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 $O(1)$ 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.

Tensor PCA and tensor decomposition (the recovery of many spikes addressed in Section 3.5) is motivated by the increasing number of problems in which it is crucial to exploit the tensorial structure Sidiropoulos et al. (2017). Recently it was successfully used to address important problems in unsupervised learning (learning latent variable models, in particular latent Dirichlet allocation Anandkumar et al. (2014), Anandkumar et al. (2015)), supervised learning (training of two-layer neural networks, Janzamin et al. (2015)) and reinforcement learning (Azizzadenesheli et al. (2016)). Moreover, we note that some of our results tends to generalize the applications of the methods to more practical settings like the case of a tensor with axes of different dimensions (adequate for data which are inherently asymmetric like a video).

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 and T_{ijk}, M_{ij}, v_i for their components. $[p]$ denotes the set $\{1, \dots, n\}$. A real $k-$th order tensor $\mathbf{T} \in \bigotimes_{i=1}^{k} \mathbb{R}^{n_i}$ is a member of the tensor product of Euclidean spaces $\mathbb{R}^{n_i}, i \in [k]$. It is symmetric if $T_{i_1 \dots i_k} = T_{\tau(i_1) \dots \tau(i_k)}\ \forall \tau \in \mathfrak{S}_k$ where $\mathfrak{S}_k$ is the symmetric group (more details are provided in Appendix B). For a vector $\boldsymbol{v} \in \mathbb{R}^n$, we use $\boldsymbol{v}^{\otimes p} \equiv \boldsymbol{v} \otimes \boldsymbol{v} \otimes \dots \otimes \boldsymbol{v} \in \bigotimes^p \mathbb{R}^n$ to denote its p-th tensor power. $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$ denotes the scalar product of $\boldsymbol{v}$ and $\boldsymbol{w}$.

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 its highest eigenvalue

in absolute value. In particular, when a matrix N is symmetric, 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 $\boldsymbol{A}^\top \boldsymbol{A}$ and take its square root since $\left\|\boldsymbol{A}^\top \boldsymbol{A}\right\|_{\text{op}} = \|\boldsymbol{A}\|_{\text{op}}^2$. The trace of $\boldsymbol{A}$ is denoted $\mathrm{Tr}(\boldsymbol{A})$. We denote the expectation of a variable X by $\mathbb{E}(X)$ and its variance by $\sigma(X)$. We say that a function f is negligible compared to a positive function g and we write $f = o(g)$ if $\lim_{n\to\infty} f/g \to 0$.

We say that algorithm detects or recovers a signal from a tensor, when it is able to do so with a probability $p(n) \to 1$ in the large n limit.

Einstein summation convention It is important to keep in mind throughout reading the paper that we will follow 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: $\mathsf{T}_{ijk}\mathsf{T}_{ijk} \equiv \sum_{ijk} \mathsf{T}_{ijk}\mathsf{T}_{ijk}$. It is a common convention when addressing tensor problems that helps make the equations more comprehensible

2 General Framework for Signal Detection and Recovery

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.

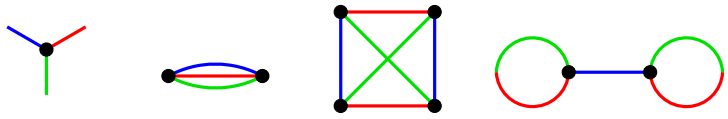
In all of these examples, an important property of the eigenvalues of a n-dimensional matrix $\boldsymbol{M}$ is its invariance under orthogonal transformations $\{\boldsymbol{M} \to \boldsymbol{O}\boldsymbol{M}\boldsymbol{O}^{-1}, \boldsymbol{O} \in \mathrm{O}(n)\}$ where $\mathrm{O}(n)$ is the n-dimensional orthogonal group (i.e. the group of real matrices that satisfies $\boldsymbol{O}\boldsymbol{O}^\top = \boldsymbol{I}_n$). 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 $\mathrm{Tr}(\boldsymbol{A}), \mathrm{Tr}\big(\boldsymbol{A}^2\big), \ldots, \mathrm{Tr}(\boldsymbol{A}^n)$. Obtaining them 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 for a tensor $\mathsf{T} \in \bigotimes_{i=1}^k \mathbb{R}^{n_i}$, tensor quantities that are invariant under orthogonal transformations ($\mathsf{T}_{a_j^1 \ldots a_j^k} \to O^{(1)}_{a_j^1 b_j^1} \ldots O^{(k)}_{a_j^k b_j^k} \mathsf{T}_{b_j^1 \ldots b_j^k}$ for $\boldsymbol{O}^{(i)} \in \mathrm{O}(n_i) \quad \forall i \in [k]$.) should capture similar intrinsic information like the intensity of the signal, and conceivably, there should be other objects related to these quantities that are able to indicate the 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 computing them is very complicated. In contrast, we have a very convenient generalization of the traces of the power matrices for the tensors that we call trace invariants. They have been extensively studied during the last years in the context of high energy physics and many important properties have been proven.

We will first give a more formal definition of trace invariants. Let T be a tensor whose entries are $\mathsf{T}_{i_1,\ldots,i_k}$. Let's define a contraction of a pair of indices as setting them equal to each other and summing over them, as in calculating the trace of a matrix ($\boldsymbol{A}_{ij} \to \sum_{i=1}^n \boldsymbol{A}_{ii}$). The trace invariants of the tensor T correspond to the different way to contract pairs of indices in a product of an even

number of copies of $\mathbf{T}$. The degree of the trace invariants consists in the number of copies of $\mathbf{T}$ contracted. For example, $\sum_{i_1,i_2,i_3} T_{i_1i_2i_3}T_{i_1i_2i_3}$ and $\sum_{i_1,i_2,i_3} T_{i_1i_2i_2}T_{i_1i_3i_3}$ are trace invariants of degree 2. In the rest of this paper, we will use the Einstein summation convention by dropping the sum symbol and imply summation when an index variable appears twice.

A trace invariant of degree d of a tensor $\mathbf{T}$ of order k admits 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. For each graph $\mathcal{G}$, we denote the invariant associated to it $I_{\mathcal{G}}$. Three important examples of trace invariants are invariants associated to the melon diagram (Fig. 1b), the tetrahedral (1c) and the tadpole (1d). Avohou et al. (2019) provides a thorough study about the number of trace invariants for a given degree d. A very useful asset of these invariants is that we are able to compute their expectation for tensors whose components are Gaussian using simple combinatory (Gurau (2017)).



(a) T_{ijk} (b) $T_{ijk}T_{ijk}$ (c) $T_{ijk}T_{ij'k'}T_{i'jk'}T_{i'j'k}$ (d) $T_{ijj}T_{ikk}$

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 if our goal is 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 in two half edges, which amounts to not summing over the two indices i_1 and i_2 associated to these two half-edges and using them to index a matrix instead (see Figure 3 for an example). We denote this matrix $M_{\mathcal{G},e}$. We will drop the index of the matrix when the context is clear.

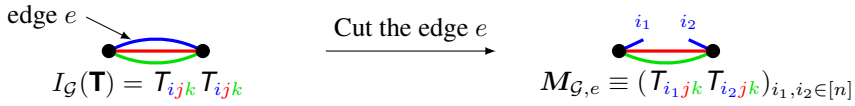


Figure 2: Obtaining a matrix by cutting the edge of a trace invariant graph $\mathcal{G}$

2.3 Phase Transition within this Framework

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

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

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 contribution and a second part $I_{\mathcal{G}}^{(S)}(\mathbf{T})$ enclosing all the other contributions, which are resulting from the addition of a signal tensor.

$$I_{\mathcal{G}} = I_{\mathcal{G}}^{(N)} + I_{\mathcal{G}}^{(S)}. \tag{2}$$

An identical decomposition can be carried out for the matrix. Let's consider a graph $\mathcal{G}$ and its associated 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 ends. We can distinguish three kind of contributions to the matrix $\boldsymbol{M}_{\mathcal{G},e}$ illustrated in Figure 3 (where we denoted the invariant $I'_{\mathcal{G}}$ by I' for simplicity). In what follows, we use the superscript (N) when we want to stress that we are talking about pure random tensors, in order to compute expectations and variances.

