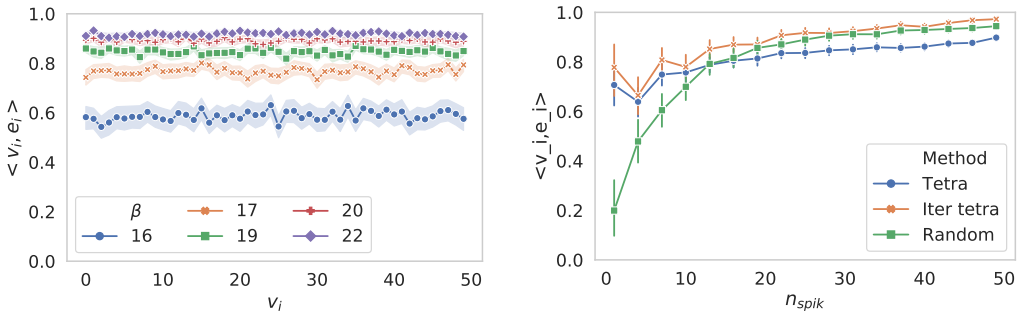


Figure 5: Spike with vectors of different dimensions (n_1 in blue, n_2 in red and n_3 in green)



(a) Recovery of a sum of 50 orthogonal spikes for different β

(b) Average of the recovery of the spikes depending on the number of spikes for $\beta = 18$

Figure 6: Recovery of a sum orthogonal spikes for $n = 50$ in the symmetric case

5 Conclusion

In this paper we introduced a novel framework for the tensor PCA based on trace invariants. Within this framework, we provide different algorithms to detect a signal vector or recover it. These algorithms use tensor contractions that has a high potential of parallelization and computing optimization. We illustrate the practical pertinence of our framework by presenting some examples of algorithms and prove their ability to detect and recover a signal vector linearly in time for β above the optimal algorithmic threshold. Note that, one of these detection algorithms requires only $\mathcal{O}(1)$ memory requirement which could be advantageous in some applications. Our experimental results show that the tetrahedral graph performs better than the the state of the art in the symmetric case. Moreover, we also show that two well known algorithms (Homotopy and Tensor Unfolding) can be mapped to our framework and result to simpler graph (e.g. the melonic graph). Interestingly, our framework is also able to extend the theoretical and practical study of tensor PCA to new and less restrictive situations like data where the dimensions of the axes are different or a sum of orthogonal spikes. One important direction of future research is to explore the potential of more general graphs. Indeed, our study gives some important insight by suggesting that the most efficient algorithm depends on the characteristics of the data (eg symmetry of the tensor).

References

Anima Anandkumar, Yuan Deng, Rong Ge, and Hossein Mobahi. Homotopy analysis for tensor pca. In *Conference on Learning Theory*, pp. 79–104. PMLR, 2017.

Gerard Ben Arous, Reza Gheissari, Aukosh Jagannath, et al. Algorithmic thresholds for tensor pca. *Annals of Probability*, 48(4):2052–2087, 2020.

Remi C. Avohou, Joseph Ben Geloun, and Nicolas Dub. On the counting of $O(N)$ tensor invariants. *arXiv e-prints*, art. arXiv:1907.04668, July 2019.

Jinho Baik, Gérard Ben Arous, Sandrine Péché, et al. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005.

Giulio Biroli, Chiara Cammarota, and Federico Ricci-Tersenghi. How to iron out rough landscapes and get optimal performances: averaged gradient descent and its application to tensor PCA. *Journal of Physics A Mathematical General*, 53(17):174003, May 2020.

Sylvain Carrozza and Adrian Tanasa. O (n) random tensor models. *Letters in Mathematical Physics*, 106(11):1531–1559, 2016.

Dustin Cartwright and Bernd Sturmfels. The number of eigenvalues of a tensor. *Linear algebra and its applications*, 438(2):942–952, 2013.

Rishabh Dudeja and Daniel Hsu. Statistical Query Lower Bounds for Tensor PCA. *arXiv e-prints*, art. arXiv:2008.04101, August 2020.

Oleg Evnin. Melonic dominance and the largest eigenvalue of a large random tensor. *arXiv preprint arXiv:2003.11220*, 2020.

Frank Ferrari, Vincent Rivasseau, and Guillaume Valette. A New Large N Expansion for General Matrix–Tensor Models. *Commun. Math. Phys.*, 370(2):403–448, 2019.

