ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Aligning random graphs with a sub-tree similarity message-passing algorithm

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Piccioli G., Semerjian G., Sicuro G., Zdeborova L. (2022). Aligning random graphs with a sub-tree similarity message-passing algorithm. JOURNAL OF STATISTICAL MECHANICS: THEORY AND EXPERIMENT, 2022(6), 1-44 [10.1088/1742-5468/ac70d2].

Availability:
This version is available at: https://hdl.handle.net/11585/958757 since: 2024-02-17

Published:
DOI: http://doi.org/10.1088/1742-5468/ac70d2

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

12 October 2026

This is the final peer-reviewed accepted manuscript of:

Piccioli, G., Semerjian, G., Sicuro, G., Zdeborová, L. Aligning random graphs with a sub-tree similarity message-passing algorithm (2022) Journal of Statistical Mechanics: Theory and Experiment, 2022 (6), art. no. 063401

The final published version is available online at: ***https://dx.doi.org/10.1088/1742-5468/ac70d2***

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

arXiv:2112.13079v2 [cs.IT] 4 May 2022

Aligning random graphs with a sub-tree similarity message-passing algorithm

Giovanni Piccioli[1], **Guilhem Semerjian**[2], **Gabriele Sicuro**[3], **Lenka Zdeborová**[1]

[1]SPOC Lab, EPFL, Rte Cantonale, 1015 Lausanne, Switzerland
[2]Laboratoire de Physique de l'École Normale Supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université Paris Cité, F-75005 Paris, France
[3] King's College London, Strand, WC2R 2LS London, United Kingdom

Abstract. The problem of aligning Erdős–Rényi random graphs is a noisy, average-case version of the graph isomorphism problem, in which a pair of correlated random graphs is observed through a random permutation of their vertices. We study a polynomial time message-passing algorithm devised to solve the inference problem of partially recovering the hidden permutation, in the sparse regime with constant average degrees. We perform extensive numerical simulations to determine the range of parameters in which this algorithm achieves partial recovery. We also introduce a generalized ensemble of correlated random graphs with prescribed degree distributions, and extend the algorithm to this case.

1. Introduction

The *graph alignment problem* (GAP) is a classical combinatorial optimization problem consisting in finding a bijection between the vertex sets of two graphs in such a way that their edge sets are maximally aligned. To make this statement more precise, let us denote $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ the (possibly weighted) adjacency matrices of two graphs having the same number n of vertices. The goal is to find the permutation $\hat{\pi} \in \mathcal{S}_n$, with $\mathcal{S}_n$ the set of the permutations of n elements, such that $\hat{\pi} = \arg\max_{\pi\in\mathcal{S}_n} \sum_{i<j} A_{ij}B_{\pi(i)\pi(j)}$. The wide interest in this problem is due to the large number of applications involving the solution of a GAP, from pattern recognition [1] to network de-anonymization [2, 3] or alignment of molecular and protein-interaction networks in biology [4–7]. The GAP has also been used as a prototypical and challenging problem to evaluate the performance of graph neural networks [8, 9].

In the general formulation above the GAP is also known under the name of *quadratic assignment problem* [10], which belongs to the computational class of NP-hard problems (although some special settings allow for a polynomial-time solution [10]). This worst-case hardness result leaves open the possibility that some "typical" instances are efficiently solvable. To give a precise meaning to this notion of typicality a number of studies focused therefore on the alignment of pairs of graphs obtained from some random ensembles. In these ensembles, each pair is generated with the same vertex set of cardinality n according to probabilistic rules implying some correlations between the two graphs, then the information of the vertex correspondence is removed by a random reshuffling $\boldsymbol{\pi}_\star$ of the labels of one of the graphs. As a result, this *planted* GAP takes the form of an inference problem in which the planted permutation $\boldsymbol{\pi}_\star$ has to be, at least approximately, recovered. The goal of recovering the permutation $\boldsymbol{\pi}_\star$ rather than maximally aligning the graphs also stems from applications where such a ground truth permutation is often assumed to exist, and the inference (rather than the optimization) version