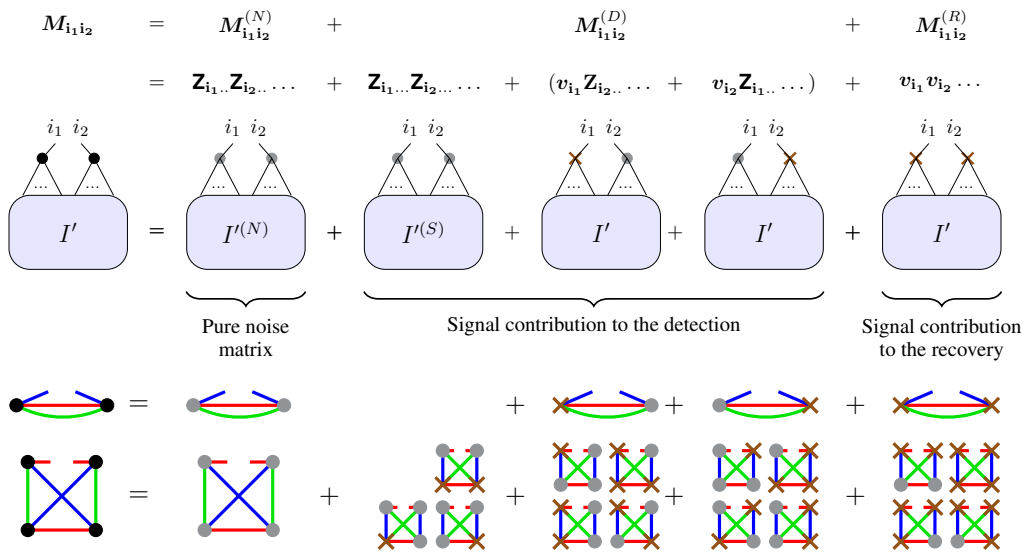


Figure 3: Decomposition of a matrix graph and the melon example

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

Using the lemma 1, for a general graph, we identify three possible phases in function of which matrix operational norm is much larger than the others: If $\left\|\boldsymbol{M}^{(N)} - \mathbb{E}(\boldsymbol{M}^{(N)})\right\|_{\text{op}} \gg \left\|\boldsymbol{M}^{(D)}\right\|_{\text{op}}, \left\|\boldsymbol{M}^{(R)}\right\|_{\text{op}}$ then no recovery, no detection, we can't distinguish if there is a signal. It is for example the phase for $\beta \to 0$. If $\left\|\boldsymbol{M}^{(D)}\right\|_{\text{op}} \gg \left\|\boldsymbol{M}^{(N)} - \mathbb{E}(\boldsymbol{M}^{(N)})\right\|_{\text{op}}, \left\|\boldsymbol{M}^{(R)}\right\|_{\text{op}}$ then detection but no recovery. 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: $\left\|\boldsymbol{M}^{(R)}\right\|_{\text{op}} \gg \left\|\boldsymbol{M}^{(N)} - \mathbb{E}(\boldsymbol{M}^{(N)})\right\|_{\text{op}}, \left\|\boldsymbol{M}^{(D)}\right\|_{\text{op}}$. We recover the signal vector. It is for example the phase for $\beta \to \infty$.

2.4 Algorithmic Threshold for a General Graph

We can now state the important algorithms that will be essential for this paper.

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

Input: The tensor $\mathbf{T} = \beta \boldsymbol{v}^{\otimes k} + \mathbf{Z}$
Goal: Detection of $\boldsymbol{v}$
1 Calculate the scalar $\alpha = I_{\mathcal{G}}(\mathbf{T})$;
2 Compare $\left|\alpha - \mathbb{E}(I_{\mathcal{G}}^{(N)})\right|$ with the variance $\sigma(I_{\mathcal{G}}^{(N)})$.
Result: Gives a probability of the presence of a spike

Let's clarify that the claims concern the large n limit and that empirically the approximation of large n limit seems valid for $n > 25$. The first algorithm gives a criteria for distinguishing a pure noise tensor from a tensor with a spike. Denoting the expectation and the variance of the trace invariant associated to a graph $\mathcal{G}$ for a gaussian random tensor model by $\mathbb{E}(I_{\mathcal{G}}^{(N)})$ and $\sigma(I_{\mathcal{G}}^{(N)})$. The algorithm consists simply in calculating the trace invariant of the tensor and comparing its distance

from $\mathbb{E}(I_{\mathcal{G}}^{(N)})$ with $\sigma(I_{\mathcal{G}}^{(N)})$. It is straightforward to see that calculating a trace invariant (which is a scalar) like $\mathsf{T}_{ijk}\mathsf{T}_{ijk}$ only needs $O(1)$ memory.

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 $\beta \geq \beta_{det}$.*

The second algorithm is able to recover the spike through the construction of the matrix of size $n \times n$ $\boldsymbol{M}_{\mathcal{G},e}$ associated to a given graph $\mathcal{G}$ and edge e.

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

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

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 $< \boldsymbol{v}, \boldsymbol{v}_0 >= 1 - o(1)$ for $\beta \geq \beta_{rec}$.*

The last theorem gives a lower bound to the threshold above which we can detect and recover a spike using a single graph. Interestingly, this threshold which appears naturally in our framework, matches the threshold below which there is no known algorithm that is able to recover the spike in polynomial time.

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

3 Some applications of this framework

Using these algorithms, we are now able to investigate the performance of our framework in various theoretical settings. In the first two subsections, we study the algorithms associated to two trace invariants of degree 2. They consist of the melonic diagram, which gives a very practical detection algorithm, and whose recovery algorithm is a variant of the unfolding algorithm, and the tadpole diagram which gives a similar algorithm to the homotopy algorithm. The third subsection goes further in terms of graph's degree and investigate the algorithms associated to the perfect one-factorization graphs (consisting in the tetrahedral when $k = 3$). The last two sections are an illustration of the versatility of this framework. We study the case $n_1 \neq n_2 \neq n_3$ which is important for practical applications where the dimensions are naturally asymmetric. Our methods allows us to derive a new algorithmic threshold. It also brings closer the Tensor PCA and Tensor decomposition problems by handling the recovery of many orthogonal spikes at the same threshold of a single spike: $n^{3/4}$.

3.1 The melon graph similar to Tensor unfolding

Let's consider the invariant $\mathsf{T}_{i_1 \ldots i_n}\mathsf{T}_{i_1 \ldots i_n}$ (illustrated by the graph in Fig. 1b when $k = 3$). Its recovery algorithm (with the matrix obtained by cutting any of the edges) is similar to the tensor unfolding method presented in Richard & Montanari (2014). The difference is that the melonic algorithm uses only a matrix $n \times n$ instead of a matrix $n^{k/2} \times n^{k/2}$ for the tensor unfolding. However, the main contribution of this framework for this graph is that it allows the detection in a linear time (there is three sums to perform so n^3 operations, and the tensor input is of size n^3) in a constant memory space (it just calculates a scalar). Giving it a potential interest as a first step for detecting the signal before deciding to use more computationally costly methods. Also, to the best of our knowledge, this framework is the first to theoretically prove a conjuncture that the unfolding algorithm works also for the symmetric case.

Theorem 5. *The algorithms 1 and 2 works for the melon with $\beta_{det} == O(n^{k/4}),\ \beta_{th} = O(n^{k/4})$ in linear time and respectively $O(n^2)$ and $O(1)$ memory requirement.*

3.2 THE TADPOLE GRAPH

Figure 1d has a special characteristic: we can obtain two disconnected parts by cutting only one line. The consequence of this characteristic is that the matrix obtained by cutting that edge is of rank one (in the form of $\boldsymbol{v}\boldsymbol{v}^T$). Thus, the vector $\boldsymbol{v}$ has a weak correlation with the signal $\boldsymbol{v}_0$, which allow variants of tensor power iteration ($v_i \leftarrow \mathsf{T}_{ijk} v_j v_k$) to recover it (simple power iteration is also able to recover the signal vector, but a formal proof needed to consider some more sophisticated variants of it like in Anandkumar et al. (2017) and Biroli et al. (2020)). It is a variant of the already existent homotopy algorithm.

Theorem 6. *The tadpole graph allows to recover the signal vector for $k \geq 3$ and $\beta = O(n^{k/4})$ by using local algorithms to enhance the signal contribution of the vector T_{ijj} (corresponding to the half graph resulting from cutting the middle edge).*

3.3 TENSOR INVARIANT BASED ON THE PERFECT ONE-FACTORIZATION GRAPHS

The next graph we investigate is of degree strictly superior to two. We investigate this example because it has nice combinatorial properties which simplify the proofs to its detection and recovery theorems. Also, the demonstrations of these theorems involve standard methods that can be used to more general graphs. It is a generalization of the tetrahedral graph and it shows very interesting empirical results.

Theorem 7. *The complete graph is able to recover the signal vector for $\beta = O(n^{k/4})$.*

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 $n_1 \geq n_2 \geq n_3$.

$$\mathbf{T} = \beta \boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{v}_3 + \mathbf{Z} \quad \text{where} \quad \boldsymbol{v}_i \in \mathbb{R}^{n_i}, \quad n_i \in \mathbb{N}. \tag{3}$$

One way to obtain the spike in this case is to insert the tensor in a new symmetric tensor $\mathbf{T}^{\text{sym}} \in \mathbb{R}^{n_1+n_2+n_3} \otimes \mathbb{R}^{n_1+n_2+n_3} \otimes \mathbb{R}^{n_1+n_2+n_3}$ (defined as $\mathsf{T}^{\text{sym}}_{i\ n_1+j\ n_1+n_2+k} = \mathsf{T}_{ijk}$ for $i \in [n_1],\ j \in [n_2],\ k \in [n_3]$ and the other components of $\mathbf{T}^{\text{sym}}$ are obtained by symmetry) and then apply our usual methods to recover a vector from which we can extract the three signal vectors, the threshold to recover the three vectors would be equal to $(n_1 + n_2 + n_3)^{3/4}$).

However our framework naturally handles this case without resorting to symmetrization. We have the following theorem:

Theorem 8. *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 $(n_1 n_2 n_3)^{1/4}$.*

To the best of our knowledge, the threshold $\max\left((n_1 n_2 n_3)^{1/4}, n_1^{1/2}\right)$ for $\boldsymbol{v}_1$ and $(n_1 n_2 n_3)^{1/4}$ for $\boldsymbol{v}_2,\ \boldsymbol{v}_3$ is the new best known threshold for this case. Not only it coincides with the previous threshold when $n_1 = n_2 = n_3$, but also the result for $n_1 > n_2 n_3$ seems logical. Indeed, in the limit case of $n_2 = n_3 = 1$, the tensor will be a vector. And since the expectation of the norm of a random gaussian vector is $n^{1/2}$ we would need $\beta > n^{1/2}$ to be able to detect the signal vector.