R. Gurau. *Random Tensors*. Oxford University Press, 2017.

Razvan Gurau. Universality for Random Tensors. *Ann. Inst. H. Poincare Probab. Statist.*, 50(4): 1474–1525, 2014.

Razvan Gurau. On the generalization of the wigner semicircle law to real symmetric tensors. *arXiv preprint arXiv:2004.02660*, 2020.

Matthew B. Hastings. Classical and Quantum Algorithms for Tensor Principal Component Analysis. *Quantum*, 4:237, February 2020.

Samuel B. Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-squares proofs. *CoRR*, abs/1507.03269, 2015.

I. T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374, 2016.

Jinsung Kim, Aravind Sukumaran-Rajam, Changwan Hong, Ajay Panyala, Rohit Kumar Srivastava, Sriram Krishnamoorthy, and Ponnuswamy Sadayappan. Optimizing tensor contractions in ccsd (t) for efficient execution on gpus. In *Proceedings of the 2018 International Conference on Supercomputing*, pp. 96–106, 2018.

T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51:455–500, 2009.

Liqun Qi. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation*, 40(6): 1302 – 1324, 2005.

Emile Richard and Andrea Montanari. A statistical model for tensor pca. In *Advances in Neural Information Processing Systems*, pp. 2897–2905, 2014.

Valentina Ros, Gerard Ben Arous, Giulio Biroli, and Chiara Cammarota. Complex energy landscapes in spiked-tensor and simple glassy models: Ruggedness, arrangements of local minima, and phase transitions. *Physical Review X*, 9(1):011003, 2019.

G. W. Stewart and Ji guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

David G. Wagner. On the perfect one—factorization conjecture. *Discrete Mathematics*, 104(2):211 – 215, 1992.

Alexander S Wein, Ahmed El Alaoui, and Cristopher Moore. The kikuchi hierarchy and tensor pca. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 1446–1468. IEEE, 2019.

A USEFUL THEOREMS

The following theorems are needed to the demonstration of the proposed theorems.

- **Wedin perturbation**: Let $\mathbf{M} = \beta \mathbf{v_0} \mathbf{v_0}^\top + \mathbf{N}$ a square matrix with $\|\mathbf{v_0}\| = 1$. Let $\mathbf{v}$ denote the eigenvector with the largest eigenvalue of $\mathbf{M}$. If $\beta > 2\|\mathbf{N}\|_{\text{op}}$ then:

$$\mathbf{Loss}(\mathbf{v}, \mathbf{v_0}) \leq \frac{8\|\mathbf{N}\|_{\text{op}}}{\beta^2}. \tag{8}$$

- **Chebyshev inequality**: Let X be a random variable with finite expected value μ and finite non-zero variance σ^2. Then for any real number $C > 0$,

$$\text{Pr}(|X - \mu| \geq C\sigma) \leq \frac{1}{C^2}. \tag{9}$$

- **Theorem (Gelfand's Formula; 1941):** For any matrix norm $\|.\|$, we have:

$$\rho(A) = \lim_{s \to \infty} \|A^s\|^{\frac{1}{s}}. \tag{10}$$

  In particular we can choose the Frobenius norm:

$$\|A\|_{\text{F}} = \sqrt{\text{trace}(A^* A)}. \tag{11}$$

B GAUSSIAN MEASURE

We will follow a similar approach to the one in Gurau (2017) to compute the expectations of trace invariants of a random tensor $\mathbf{T}$ whose components are normally distributed. We focus on the case $k = 3$.

By the definition of the Gaussian measure we have the equation:

$$\mathbb{E}(\text{Tr}_{\mathcal{B}}(\mathbf{T})) = \sum_a \delta^{\mathcal{B}}_{aa} \sum_{\tau^{(0)} \in \mathfrak{S}(k)} \prod_{j=1}^{k} \delta_{a_j{}^{\mathcal{D}} a_{j^{\mathcal{D}}_{\tau^0(j)}}}, \tag{12}$$

where $\sum_a$ indicates a sum over all the indices, and $\delta^{\mathcal{B}}_{aa}$ indicates that contracted indices (illustrated by an edge) of the tensors have to be equal.

We introduce the concept of covering graph used in Gurau (2014): a covering graph of $\mathcal{B}$ consists in adding edges of color 0 (also called propagators).