of the GAP problem is thus our interest in this paper. We may wonder at this point if the exact, or perfect, recovery of $\boldsymbol{\pi}_\star$ is achievable with high probability over the ensemble samples. We can also ask, less ambitiously, if a *partial* recovery of $\boldsymbol{\pi}_\star$ is feasible with finite probability, i.e., if it is possible to recover the correct matching of a finite fraction of vertices. Another question concerns the possibility of detecting the correlations between the graphs, namely to distinguish between a sample of the correlated ensemble and one made of two independent graphs. These questions have been studied in a series of theoretical works, for different relevant graph ensembles, in the limit of large graph sizes $n \to +\infty$.

In this contribution we will mostly focus on the correlated Erdős–Rényi ensemble $\mathrm{G}(n, \lambda/n, s)$. This ensemble, that we will detail in Section 2.1, was introduced in [3] in the context of de-anonymization of social networks. Here we anticipate that an element of this ensemble is given by a pair of correlated Erdős–Rényi graphs, both with the same average degree λ, on the same set of n vertices: the parameter $0 \leq s \leq 1$ measures the degree of correlation, so that $s = 1$ corresponds to the case of identical graphs and $s = \lambda/n$ corresponds to a pair of independently generated Erdős–Rényi graphs. After the generation, the labels of one of the two graphs are reshuffled by a permutation $\boldsymbol{\pi}_\star$ to be recovered. The answers to the questions raised above on the possibility of exact recovery, partial recovery, and detection, depend on the scaling of the parameters λ and s with the size n of the graphs. Cullina and Kiyavash [11] studied the exact recovery question in the regime of diverging degrees, showing that it is possible to exactly recover $\boldsymbol{\pi}_\star$ if and only if $\lambda s - \ln n \to +\infty$ as $n \to +\infty$. The range of parameters for which polynomial time algorithms succeed in this exact recovery task have been progressively improved in [12–14], covering the case where the average degree λ is slightly greater than $\ln n$ and s is a constant sufficiently close to 1. On the other hand, it is impossible to exactly recover $\boldsymbol{\pi}_\star$ for $\lambda = O(1)$ [11]. For this reason the authors of [15–18] focused in this regime on the possibility of a *partial* recovery of the true labelling. In [17] it is shown that the fraction of correctly matched pairs of vertices is upper-bounded by $c(\lambda s)$, the largest non-negative solution of the equation $1 - c = \mathrm{e}^{-\lambda s c}$, for any statistical estimator. This implies that for $\lambda s \leq 1$ partial recovery is *information-theoretically infeasible*, i.e., for this set of parameters it is not possible to recover $\boldsymbol{\pi}_\star$, not even partially. On the other hand, in [16] it is proven instead that for $\lambda s > 4$ partial recovery is information-theoretically feasible (improving on a previous bound in [19]). It has also been shown in [15, 18] that there exists a *polynomial-time feasible phase* in the region $\lambda s > 1$ (for large enough values of s). All these results leave open a complete determination of the phase diagram of the partial recovery problem in the (λ, s) plane, namely the boundaries of the impossible, easy (meaning feasible in polynomial time) and hard (information-theoretically feasible but in an a priori exponential time under some computational complexity hypothesis) phases. The existence of an hard phase for large enough λ follows from the bounds of [16, 18], but its precise boundary is not accurately known.