3.5 HANDLING MULTIPLE ORTHOGONAL SPIKES

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

$$\mathbf{T} = \sum_{i=1}^{p} \beta_i \boldsymbol{v}_i^{\otimes k} + \mathbf{Z} \quad \text{where} \quad \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \quad \forall i \neq j. \tag{4}$$

Theorem 9. *If we have a number of spikes p that does depend on n, we can recover the p spikes by an alternation of the melonic diagram, power iteration and deflection.*

We first use the melonic diagram to obtain a vector correlated with the signal vectors, then we follow by power iteration to obtain a normalized vector $\boldsymbol{v}$ highly correlated to one of the spikes and then deflation which consists in replacing the tensor $\mathbf{T}$ by $\mathbf{T} - \alpha \boldsymbol{v}^{\otimes 3}$ where $\alpha = \mathbf{T}.\boldsymbol{v}^{\otimes 3}$. These last two steps are needed in the worst case scenario where there is no gap of intensity between the of the spikes. The experimental results suggest that it works not only for small p compared to n but also for values of p up to n.

4 Numerical experiments

In this section we will investigate the empirical results of the previously mentioned applications to compare with our theoretical results. We restricted to the dimension $k = 3$ for simplicity. We give more details about the experiments settings in the appendix.

4.1 Detection and comparison of recovery methods

For the detection algorithm, we use the tetrahedral graph. We vary the dimension n, then. The detection uses the fact that the distributions of $\mathbf{Z}_{ijk}\mathbf{Z}_{ijk}$ and $(\mathbf{Z}_{ijk} + \beta \boldsymbol{v}^{\otimes 3})(\mathbf{Z}_{ijk} + \beta \boldsymbol{v}^{\otimes 3})$, where $\mathbf{Z}$ is a random gaussian tensor and $\boldsymbol{v}$ an unitary vector, become distinct when β is large enough. To illustrate this fact, in Fig. 4 left, we have drawn the frequency of the distributions of the melonic invariant over 500 instances of $\mathbf{Z}$, in blue without signal and in orange with signal. Thus, we use Intersection over Union as an evaluation metric to measure the accuracy of the signal detector.

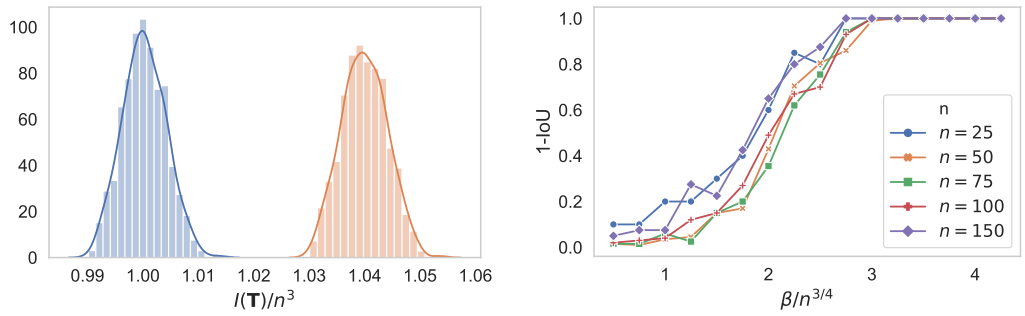


Figure 4: Detection using the melonic graph

For the recovery algorithm in the symmetric case, we will compare our method (tetrahedral) to two type of results: (i) Other algorithmic methods: the melonic (tensor unfolding) and the homotopy. To the best of our knowledge, they give the state of art respectively for the symmetric and asymmetric tensor (Biroli et al. (2020)). Other methods exist but are either too computationally expensive (sum of squares) or are variants of these algorithms. (ii) Information-theoretical results: In (Richard & Montanari (2014)), it was proven that computing the global minimum $\boldsymbol{v}$ of the function $\boldsymbol{v} \mapsto T_{ijk} v_i v_j v_k$ recovers the signal vector $\boldsymbol{v}_0$ above a theoretical threshold $\beta_{\text{th}} = 2.87\sqrt{n}$ but with exponential time, and that no other approach can do significantly better than that. Thus, to assess the performance of the tetrahedral, we plot in red dashed line denoted "perf" the deep minimum that is closest to $\boldsymbol{v}_0$, by using gradient method with an initialization in $\boldsymbol{v}_0$. Interestingly, we note that not only the tetrahedral is able perform better than other methods, but is able as well to recover the spike at the theoretical threshold even at $n \sim 100$.

4.2 A spike with different dimensions

We have in this case a spike $\boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \boldsymbol{v}_3$ and we wish to recover the three vectors. 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 it stays $(n_1 n_2 n_3)^{1/4}$ for v_1 and v_2), corresponding to what our theoretical study predicted.

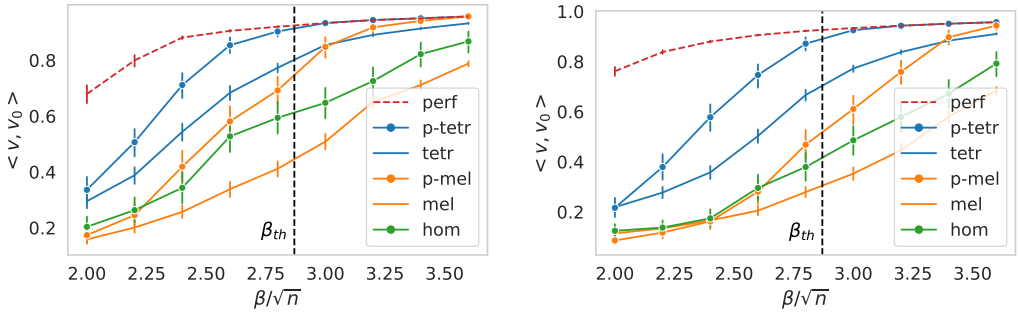


Figure 5: Comparison of different methods (the prefix p- for power iteration). Left: symmetric n=100, right: symmetric n=150.

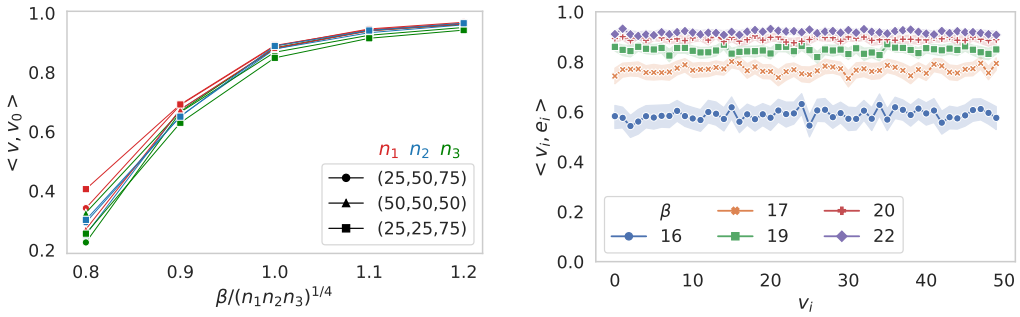


Figure 6: Left: Recovery of 50 orthogonal spikes for different β,right: Detection of a spike

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. Important directions of future research is to explore the potential of more general graphs and to try to apply these new methods to real data.

References

Anima Anandkumar, Dean P Foster, Daniel Hsu, Sham M Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. *Algorithmica*, 72(1):193–214, 2015.

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

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:

2773–2832, 2014.

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.

Kamyar Azizzadenesheli, Alessandro Lazaric, and Animashree Anandkumar. Reinforcement learning of pomdps using spectral methods. *arXiv preprint arXiv:1602.07764*, 2016.

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.

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 and Theoretical*, 53(17):174003, 2020.

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.

Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 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.

Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

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 $\boldsymbol{M} = \beta \boldsymbol{v}_0 \boldsymbol{v}_0^\top + \boldsymbol{N}$ a square matrix with $\|\boldsymbol{v}_0\| = 1$. Let $\boldsymbol{v}$ denote the eigenvector with the largest eigenvalue of $\boldsymbol{M}$. If $\beta > 2\|\boldsymbol{N}\|_{\text{op}}$ then:

$$1 - \langle \boldsymbol{v}, \boldsymbol{v}_0 \rangle \leq \frac{8\|\boldsymbol{N}\|_{\text{op}}}{\beta^2}. \tag{5}$$

- **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 $C > 0$,

$$\Pr(|X - \mu| \geq C\sigma) \leq \frac{1}{C^2}. \tag{6}$$

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

$$\rho(\boldsymbol{A}) = \lim_{s \to \infty} \|\boldsymbol{A}^s\|^{\frac{1}{s}}. \tag{7}$$

  In particular we can choose the Frobenius norm:

$$\|\boldsymbol{A}\|_{\text{F}} = \sqrt{\text{trace}(\boldsymbol{A}^* \boldsymbol{A})}. \tag{8}$$

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.