In the case of $\mathcal{O}(n)^k$ invariants, where the edges could only have one color, we have the formula:

$$\begin{aligned} \mathbb{E}(\text{Tr}_{\mathcal{B}}(\mathbf{T})) = \sum_{\mathcal{G}, \mathcal{G} \backslash \mathcal{E}^0(\mathcal{G}) = \mathcal{B}} \sum_{aa} \left(\prod_{c=1}^{3} \prod_{e^c = (v_j, v_k) \in \mathcal{E}^c(\mathcal{G})} \delta_{a_j{}^c, a_k{}^c} \right) \\ \left(\prod_{e^0 = (v_p, v_q) \in \mathcal{E}^0(\mathcal{G})} \delta_{a_p{}^1, a_q{}^1} \delta_{a_p{}^2, a_q{}^2} \delta_{a_p{}^3, a_q{}^3} \right), \end{aligned} \tag{13}$$

giving

$$\mathbb{E}(\text{Tr}_{\mathcal{B}}(\mathbf{T})) = \sum_{\mathcal{G}, \mathcal{G} \backslash \mathcal{E}^0(\mathcal{G}) = \mathcal{B}} n^{F(\mathcal{G}) - F(\mathcal{B})} = \sum_{\mathcal{G}, \mathcal{G} \backslash \mathcal{E}^0(\mathcal{G}) = \mathcal{B}} n^{\sum_c F^{0,c}(\mathcal{G})}. \tag{14}$$

An edge of color $1, 2, 3$ identify one index of appropriate color, while edges of color 0 identify all the indices of their two end vertices. So an index a^c is identified along an edge of color c, then along an edge of color 0, then along an edge of color c and so on, until the cycle of colors $(0, c)$ closes. Thus when the graph is complete, we obtain a free sum for every face of colors $(0, c)$, which gives us equation 14.

Example: Let's calculate the expectation of the tetrahedral graph $\mathcal{B}$:

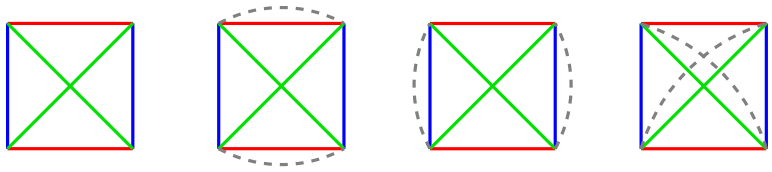


Figure 7: Tetrahedral and its covering graph

For every covering graph, there is one color having two faces of two colors having only one face. Hence the expectation of the tetrahedral graph is given by $3N^4$.

B.1 THE CASE OF EDGES OF MIXED COLOR

In the case of $\mathcal{O}(n)$ invariants, an edge can have two half edges of different colors. The propagators (edges of color 0) still identify all the indices of their two end vertices. In this case, the cycle along which the index is identified to end in a free sum is different. Indeed an index a is identified along a regular edge where it can change of color from a half edge to another, then it is identified along a propagator where it conserves its color, and so on until the cycle closes.

B.2 THE CASE OF A GRAPH WITH AN OPEN EDGE

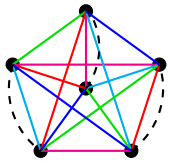
We encounter this case when we want to study the statistics of a matrix built out of a trace invariant graph. Indeed cutting an edge of color c replaces the contraction of two indices and leaves the two indices of the open edge as variables. We name them i_1 and i_2. Then in the formula $\left(\prod_{c=1}^{3}\prod_{e^c=(v_j,v_k)\in\mathcal{E}^c(\mathcal{G})}\delta_{a_j^c,a_k^c}\right)$ we replace $\delta_{a_j^c,a_k^c}$ by $\delta_{a_j^c,i_1}\delta_{i_2,a_k^c}$ with i_1 and i_2 fixed. Thus if a face of color $0,c$ contains only one of the two vertices of the edge (if the graph is connected, this only happens when we calculate the variance), its contribution is equal to one. If it contains the two vertices, it is equal to one if $i_1 = i_2$ and zero otherwise.