Let us also mention other ensembles on which the graph alignment problem has been studied. Correlated random graphs with a hidden community structure have been considered in [20, 21] in the context of network de-anonymization problems. They are built by first generating a "parent graph" using the stochastic block model (SBM), in which the vertices bear some labels interpreted as communities, the probability of presence of an edge between two vertices depending on their labels. Then two graphs are obtained from it by randomly removing edges from the parent one with probability $1 - s$, independently for each of the two copies, and reshuffling the vertex labels of one of the graphs. The information-theoretical possibility of exact recovery in this ensemble has been characterized in [21] for the regime of degrees logarithmic in n. Another variant of the problem concerns the alignment of correlated random matrices, which corresponds to the case of weighted complete graphs [12, 13, 16, 22, 23]: one draws a pair of correlated random matrices $(\mathbf{A}, \mathbf{C})$ such that, independently for all $i < j$, $\mathbf{A}_{ij}$ and $\mathbf{C}_{ij}$ are standard Gaussian variables

with correlation coefficient $s \in [0,1]$. The observed pair is obtained by reshuffling the rows and columns of one of the matrices, $\mathbf{C}_{ij} = \mathbf{B}_{\boldsymbol{\pi}_\star(i)\,\boldsymbol{\pi}_\star(j)}$, and the goal is to recover the uniformly random permutation $\boldsymbol{\pi}_\star$ from the observation of $(\mathbf{A}, \mathbf{B})$. In [12, 13] it is shown that a spectral (polynomial-time) algorithm exactly recovers $\boldsymbol{\pi}_\star$ if $\sqrt{1-s^2} = O(\ln^{-1} n)$. On the other hand, in [16, 23] it is proven that it is information-theoretically possible to recover $\boldsymbol{\pi}_\star$ if $s^2 \geq \frac{c \ln n}{n}$ for some $c > 4$: this suggests that in this problem there might be a wide *hard phase* in which polynomial-time algorithms cannot exactly recover $\boldsymbol{\pi}_\star$.

In this paper we develop a message-passing strategy for the graph alignment problem on correlated random graphs of constant degrees. It is substantially different from the message-passing algorithms of [24, 25], that were based on a belief-propagation approximation of the posterior distribution on the unknown permutation, and whose convergence required some side-information or the use of sophisticated numerical tricks (decimation, reinforcement and/or the introduction of biases). Instead we use crucially the locally tree-like character of the random graphs to compute a *score* for each pair of vertices of the two graphs to align based on the similarity of their neighborhoods, that quantifies how likely two vertices were matched through $\boldsymbol{\pi}_\star$ and allows thus to build an estimator of this unknown permutation achieving partial recovery in some portion of the parameter space. While we were working on this project we became aware of the independent work of [18], which followed a very similar reasoning and contains an essentially equivalent algorithm. Our derivation is, however, slightly different, and we believe the extensive numerical simulations we present are a useful complement to the rigorous bounds of [18].

The rest of the paper is organized as follows. In Section 2 we introduce the ensemble $\mathrm{G}(n, \lambda/n, s)$ that is going to be the main object of our analysis, and we discuss the local properties of pairs of correlated graphs drawn from this ensemble. In Section 3 we present the Bayesian formulation of the inference problem and the message-passing algorithm obtained via a truncation of the posterior distribution. The results of the numerical experiments on this algorithm are discussed in Section 4, along with comparisons with the known theoretical bounds. Section 5 is devoted to the liming tree problem that arises from the analysis of the algorithm. Finally, in Section 6 we draw our conclusions. In Appendix A we introduce a generalized ensemble of correlated random graphs with prescribed degree distributions and extend the message-passing algorithm to this ensemble (considering also the case of weighted graphs). Further results of the numerical experiments are presented in Appendix B.

Throughout the paper we will denote $[l] \coloneqq \{1, \dots, l\}$, and we will use bold fonts for random variables, a notation that we anticipated throughout this introduction.