Let's consider a graph $\mathcal{G}$ of order d associated to a trace invariant $I_{\mathcal{G}}(\mathbf{T})$ where $\mathbf{T}$ is a tensor of order k. since the trace invariant is a contraction of pairs of indices of d copies of $\mathbf{T}$, we denote the c-th index of the i-th copy of $\mathbf{T}$ by ${a_i}^c$ and the set of the d indices of the i-th copy by ${a_j}^{\mathcal{D}}$. By the definition of the Gaussian measure we have the equation:

$$\mathbb{E}(\text{Tr}_{\mathcal{G}}(\mathbf{T})) = \sum_a \delta^{\mathcal{G}}_{aa} \sum_{\tau \in \mathfrak{S}(k)} \prod_{j=1}^{k} \delta_{{a_j}^{\mathcal{D}} {a_j}^{\mathcal{D}}_{\tau(j)}}, \tag{9}$$

where $\mathfrak{S}(k)$ is the symmetric group (the group of all permutations of a set made of n elements, where we define a permutation of a set by an arrangement of its elements in a linear order. For example the permutations of $\{1, 2, 3\}$ are $(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)$.). $\sum_a$ indicates a sum over all the indices involved in the computation of the trace invariant, illustrated by a half-edge in the graph (we have $k \times d$ indices in total: k indices for each one of the d copies of $\mathbf{T}$ of the trace invariant expression). $\delta^{\mathcal{G}}_{aa}$ indicates that contracted indices (illustrated by being two ends of the same edge) of the tensors have to be equal. The term $\delta_{{a_j}^{\mathcal{D}} {a_j}^{\mathcal{D}}_{\tau(j)}}$ indicates that we set equal the c-th indices of the copy j and $\tau(j)$ of $\mathbf{T}$ for $c \in \{1, \ldots, d\}$

In order to have a clearer representation of the expression in equation 9, in particular the sum over the symmetric group, we introduce the concept of covering graph used in Gurau (2014): a covering graph of $\mathcal{G}$ consists in adding $d/2$ new edges of color 0 (also called propagators) relying pairwise the vertices of $\mathcal{G}$ (see Fig. 7 for an example). We use the covering graphs to illustrate the permutations $\tau \in \mathfrak{S}(k)$. Then, an edge of color 0 between two vertices i and j will identify all the indices of the i-th and j-th copies of $\mathbf{T}$. Thus, it will be equivalent to the term $\delta_{{a_j}^{\mathcal{D}} {a_j}^{\mathcal{D}}_{\tau(j)}}$. Using the covering graph, we will be able to rewrite equation 9.

For example, in the case where the edges could only have one color, we have the formula:

$$\begin{aligned} \mathbb{E}(\text{Tr}_{\mathcal{G}}(\mathbf{T})) = & \sum_{\mathcal{G}', \mathcal{G}' \setminus \mathcal{E}^0(\mathcal{G}') = \mathcal{G}} \sum_{aa} \left(\prod_{c=1}^{3} \prod_{e^c = (v_j, v_k) \in \mathcal{E}^c(\mathcal{G})} \delta_{{a_j}^c, {a_k}^c} \right) \\ & \left(\prod_{e^0 = (v_p, v_q) \in \mathcal{E}^0(\mathcal{G})} \delta_{{a_p}^1, {a_q}^1} \delta_{{a_p}^2, {a_q}^2} \delta_{{a_p}^3, {a_q}^3} \right), \end{aligned} \tag{10}$$

Where we denote $\mathcal{E}^c(\mathcal{G})$ the set of edges of color c of $\mathcal{G}$. Thus $\{\mathcal{G}', \mathcal{G}' \backslash \mathcal{E}^0(\mathcal{G}') = \mathcal{G}\}$ denotes the graphs which restrict to the graph $\mathcal{G}$ when we remove their edges of color 0, which are by the definition the covering graphs.

Let $c_1, c_2 \in \{0, \dots, d\}$ be two different colors of edges. We denote $F^{c_1,c_2}(\mathcal{G})$ the number of closed cycles (that we also call face) of 2 colors of $\mathcal{G}$. More explicitly, it consists of the number of connected subgraphs left when we keep in $\mathcal{G}$ only the edges of colors c_1, c_2 (see Fig 8 for an illustration). We denote $F(\mathcal{G}) \equiv \sum_{c_1 \neq c_2 \in \{1,\dots,d\}}$. Then, with a little work (see Gurau (2014) for a proof), we obtain :

$$\mathbb{E}(\mathrm{Tr}_{\mathcal{G}}(\mathbf{T})) = \sum_{\mathcal{G}', \mathcal{G}' \backslash \mathcal{E}^0(\mathcal{G}') = \mathcal{G}} n^{F(\mathcal{G}') - F(\mathcal{G})} = \sum_{\mathcal{G}', \mathcal{G}' \backslash \mathcal{E}^0(\mathcal{G}') = \mathcal{G}} n^{\sum_c F^{0,c}(\mathcal{G}')}. \tag{11}$$

This will be the expression we will use to calculate the expectations of our graphs if each edges has only one color.

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

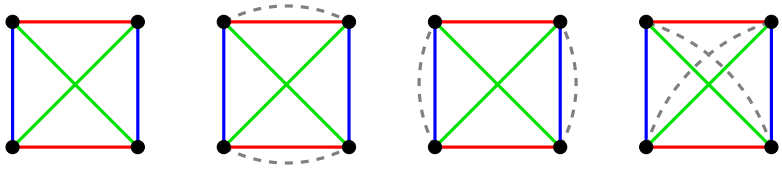


Figure 7: Tetrahedral and its covering graph

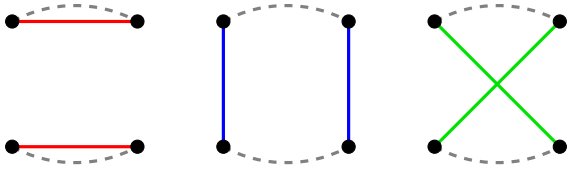


Figure 8: The faces of the first covering graph

For each of the three covering graphs, there is one color having two faces (the red in Fig. 8) and two colors having only one face. Hence the expectation of the tetrahedral graph is given by $3n^4$:

$$\mathbb{E}(\mathrm{Tr}_{\mathcal{G}}(\mathbf{T})) = \sum_{\mathcal{G}', \mathcal{G}' \backslash \mathcal{E}^0(\mathcal{G}') = \mathcal{G}} n^{\sum_c F^{0,c}(\mathcal{G}')} = n^{2+1+1} + n^{1+2+1} + n^{1+1+2} = 3n^4. \tag{12}$$

B.1 THE CASE OF EDGES OF MIXED COLOR

For a general trace invariant, 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. So we adapt the steps in Gurau (2014) to obtain equation 12 to this new situation. More precisely, the cycles that we will count in a covering graph $\mathcal{G}'$ of $\mathcal{G}$ are built by beginning from a half-edge of $\mathcal{G}$ then alternating until they go back to the starting half-edge by:

- go from a half-edge of $\mathcal{G}$ of color $c_1 \neq 0$ to its associated half edge of color $c2$.
- go through the propagator to the half-edge of color c_2 connected to the other end of the propagator.

$$\mathbb{E}(\mathrm{Tr}_{\mathcal{G}}(\mathbf{T})) = \sum_{\mathcal{G}', \mathcal{G}' \backslash \mathcal{E}^0(\mathcal{G}') = \mathcal{G}} n^{\text{mixed cycles}} \tag{13}$$

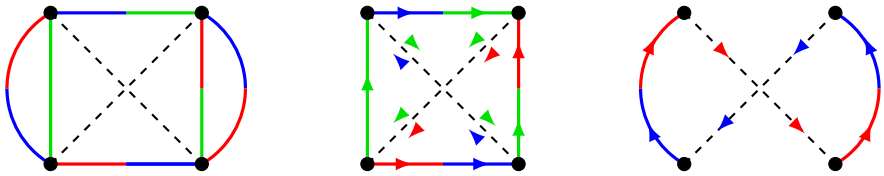


Figure 9: A mixed graph and its two cycles

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 removes the contraction of two indices and leaves them free as variables 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 $(1, 0, \dots, 0)$. 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.

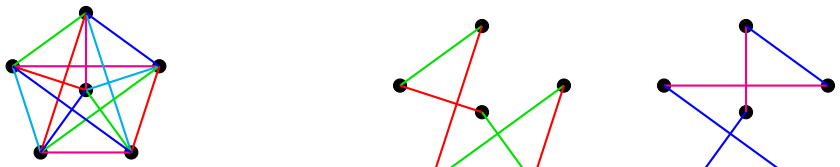
C THE PERFECT ONE-FACTORIZATION GRAPH

In the mathematical field of graph theory, a graph $\mathcal{G}$ is said to be one-factorizable if:

- It is k-regular, which means that each vertex is incident to exactly k edges.
- Its edges can be colored in k colors such that no two incident edges have the same color.

A perfect one-factorization of $\mathcal{G}$ happens if, for every two colors c_1 and c_2 of the k colors, restricting $\mathcal{G}$ to the edges of color c_1 and c_2 leads to a connected subgraph cycle (see figure below).

It is conjectured (Wagner (1992)) that they exist for all n but, until now, they were proven only for prime numbers and up to 56. We use this example because of its combinatorial simplicity. The perfect one-factorization graphs have already been studied in the context of random tensors in Ferrari et al. (2019).