B.3 THE CASE OF A GRAPH WITH A MIX OF SPIKES AND RANDOM TENSORS

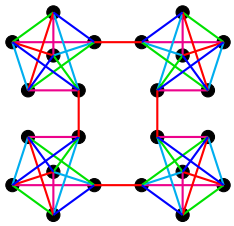
We encounter this case when we want to calculate the contribution of the signal to an invariant, which lead us to replace a vertex by the tensorial product of our signal vector. Since we are interested by an expectation, and knowing that the gaussian distribution is invariant by rotation, we may as well take the signal vector as the unit vector $(1, 0, \ldots, 0)$. Then in the formula $\left(\prod_{c=1}^{3}\prod_{e^c=(v_j,v_k)\in\mathcal{E}^c(\mathcal{G})}\delta_{a_j^c,a_k^c}\right)$ we replace $\delta_{a_j^c,a_k^c}$ by $\delta_{a_j^c,0}$ and we restrict the propagators to the tensors vertices. First, we notice that if there is an odd number of spikes, the expectation will be equal to zero. Furthermore, there is no free sum if the index identified in a cycle ends up with a contraction with a spike. Its contribution is equal to one.

C THE PERFECT ONE-FACTORIZATION GRAPH

In the mathematical field of graph theory, a spanning subgraph of a graph $\mathcal{G}$ is another graph that has all the vertices of $\mathcal{G}$ but only a subset of its edges. It is called a one-factor when all the vertices are connected to only one edge. A perfect one-factorization happens when we can find a partition of the edges into one graphs where every pair of one factors induce a connected cycle in $\mathcal{G}$. In other words, restricting to two colors we should obtain one connected subgraph. It is conjectured Wagner (1992) that they exist $\forall n$ but they were proven only for prime numbers and up to 56. We use this example because it is simple combinatorially speaking and we believe that the method used for the demonstration could be standard to many others. At least it shows well how it is done. it is a good example. They were studied in the context of random tensors in Ferrari et al. (2019).



(a) Complete graph for $k = 5$



(b) Graph representing $\mathrm{Tr}\left((A^T A)^2\right)$

Figure 8: Complete graph for $k = 5$

D PROOFS

Proof Lemma 1 Let $\boldsymbol{M}$ be the matrix obtained by cuting the edge e of a trace invariant graph. The non diagonal components $M_{ij},\ i \neq j$ have a null expectation since the cycle involving the edge e will identify i and j with a $\delta_{ij} = 0$. By invariance, $\mathbb{E}(M_{ii})$ are equal for all i. Then, by linearity of the expectation, $\mathbb{E}(M_{ii}) = \mathbb{E}(\mathrm{Tr}(\boldsymbol{M}))/n = \mathbb{E}(\mathcal{G})/n$.

Proof theorem 2 Let $n \in \mathbb{N}$ and $\mathcal{G}$ a graph of degree d. When $\beta \to \infty$, the dominant graph will be the one where all the tensors are replaced by the signal spike, with a value of β^d. The subdominant ones will be the graphs where all but one tensors vertices have been replaced by a spike. Thus it is easy to show that $\mathbb{E}(I_{\mathcal{G}}(T)) \to \beta^d$ while $\sigma(I_{\mathcal{G}}(T)) \sim \beta^{(d-1)}$. Then, using Chebyshev's inequality demonstrates that the distributions of $I_{\mathcal{G}}(T),\ \beta \to \infty$ and $I_{\mathcal{G}}(T),\ \beta = 0$ are completely disjoint which allows the detection of a signal. Thus, $\exists \beta_{\text{det}}$ such that $\forall \beta \geq \beta_{\text{det}}$, the detection is possible.

Proof theorem 3 Let $n \in \mathbb{N}$, $\mathcal{G}$ a graph of degree d and $\boldsymbol{M}$ the matrix obtained by cutting the edge e. When $\beta \to \infty$ for a fixed n, the dominant matrix by operator norm will be the one where all the vertices are replaced by signal spike, and which is proportional to $\boldsymbol{v}\boldsymbol{v}^T$. Then applying Wedin perturbation theorem proves that $\exists \beta_{\text{rec}}$ such that $\forall \beta \geq \beta_{\text{rec}}$, the recovery is possible.