(a) Complete graph for $k = 5$

Figure 10: Complete graph for $k = 5$ and some two colors restriction

D DETAILS ON THE EXPERIMENTS

D.1 DETECTION EXPERIMENTS

D.1.1 THE SETTINGS

D.2 THE SETTINGS

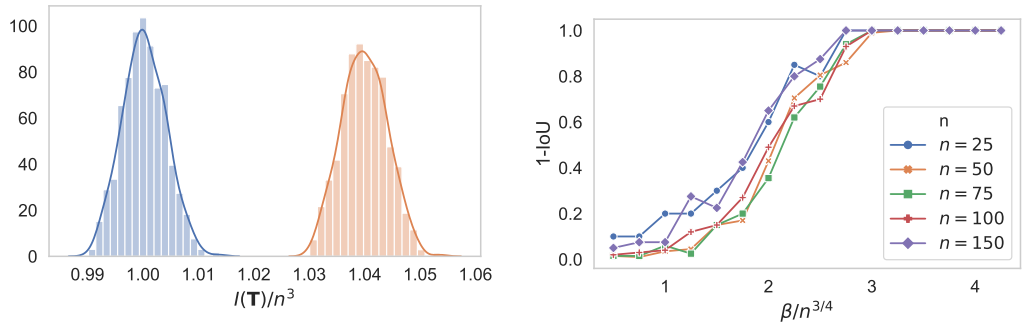
For each of the recovery figures, we repeated 50 times the experiment to draw the error bar (corresponding to a 95% interval confidence). Then we plot the correlation between the vector returned by the algorithm $\boldsymbol{v}$ and the signal vector $\boldsymbol{v}_0$

Each experiment is an independent instance consisting in the following steps:

- We generate randomly the n components of the signal vector $\boldsymbol{v}_0$ and then normalize it.
- We generate randomly the n^3 components of the random tensor $\mathbf{Z}$. If we are in the symmetric case, we symmetrize it with the same normalization than Richard & Montanari (2014)):
$$Z_{ijk} = (Z_{ijk} + Z_{ikj} + Z_{jik} + Z_{jki} + Z_{kij} + Z_{kji})/\sqrt{12}$$
- We compute the tensor $\mathbf{T} = \mathbf{Z} + \beta \boldsymbol{v}^{\otimes k}$
- We compute our matrix associated to the invariant $\boldsymbol{M}$ constructed from $\mathbf{T}$ using the numpy method tensordot in Python.
- We compute its leading eigenvector $\boldsymbol{v}$
- We plot the correlation between the vector resulted from the algorithm $\boldsymbol{v}$ and the signal vector we aim to recover $\boldsymbol{v}_0$ (namely scalar product $< \boldsymbol{v}, \boldsymbol{v}_0 >$).

D.3 THE DETECTION

We see that the detection agrees with the threshold found theoretically. The figure in the left shows the distribution of the melonic invariant without and with noise to illustrate how the detection works .



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

(b) Detection with the melon graph.

Figure 11: Detection using the melon graph

D.4 RECOVERY

D.4.1 THE SETTINGS

D.5 THE SETTINGS

For each of the recovery figures, we repeated 50 times the experiment to draw the error bar (corresponding to a 95% interval confidence). Then we plot the correlation between the vector returned by the algorithm $\boldsymbol{v}$ and the signal vector $\boldsymbol{v}_0$

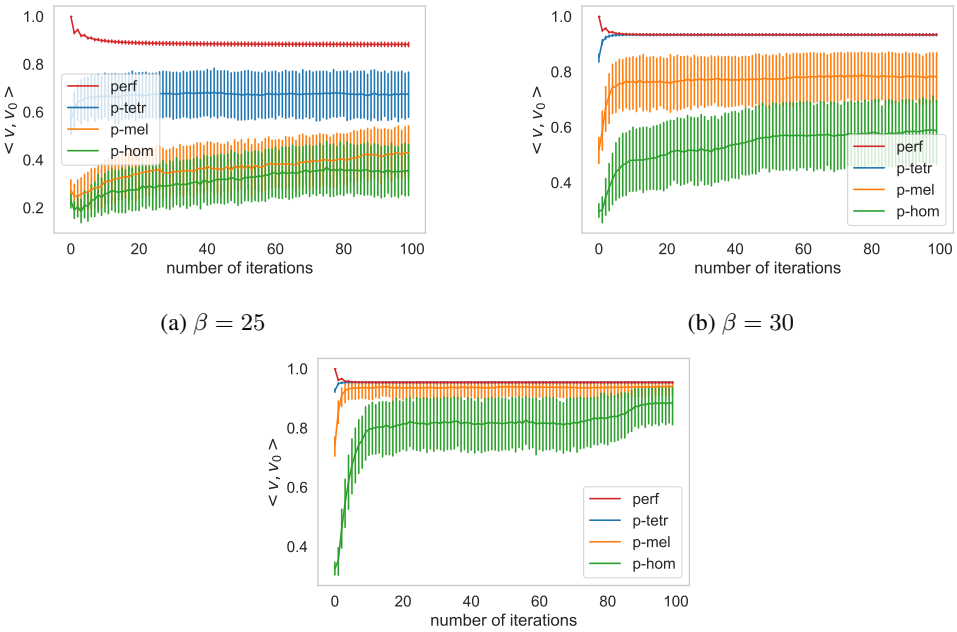
Each experiment is an independent instance consisting in the following steps:

- We generate randomly the n components of the signal vector $\boldsymbol{v}_0$ and then normalize it.
- We generate randomly the n^3 components of the random tensor $\mathbf{Z}$. If we are in the symmetric case, we symmetrize it with the same normalization than Richard & Montanari (2014)):

$$Z_{ijk} = (Z_{ijk} + Z_{ikj} + Z_{jik} + Z_{jki} + Z_{kij} + Z_{kji})/\sqrt{12}$$

- We compute the tensor $\mathbf{T} = \mathbf{Z} + \beta \boldsymbol{v}^{\otimes k}$
- We compute our matrix associated to the invariant $\boldsymbol{M}$ constructed from $\mathbf{T}$ using the numpy method tensordot in Python.
- We compute its leading eigenvector $\boldsymbol{v}$
- We plot the correlation between the vector resulted from the algorithm $\boldsymbol{v}$ and the signal vector we aim to recover $\boldsymbol{v}_0$ (namely scalar product $< \boldsymbol{v}, \boldsymbol{v}_0 >$).

D.5.1 THE NUMBER OF ITERATIONS OF POWER ITERATION FOR EACH METHOD



(a) $\beta = 25$

(b) $\beta = 30$

(c) $\beta = 35$

Figure 12: Correlation with the signal vector in function of the number of iterations of the power iteration for $n = 100$

Empirical results shows the fast convergence using the power iteration method following the tetrahedral method.

D.6 MEMORY AND TIME REQUIREMENTS OF THE METHODS

General requirements:

Since our method consists on computing algebraic operations on tensors entries to store in matrix $n \times n$, the naive recovery method for a tensor of rank k and for a tensor invariant of degree d would require $O(n^2)$ space requirements and $O(n^{kd})$ time complexity.

However, similarly to the matrix case where naive implementation requires n^3 but better asymptotic bounds on the time required to multiply matrices have been known since the work of Strassen in the 1960s. It is possible to simplify the computations by using decomposition on blocks. Many different algorithms have been designed for multiplying matrices on different types of hardware, including parallel and distributed systems, where the computational work is spread over multiple processors (perhaps over a network). Optimizing tensor contractions is out of the scope of this paper.

Tetrahedral method with $\tilde{O}(n^4)$ time and $\tilde{O}(n^3)$ space requirement:

We can calculate the leading eigenvector of the matrix associated to the tetrahedral: $M^t_{i_1 i_2} \equiv \mathsf{T}_{i_1 jk} \mathsf{T}_{i_2 j' k'} \mathsf{T}_{ij'k} \mathsf{T}_{ijk'}$. Using the decomposition, we can prove that there is an eigengap between the highest eigenvalue $\sim \beta^4$ and the second one. Thus, we can use matrix power method with $\log(n)$ steps in order to recover the leading eigenvector $\boldsymbol{v} \leftarrow \boldsymbol{M}^t \boldsymbol{v}$.

We can compute it in a time of $\tilde{O}(n^4)$ and space $\tilde{O}(n^3)$. Indeed this method requires two intermediate matrices we denote $\boldsymbol{A}^1, \boldsymbol{A}^3$ and a intermediate $3-$dimensional tensor we denote $\mathbf{A}^2$, in every step there is at most n^4 operations. we follow these steps:

- We take an initial vector $\boldsymbol{v}$
- We compute the matrix $\boldsymbol{A}^1$ defined as $A^1_{j'k'} \equiv \mathsf{T}_{i_2 j' k'} v_{i_2}$
- We compute the tensor $\mathbf{A}^2$ defined as $\mathsf{A}^2_{k'ik} \equiv A^1_{j'k'} \mathsf{T}_{ij'k}$
- We compute the matrix $\boldsymbol{A}^3$ defined as as $A^3_{kj} \equiv \mathsf{A}^2_{k'ik} \mathsf{T}_{ijk'}$
- Return the vector $\boldsymbol{v}'$ defined as $v'_{i_1} \equiv A^3_{kj} \mathsf{T}_{i_1 jk}$

Ours methods have in general n^2 memory requirements (where to store the matrix).

Comparison between different methods

We use our results and the results in Anandkumar et al. (2017) to fill the following table

Table 1: algorithmic threshold, time and space requirements for each method

Method	Time	Space	Threshold
Melonic detection	$O(n^3)$	$O(1)$	$\Omega(n^{3/4})$
Melonic recovery /Unfolding	$O(n^3)$	$O(n^2)$	$\Omega(n^{3/4})$
Tetrahedral recovery	$O(n^3)$	$O(n^3)$	$\Omega(n^{3/4})$
Sum of squares	$> \Omega(n^6)$	$> \Omega(n^6)$	$\Omega(n^{3/4})$
Homotopy	$O(n^3)$	$O(n)$	$\Omega(n^{3/4})$
Information-theoretic	Exp	$O(n)$	$\Omega(n^{1/2})$

We didn't fill the time requirement of the tetrahedral method because we don't know how to compute it efficiently. Indeed, in the same way that there exist algorithms optimized for matrix multiplications (dividing the matrices into blocks), we use tensordot from Python for our contractions. It uses optimized matrix multiplications to fasten drastically the calculations. Naive implementation of the tetrahedral matrix calculation would require $O(n^6)$, but wee see in the empirical plot that it is sharply below that complexity. We hope that there is still a lot of optimization to be done in tensor contractions in the same way it was done for matrix calculations, especially with the rise of the

use of tensorial objects. The ability to be parallelized (each component of the matrix $\boldsymbol{M}_{\mathcal{G},e}$ can be computed independently) is also an important feature of our methods for practical applications.

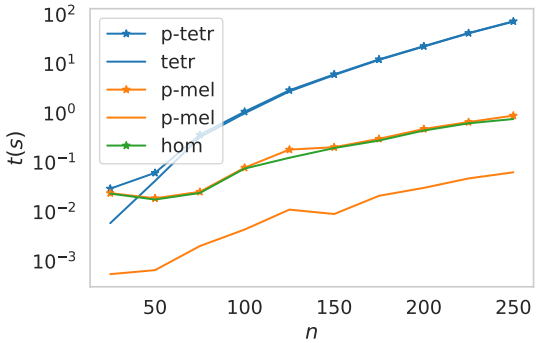


Figure 13: Empirical time requirement for different methods in function of n without taking into account the input reading

However, to have a more accurate estimation of the time requirement, one has to take into account time to read the input (a tensor of size n^3).

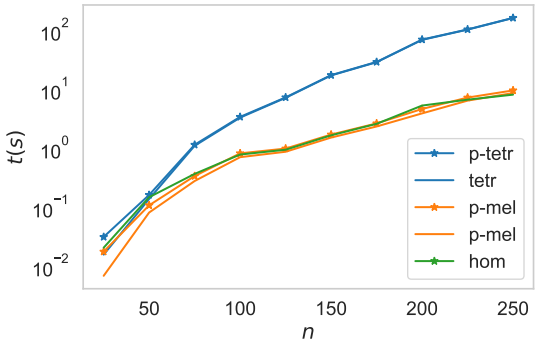


Figure 14: Empirical time requirement for different methods in function of n with taking into account the input reading

E PROOFS

Useful lemmas and definitions:

Lemma 10. *The covering graphs involved in the calculation of $E(\mathrm{Tr}(\boldsymbol{M} - \mathbb{E}(\boldsymbol{M}))^r)$ are the same that are involved in the calculation of $E(\mathrm{Tr}(\boldsymbol{M}^r))$ minus the covering graphs where one of the elementary subgraphs has only internal propagators.*

Lemma 11. *If $\beta_{det} = \mathcal{O}(n^{3/4})$, the intermediate matrices and the intermediate graphs are never dominant.*

Calculating the operational norm of a matrix The trace of the power r of a matrix $\mathrm{Trace}(M^r)$ 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.

Elementary subgraph The subgraph corresponding to $\boldsymbol{M}$ like in Figure. They are the building bricks.

Internal propagator: When the propagators connect vertices in the same elementary subgraph.

Proof Lemma 1 Let $\boldsymbol{M}$ be the matrix obtained by cuting the edge e of a trace invariant graph $\mathcal{G}$. We know that the equation 9 identify the indices i and j that are ends of a same edge by including a factor δ_{ij}. Hence, 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, $\mathbb{E}(M_{ii}) = \mathbb{E}(\mathrm{Tr}(\boldsymbol{M}))/n = \mathbb{E}(\mathcal{G})/n$.

(a) Example of a covering graph

(b) The cycle containing the open edge contributing by δ_{ij}

Figure 15: Cycle with an open edge

Proof theorem 2 Detecting a signal could be done through calculating the value at $\mathbf{T}$ of the invariant associated to $\mathcal{G}$ and comparing it to the pure noise distribution of this invariant.

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.

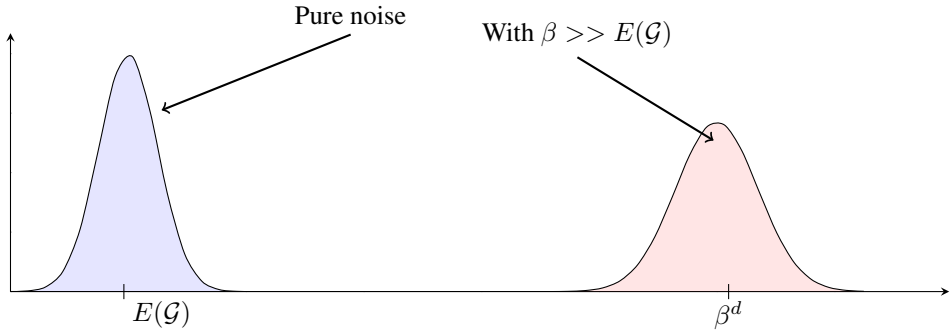
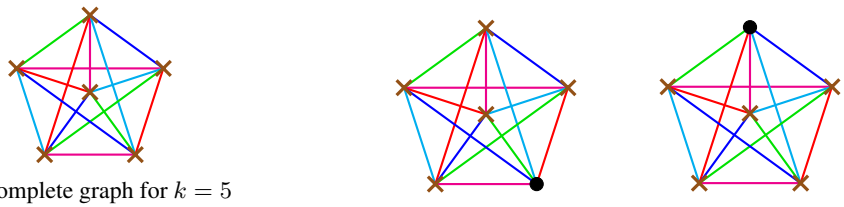


Figure 16: The pure noise and strong signal distributions of an invariant of degree d

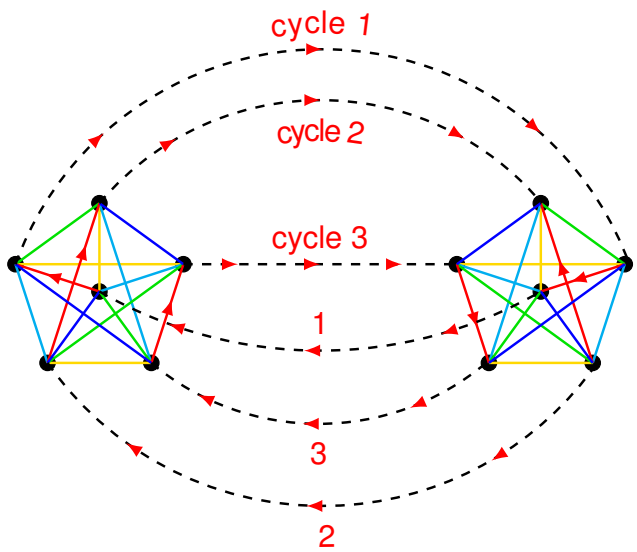


(a) Complete graph for $k = 5$

Figure 17: Complete graph for $k = 5$ and examples of graphs contributing to the detection

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.

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 than $dk/4$. 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). Since the expectation is equal to the sum on covering graphs, the variance will always be larger than $n^{dk/4}$. As for the variance of $\mathrm{Tr}\big((MM^T)^r\big)$ it will be always superior to $2rd$.



Proof of theorem 5 : We will use the lemma 10.

Let's prove by induction that $\mathbb{E}(\mathrm{Tr}\big((MM^T)^r\big)) \leq n^{kr+1}$ where M is the matrix associated to the melon by opening one of its edge.

For $r = 1$ it is straightforward by simply enumerating the cycles. Now, let $r > 1$ and let's assume the statement is true for all $r' < r$. Let's consider one of its elementary subgraphs (defined in the beginning of the section). Using lemma 10, 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 exactly two edges, itt means that the propagators at both the left and right vertices are connected to the vertices of the same other elementary subgraph.

This will lead to two possibilities:

- The two elementary subgraphs are next to each other. Then we remove these two elementary subgraphs and connect the two left open edges. We will have a contribution of n^k from the two elementary subgraphs and use the induction hypothesis to show that the remaining graph has a contribution of maximum $n^{k(r-1)+1}$. Combining them we have a total contribution of $n^{k(r-1)+1+k} = n^{kr+1}$ which proves the theorem.
- The two subgraphs are not next to each other: Then the open edges have also a contribution of maximum $1/2$. Cutting this elementary subgraph and using the induction hypothesis proves the theorem.

Thus using Gelfand's Formula, the norm operator is equal to $n^{3/2}$. Which gives an recovery threshold for the melon of $\beta = n^{3/4}$ using Wedin perturbation.

For the symmetric case, since $T^{sym}_{ijk} = \sum_{\tau \in \mathfrak{S}_k} T_{\tau(i)\tau(j)\tau(k)}$ where $\mathfrak{S}_k$ is the symmetric group introduced previously. We just need to expand the melonic matrix $T^{sym}_{i_1jk} T^{sym}_{i_2jk}$ 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 6 : Using the matrix decomposition illustrated in 18, we can see that we have a sum of four matrices of rank one. Using the usual formula to calculate the expectation of the operational norm of the matrices we obtain:

- $\left\|\boldsymbol{M}^{(N)}\right\| = n^2$
- $\left\|\boldsymbol{M}^{(D)}\right\| = n$
- $\left\|\boldsymbol{M}^{(R)}\right\| = n^{3/2}$

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.