Sketch of proof for theorem 4 To show that $\left\|\boldsymbol{M}^{(N)} - \mathbb{E}(\boldsymbol{M}^{(N)})\right\|_{\text{op}} \geq n^{dk/4}$, it is enough to see that the variance of any graph of degree d is superior $dk/4$. Indeed the variance of $\mathrm{Tr}\left((MM^T)^r\right)$ will be always superior to $2rd$. Take two copies of a graph $\mathcal{G}$ and connect corresponding points from each copy to the other. The number of cycles will be equal to $dk/2$. And the variance will be at least equal to $n^{dk/4}$. See example below

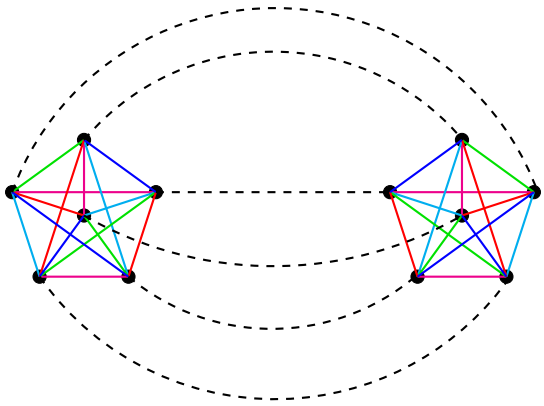


Figure 9: Generalized melon for the complete graph for $k = 5$

Sketch of proof of lemma 5 : We provide a sketch of proof: Let's consider the graph $\mathrm{Tr}\left((A^T A)^r\right)$. Let $\mathcal{B}$ a graph that has $\beta_{\text{dec}} = n^{k/4}$. Let's assume that an intermediate graph dominate

for β slightly larger than β_{dec} and let $\mathcal{G}$ the covering graph giving the contribution for the domination. Then considering the covering graph built from $\mathcal{G}$ by connecting each vertex of spike to its mirror. This has a strictly superior expectation if the graph is connected. The demonstration for the matrices is similar.

Proof theorem 6 : Let's define a set of even numbers $(s_i)_{i\in\{0,\dots,r\}}$ where $s_i/2$ is equal to the number of propagators having its two vertices in the subgraph G_i (we call these propagators internal). Thus we have $s/2 \equiv= \sum_i s_i/2$ internal propagators in total. We can divide the edges of the subgraph $\mathcal{G}$ in three classes. Each edge could be part of a face of either one edge, two edges, three edges, etc.. Let's count how many we can have of each type in an optimal configuration (maximizing the number of faces) for each class.

- Edges only connected to internal propagators, there is at most $\sum_i s_i(s_i-1)/2$ such edges.
  - We can have a maximum of $s_i/2$ faces of one edge since that is the number of internal edges.
  - Then, from the $s_i(s_i-2)/2$ remaining ones, if we want to put them in faces of two edges, the two edges needs too to be in the same subgraph. Thus, we can create a maximum of $s_i/2(s_i/2-1)/2 = s_i(s_i-2)/8$ different couples of propagators. Each couple can't create more than one face since otherwise it will contradict the maximally single trace hypothesis of the graph. So there is a maximum of $s_i(s_i-2)/4$ edges that can be put in two edges faces.
  - The remaining $s_i(s_i-2)/4$ will be in faces of at least three edges.
- Edges connected to one internal and one external propagators, there is at most $\sum_i s_i(k+1-s_i)$ one such edges. They can only be in faces of at least three edges.
- Edges only connected to external propagators, there is at most $\sum_i ((k+1-s_i)(k-s_i)/2)$ such edges. They can only be put in faces of at least two edges.
- The open edge can be put in a face of maximum one edge.

So we have a maximal number of faces given by:

$$\begin{aligned}&\sum_{i=1}^{r}\left(\frac{s_i}{2}+\frac{s_i(s_i-2)}{8}+\frac{s_i(s_i-2)}{12}+\frac{s_i(k+1-s_i)}{3}+\frac{(k+1-s_i)(k-s_i)-2}{4}+1\right)\\ =&\sum_{i=1}^{r}\left(\frac{k(k+1)}{4}-s_i\frac{4k-3s_i-4}{24}+\frac{1}{2}\right).\end{aligned} \tag{15}$$

Let's focus in the case where $s_i = 0$. If all the points of the graph are connected to points of another graph, then the two extremes of the open edge connects to the same graph. In that case the open edge has a contribution of only $1/2$. In the other case, then there is at least k edges which contributes only by at most $1/3$ making them lose $1/6$. Either way the contribution of the subgraph can't go further then $k(k+1)/4$. We obtain:

$$\sum_{i=1}^{r}\left(\frac{k(k+1)}{4}-s_i\frac{4k-3s_i-4}{24}+\frac{1-\delta_{s_i,0}}{2}\right). \tag{16}$$

We can easily check that this quantity is always smaller than $\frac{k(k+1)}{4}$ for $k>5$.