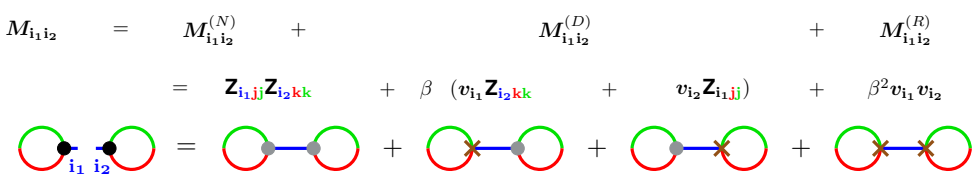


Figure 18: Decomposition of a tadpole matrix

Proof theorem 7 : 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:

$$\begin{aligned}&\sum_{i=1}^{r}\left(\frac{s_i}{2}+\frac{s_i(s_i-2)}{8}+\frac{s_i(s_i-2)}{12}+\frac{s_i(k+1-s_i)}{3}+\frac{(k+1-s_i)(k-s_i)-2}{4}+1\right)\\ =&\sum_{i=1}^{r}\left(\frac{k(k+1)}{4}-s_i\frac{4k-3s_i-4}{24}+\frac{1}{2}\right).\end{aligned} \tag{14}$$

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:

$$\sum_{i=1}^{r}\left(\frac{k(k+1)}{4} - s_i\frac{4k-3s_i-4}{24} + \frac{1-\delta_{s_i,0}}{2}\right). \tag{15}$$

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

Proof of theorem 8 : Let's consider the melonic graph with the edge associated to the index of the space of dimension n_l open. It is associated to the matrix:

$$M_{pq} = T_{i_1\dots i_{l-1}pi_{l+1}\dots i_k}\, T_{i_1\dots i_{l-1}qi_{l+1}\dots i_k} \tag{16}$$

For simplicity the graphs will be drawn for $k = 3$.

We will treat two separate cases:

- Case $n_l \leq \prod_{i\neq l} n_i$:

  Let's prove by induction the statement: S_r: $\mathbb{E}[\mathrm{Tr}((\boldsymbol{M} - \mathbb{E}(\boldsymbol{M}))^r)] = O(n_l \prod_{i=1}^{k} n_i^{r/2})$.

  - Initialisation with $r = 2$: For $\boldsymbol{M} - \mathbb{E}(\boldsymbol{M})$ we have the following graph and its covering graphs:

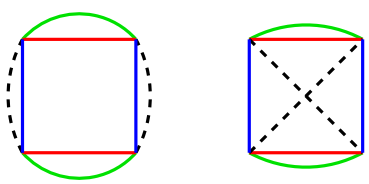


Figure 19: The two covering graphs of $\mathrm{Tr}\big(M^2\big)$

    which gives

$$\mathbb{E}(\mathrm{Tr}(M^r) = n_l \prod_{i=1}^{k} n_i + \prod_{i=1}^{k} n_i \leq 2n_l \prod_{i=1}^{k} n_i^{r/2}$$

  - Assume S_r is true for all $r' < r$, let's prove it for r.
    * In the case where there is no cycle of color l of length 1 (which means a propagator that shares the same ends that an edge of color l), then all edges have a contribution of $1/2$ at maximum. So the contribution of that covering graph will be at most equal to $\prod_{i=1}^{k} n_i^{r/2}$.
    * In case there is such a propagator, which means that there is a propagator coinciding with an edge of color l between two elementary subgraphs. Let r Then either the two other ends of the two elementary subgraphs are connected by a propagator like in 22 or not. In the first case, the total number of cycles will be at most equal to one cycle of each color, plus the number of cycles of a graph with $2(r-1)$ elementary subgraphs, so $\mathbb{E}(\mathrm{Tr}\big(M^{2r}\big) \leq 2n_l \prod_{i=1}^{k} n_i^{r}$.

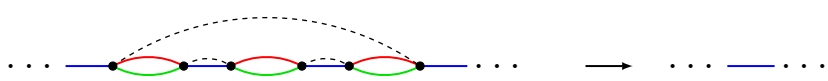


Figure 20: Elementary subgraphs connecting to the ementary subgraph next to it

    * In the second case, we can remove the cycle of one edge of color l and mix the two elementary subgraphs adjacent to it into one, we use then the induction hypothesis to show that Which proves the statement.

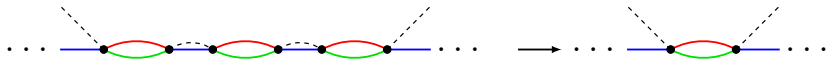


Figure 21: Elementary subgraphs connecting to the ementary subgraph next to it

- Case $n_l \geq \prod_{i \neq l} n_i$:

  Let's prove by induction the statement: S_r: $\mathbb{E}[\mathrm{Tr}((\boldsymbol{M} - \mathbb{E}(\boldsymbol{M}))^r)] = O(n_l^r \prod_{i \neq l} n_i)$.

  - Initialisation with $r = 2$: For $\boldsymbol{M} - \mathbb{E}(\boldsymbol{M})$ we have the covering graphs in 19 which checks S_2
  - Assume S_r is true for all $r' < r$, let's prove it for r.
    * In the case where there is no cycle of color l of length 1 (which means a propagator that shares the same ends that an edge of color l), then all edges have a contribution of $1/2$ at maximum. So the contribution of that covering graph will be at most equal to $\prod_{i=1}^k n_i^{r/2}$.
    * In case there is such a propagator, which means that there is a propagator coinciding with an edge of color l between two elementary subgraphs. Let r Then either the two other ends of the two elementary subgraphs are connected by a propagator like in 22 or not. In the first case, the total number of cycles will be at most equal to one cycle of each color, plus the number of cycles of a graph with $2(r-1)$ elementary subgraphs, so $\mathbb{E}(\mathrm{Tr}\big(M^{2r}\big) \leq 2n_l \prod_{i=1}^k n_i^r$.

Figure 22: Elementary subgraphs connecting to the ementary subgraph next to it

Proof of theorem 9 : Let $\tilde{\boldsymbol{A}} = \boldsymbol{M}_{\mathcal{G}}^e(\mathbf{T})$ the matrix we have with noise and $\boldsymbol{A} = \sum_{i=1}^p \beta_i^d \boldsymbol{v}_i^k$ the matrix when there is no noise and only signal. We call an eigenspace of $\boldsymbol{A}$ the subspace of $\mathbb{R}^n$ spanned by some eigenvectors of $\boldsymbol{A}$.

The idea of the proof of the theorem is that for β large enough, the eigenspace spanned by the signal vectors S should be close to the eigenspace spanned by the p largest eigenvectors of $\tilde{\boldsymbol{A}}$ denoted $\tilde{S}$. Then, we take a vector from this subspace $\tilde{S}$, we perform power iteration on it in order to recover a signal vector w_1, and repeat this step by restricting to the subspace of $\tilde{S}$ orthogonal to w_1.

If there were only one signal, there is a simple way to calculate how close are v_1 and w_1: $\theta(v_1, w_1) = \cos^{-1}(v_1.w_1)$. To quantify the closeness of two subspaces S and $\tilde{S}$, we use the canonical angles. The first canonical angle is defined as follow:

$$\theta_1 = \cos^{-1}(\max_{\substack{x \in S \\ \|x\|=1}} \max_{\substack{y \in \tilde{S} \\ \|y\|=1}} |x.y|) = \cos^{-1}(x_1.y_1) \tag{17}$$

where we denoted the maximizers by x_1 and y_1 and $x.y$ denotes the scalar product between x and y.

Then we define by iteration the kth canonical angle by:

$$\theta_k = \cos^{-1}(\max_{\substack{x \in S \\ \|x\|=1 \\ x.x^{k'}=0 \\ \forall k' \in \{1,\dots,k-1\}}} \max_{\substack{y \in \tilde{S} \\ \|y\|=1 \\ y.y^{k'}=0 \\ \forall k' \in \{1,\dots,k-1\}}} |x.y|) = \cos^{-1}(x_k.y_k) \tag{18}$$

We put these angles in a matrix Θ

$$\Theta = \begin{pmatrix} \theta_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \theta_r \end{pmatrix} \tag{19}$$

We define now the distance between S and $\tilde{S}$ by $\|\sin(\Theta)\|_F$ with $\sin(\Theta)$ defined as:

$$\sin(\Theta) = \begin{pmatrix} \sin(\theta_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sin(\theta_r) \end{pmatrix} \tag{20}$$

The Davis-Kahan Theorem uses these tools to provide a bound on subpaces spanned by eigenvectors under perturbation (Stewart & guang Sun (1990)):

Davis-Kahan: Let $\boldsymbol{A}$, $\boldsymbol{A}'$ two symmetric matrices with respective eigenvectors:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r$$
$$\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \dots \geq \tilde{\lambda}_r$$

Let's define the eigengap as:

$$\delta = \inf\{\left|\lambda - \tilde{\lambda}\right| : \lambda \in [\lambda_s, \lambda_r], \tilde{\lambda} \in (-\infty, \tilde{\lambda}_{s+1}] \cup [\tilde{\lambda}_{s+1}, \infty)\} \tag{21}$$

Then if $\delta > 0$

$$\left\|\sin\Theta(S, \tilde{S})\right\|_F \leq \frac{\left\|\tilde{\boldsymbol{A}} - \boldsymbol{A}\right\|_F}{\delta} \tag{22}$$

The result also holds for $\|.\|_{\text{op}}$

Let's study the case where $\beta_d > cn^{k/4}$ where c is a constant large enough. The gap will be equal to $\delta = (c-1)n^{k/4}$. In the case of the melonic graph, $\left\|\tilde{\boldsymbol{A}} - \boldsymbol{A}\right\|_{op}$ can be bounded by $2n^{k/4}$. Let's take a vector z from $\tilde{S}$. It can be put in the form of:

$$\begin{aligned} z = \sum_{i=1}^{p} \alpha_i \boldsymbol{y}_i &= \sum_{i=1}^{p} \alpha_i \boldsymbol{x}_i + \alpha_0 \boldsymbol{q}' \\ &= \sum_{i=1}^{p} \gamma_i \boldsymbol{v}_i + \gamma_0 \boldsymbol{q} \end{aligned} \tag{23}$$

where $\boldsymbol{q}'$ and $\boldsymbol{q}$ are unitary vectors, and $\boldsymbol{q} = \boldsymbol{q}' - \sum_{i=1}^{p} (\boldsymbol{q}'.\boldsymbol{x}_k)\boldsymbol{x}_k$ so it is orthogonal to all the vectors $\boldsymbol{v}_i$ and that $|\gamma_0| \leq |\alpha_0|$. Let's put a high bound for $alpha_0$. Since $x_k.y_k = \cos(\theta_k),\ \forall k \in \{1, \dots, p\}$

$$\begin{aligned} |\alpha_0| &= \left\|\sum_{i=1}^{p} \alpha_i \boldsymbol{y}_i - \sum_{i=1}^{p} \alpha_i \boldsymbol{x}_i\right\| \\ &\leq p \max_k |\sin(\theta_k)| \\ &\leq p\left\|\sin\Theta(S, \tilde{S})\right\|_{\text{op}} \end{aligned} \tag{24}$$

Using these results, we show that power iteration on a vector z of S is able to recover a signal vector $\boldsymbol{v}_k$:

$$\begin{aligned} \mathsf{T}_{ijk} z_j z_k &= \left(\mathsf{Z}_{ijk} + \sum_l \beta_l\ \boldsymbol{v}_l^{\otimes k}\right) z_j z_k \\ &= \mathsf{Z}_{ijk} z_j z_k + \sum_l \beta_l \gamma_l^2\ \boldsymbol{v}_l \end{aligned} \tag{25}$$

Thus, repeating $\boldsymbol{z} \leftarrow \mathbf{T}\boldsymbol{z}\boldsymbol{z}$ will converge to $\boldsymbol{v}_{l_{\max}}$ with $l_{\max} = \max(\{\beta_l \gamma_l^2, l \in \{1, \dots, r\}\})$

We repeat this step p times, but before each step we use deflation by computing:

$$\mathbf{T} \leftarrow \mathbf{T} - (\mathbf{T}\boldsymbol{z}_k\boldsymbol{z}_k\boldsymbol{z}_k)\boldsymbol{z}_k^{\otimes k}$$

.

Proof Lemma 10 The covering graphs involved in the calculation of $E(\mathrm{Tr}(\boldsymbol{M}-\mathbb{E}(\boldsymbol{M}))^r)$ are the same that are involved in the calculation of $E(\mathrm{Tr}(\boldsymbol{M}^r))$ minus the covering graphs where one of the elementary subgraphs has only internal propagators.

We can first begin by showing that if we have two matrices $\boldsymbol{M}_1$ of degree d_1 and $\boldsymbol{M}_2$ of degree d_2, then the covering graphs involved in $\mathrm{Tr}((\boldsymbol{M}_1-\mathbb{E}(\boldsymbol{M}_1))\boldsymbol{M}_2)$ are the covering graphs of $\mathrm{Tr}(\boldsymbol{M}_1\boldsymbol{M}_2)$ minus the ones where the propagators are internal. Let's denote the color of the open edge by l. To distinguish the d_1+d_2 vertices and their indices, we assign numbers to each vertex. Namely the vertices of $\boldsymbol{M}_1$ are denoted by $1,\dots,d_1$ with 1 and 2 associated to the two vertices adjacent to the open edge of $\boldsymbol{M}_1$, and $d_1,\dots,d_1+d_2$ where the vertices of the open edge of $\boldsymbol{M}_2$ are d_1+1, d_1+2.

We denote all the indices by a, and the indices of all colors associated to a vertex l by $a_l^{\mathcal{D}}$.

A subset of the symmetric group $\sigma(d_1+d_2)$ are the permutations which could be written as $\tau_1\tau_2$ where τ_1 is a permutation of $(1,\dots,d_1)$ and τ_2 is a permutation of $(d_1,\dots,d_1+d_2)$.

We denote by the symbol $\delta_{aa}^{\mathcal{B}}$ the δ of all the indices that are contracted in the graph.

Let $\mathcal{B}_1$ the graph associated to $\boldsymbol{M}_1$ and $\mathcal{B}_2$ the graph associated to $\boldsymbol{M}_2$.

Let $\mathcal{B}'$ the subgraph of $\mathcal{B}$ obtained by opening an edge of color c corresponding to the vertices i and j, it is an elementary subgraph of $\mathcal{B}^r$, the graph corresponding to $\mathrm{Tr}(M^r)$. We have

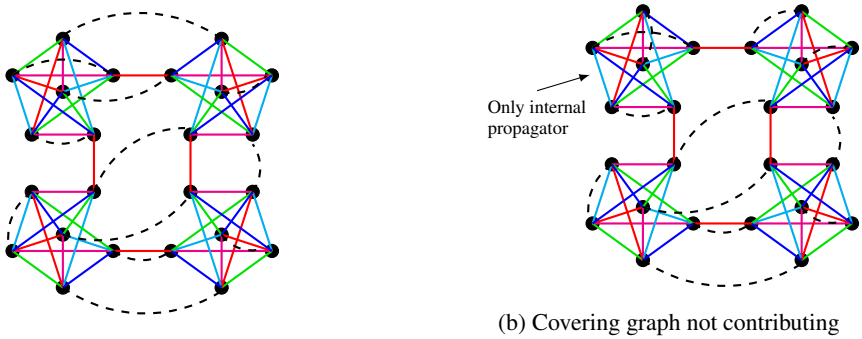
$$\mathbb{E}(\boldsymbol{M}_1)_{ij} = \delta_{a_1{}^l,i}\delta_{a_2{}^l,j}\sum_{\substack{a\\ a\neq a_1{}^l,a_2{}^l}}\delta_{aa}^{\mathcal{B}_1}\sum_{\tau\in\mathfrak{S}(d_1)}\prod_{l=1}^{d_1}\delta_{a_l{}^{\mathcal{D}}a_{\tau(l)}^{\mathcal{D}}}, \tag{26}$$

$$\mathbb{E}(\mathrm{Tr}(\boldsymbol{M}_1\boldsymbol{M}_2)) = \sum_a \delta_{aa}^{\mathcal{B}}\sum_{\tau\in\mathfrak{S}(d_1+d_2)}\prod_{l=1}^{d_1+d_2}\delta_{a_l{}^{\mathcal{D}}a_{\tau(l)}^{\mathcal{D}}}, \tag{27}$$

Since the edges of $\mathcal{B}$ are the union of the edges of $\mathcal{B}_1$ and $\mathcal{B}_1$ and an edge of color l connecting the vertices of the open edges: 1 with d_1+1 and 2 with d_1+2, we have: $\delta_{aa}^{\mathcal{B}} = \delta_{aa}^{\mathcal{B}^1}\delta_{aa}^{\mathcal{B}^2}\delta_{a_1^l,a_{d_1+1}^l}\delta_{a_2^l,a_{d_1+2}^l}$. This let us write:

$$\begin{aligned}\mathbb{E}(\mathrm{Tr}((\boldsymbol{M}_1-\mathbb{E}(\boldsymbol{M}_1))\boldsymbol{M}_2)) = \sum_a \delta_{aa}^{\mathcal{B}}\Bigg[&\sum_{\tau\in\mathfrak{S}(d_1+d_2)}\prod_{l=1}^{d_1+d_2}\delta_{a_l{}^{\mathcal{D}}a_{\tau(l)}^{\mathcal{D}}}\\ &-\sum_{\tau_1\in\mathfrak{S}(d_1),\tau_2\in\mathfrak{S}(d_2)}\prod_{l=1}^{d_1}\delta_{a_l{}^{\mathcal{D}}a_{\tau_1(l)}^{\mathcal{D}}}\prod_{m=d_1+1}^{d_1+d_2}\delta_{a_m{}^{\mathcal{D}}a_{\tau_2(m)}^{\mathcal{D}}}\Bigg],\end{aligned} \tag{28}$$

Thus the only permutations that are not cancelled are the permutations that does not live invariant the subset $\{1,\dots,d_1\}$, which means that $\exists i\in\{1,\dots,d_1\}$ such that $\tau(i)\in\{d_1+1,\dots,d_1+d_2\}$. These permutations correspond to the covering graphs that does not have only internal propagators connecting the vertices of $\boldsymbol{M}_1$. This proves our claim.



(a) Covering graph contributing

(b) Covering graph not contributing to $\mathbb{E}(\mathrm{Tr}((\boldsymbol{M} - \mathbb{E}(\boldsymbol{M}))\boldsymbol{M}^r))$

Figure 23: Complete graph for $k = 5$

Proof of lemma 11 : Let $\mathcal{G}$ a graph that has $\beta_{\text{dec}} = n^{k/4}$ as a detection threshold. Let's assume that an intermediate graph $\mathcal{G}^{\text{in}}$ with r spikes dominate for $\beta > \beta_{\text{dec}}$ and let's take the covering graph of ${\mathcal{G}^{\text{in}}}^2$ which gives the contribution for the domination. We cut the the propagators that involves the spikes and connect them together, while we connect by a new propagator each vertex of spike to its mirror. This covering graph will give a strictly superior contribution to the expectation if the graph is connected since:

- The cycles that didn't involve a spike stay unchanged
- The cycles that involved a spike adds together a new contribution of n^{α} with $\alpha > 3r/4$.

Thus for $n^{3/4} << \beta << n^{\alpha}$, the pure noise graph dominates the intermediate graph, which leads to a contradiction. The demonstration for the matrices $\boldsymbol{M}$ follows the same steps but instead of $\mathcal{G}^2$, one has to consider $\mathrm{Tr}(\boldsymbol{M}^r)$.