Proof of theorem 7 :

Proof. We will use the fact that studying $\mathrm{Tr}(\boldsymbol{M}-\mathbb{E}(\boldsymbol{M}))^r$ is equivalent to studying the graph of $\mathrm{Tr}(\boldsymbol{M}^r)$ without internal propagators for subgraphs.

Let $\mathcal{B}'$ the subgraph of $\mathcal{B}$ obtained by opening an edge, and $\mathcal{B}^r$ the graph corresponding to $\mathrm{Tr}(M^r)$

$$\mathbb{E}(\boldsymbol{M}_{ij}) = \sum_{a\setminus\{ij\}}\delta^{\mathcal{B}}_{aa}\sum_{\tau^{(0)}\in\mathfrak{S}(k)}\prod_{j=1}^{k}\delta_{a_j{}^{\mathcal{D}}a_{j\,\tau^0(j)}^{\mathcal{D}}}, \tag{17}$$

$$\mathbb{E}(\mathrm{Tr}_{\mathcal{B}^r}(\mathbf{T})) = \sum_a \delta^{\mathcal{B}}_{aa} \sum_{\tau^{(0)} \in \mathfrak{S}(rk)} \prod_{j=1}^{rk} \delta_{a_{j^{\mathcal{D}}} a_{j^{\mathcal{D}}_{\tau^0(j)}}}. \tag{18}$$

We can factorize the $\delta_{a_{j^{\mathcal{D}}} a_{j^{\mathcal{D}}_{\tau^0(j)}}}$ when it is restricted to one subgraph. □

By induction. Let's assume it's true for all $r' \leq r$. Then let's consider $\mathrm{Tr}\big((MM^T)^r\big)$ where M is the matrix associated to the melon by opening one of its edge. Let's consider one of its subgraphs. We can divide the edges of the subgraph in two sets:

- $k-1$ edges that are in faces of at least two edges.
- One external edge divided in half edges.

If the $k-1$ edges are in faces of at least two edges, that means that the propagator at the left vertex and the one at the right vertex are connected to vertices of a same other subgraph.

Then we have two possibilities:

- The two subgraphs are next to each other. Then the middle open edge have a contribution of one, and we can remove the two subgraphs, so it works.
- The two subgraphs are not neighbors. Then the open edges have a contribution of maximum $1/2$ too.

Thus the norm operator is equal to $n^{3/2}$.

For the symmetric case, since $\mathcal{T}^{sym}_{ijk} = \sum_{\tau \in \mathfrak{S}_k} \mathcal{T}_{\tau(i)\tau(j)\tau(k)}$. We just need to expand the melonic matrix $\mathcal{T}^{sym}_{i_1jk}\mathcal{T}^{sym}_{i_2jk}$ in function of the asymmetric tensor $\mathbf{T}$. We obtain a sum of nine matrices (corresponding to graphs with mixed colors). We can show that the operational norm of each of these matrices is at most equal to $n^{3/2}$.

Proof of theorem 8 : Using the matrix decomposition, we can see that we have a sum of four matrices of rank one. Using the usual formula, we can see that the first one dominates by $n^{1/4}$ the last one (consisting in the signal vector) so we are able to use power iteration or one of its variants (like in Anandkumar et al. (2017) and Biroli et al. (2020)) to recover it.

Sketch of Proof of theorem 9 : Let's consider the melonic graph with the edge associated to the index of the space of dimension n_i open. If the open edge is not associated to the larger dimension, then the proof is similar to the proof of theorem 7 and we have a threshold of $(n_1n_2n_3)^{1/4}$. If the open edge is associated to the largest dimension. Let's consider that the other edges of the melonic are in a face of length p. By induction, let's assume there is s propagators associated to the open edges. Then we have a total contribution of $(n_1^{(s-1)/s}n_2^{1/s}n_3^{1/s})^{1/2}$ which is always smaller than $(n_1n_2n_3)^{1/4}$ unless $n_1 \geq n_2n_3$ but then $n_1^{1/2}$ is the largest one.

Sketch of Proof of theorem 10 The Davis-Kahan Theorem provides a bound on subspaces spanned by eigenvectors under perturbation Stewart & guang Sun (1990). If the subspace spanned by the signal vectors is small enough comparing to the total space, the correlation with any signal vector is strong enough for power iteration to find the vector. Then we can use deflation to find the remaining signal vectors.