2. Correlated Erdős-Rényi random graphs

2.1. Definition

Let us start by defining the correlated Erdős-Rényi (ER) random graph ensemble [3] denoted $\mathrm{G}(n, \lambda/n, s)$, that depends on two real parameters, $\lambda > 0$ and $s \in [0,1]$. An element of this ensemble is a pair $(\mathbf{A}, \mathbf{C})$ of random graphs on a common vertex set $V = [n]$, defined by their adjacency matrices $\mathbf{A}$ and $\mathbf{C}$ (here and in the following we use for simplicity the same notation for a graph and its adjacency matrix) generated as follows: independently for each of the $\frac{n}{2}(n-1)$ pairs of vertices $i < j$,

- $\mathbf{A}_{ij} = \mathbf{C}_{ij} = 1$ with probability $\frac{\lambda}{n} s$;
- $\mathbf{A}_{ij} = 1$, $\mathbf{C}_{ij} = 0$ with probability $\frac{\lambda}{n}(1-s)$;
- $\mathbf{A}_{ij} = 0$, $\mathbf{C}_{ij} = 1$ with probability $\frac{\lambda}{n}(1-s)$;

- $\mathbf{A}_{ij} = \mathbf{C}_{ij} = 0$ with probability $1 - \frac{\lambda}{n}(2 - s)$.

The diagonal elements of the adjacency matrices are set to zero, $\mathbf{A}_{ii} = \mathbf{C}_{ii} = 0$ (there is no self-loop in the graphs), and the adjacency matrices are completed by symmetry, $\mathbf{A}_{ji} = \mathbf{A}_{ij}$ and $\mathbf{C}_{ji} = \mathbf{C}_{ij}$ (the graphs are undirected).

The marginal distributions of $\mathbf{A}$ and $\mathbf{C}$ are easily seen to coincide with the usual ER ensemble $\mathrm{G}(n, p = \lambda/n)$, where each possible edge is present with probability p. The parameter $s \in [0, 1]$ controls the correlation between the graphs $\mathbf{A}$ and $\mathbf{C}$: for $s = 1$, they are strictly identical, $\mathbf{A} = \mathbf{C}$, whereas for $s = \frac{\lambda}{n}$ they are independent. As we will concentrate on the large size limit $n \to \infty$ with λ and s kept fixed any $s > 0$ thus corresponds to a (positively) correlated situation.

We finally introduce a random permutation $\boldsymbol{\pi}_\star$ uniformly drawn from the symmetric group $\mathcal{S}_n$, and define a graph $\mathbf{B}$ as the image of $\mathbf{C}$ through the reshuffling of its vertices' labels by the permutation $\boldsymbol{\pi}_\star$. More explicitly, we define the adjacency matrix $\mathbf{B}$ as $\mathbf{B}_{ij} = \mathbf{C}_{\boldsymbol{\pi}_\star^{-1}(i),\boldsymbol{\pi}_\star^{-1}(j)}$, or equivalently $\mathbf{C}_{ij} = \mathbf{B}_{\boldsymbol{\pi}_\star(i),\boldsymbol{\pi}_\star(j)}$. The marginal distribution of $\mathbf{B}$ is then of course the same as $\mathbf{A}$ and $\mathbf{C}$, namely $\mathrm{G}(n, p = \lambda/n)$.

2.2. Local properties

2.2.1. Single ER graphs The analysis of the graph alignment problem presented in the rest of the paper will rely crucially on the local properties of the correlated random graph ensemble that we shall now present. As a first step let us discuss the case of a single ER graph drawn from the $\mathrm{G}(n, p = \lambda/n)$ ensemble (i.e., considering only $\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$). It is well-known that these graphs locally converge, in the large size limit $n \to \infty$, to Galton-Watson (GW) random trees with an offspring distribution given by the Poisson law of parameter λ, that we shall denote in the following $\mathrm{Po}(\lambda; l) = \mathrm{e}^{-\lambda}\frac{\lambda^l}{l!}$. To spell out more precisely the meaning of this statement we shall denote $\mathbf{A}_{i,d}$ the subgraph of $\mathbf{A}$ obtained by retaining the vertices that are at a distance smaller or equal than d from the reference vertex i, where the distance between two vertices is the minimal number of edges on a path linking them. For an arbitrary choice of i, and for any fixed d, $\mathbf{A}_{i,d}$ is a tree T rooted in i with high probability (w.h.p.), meaning with a probability going to 1 when $n \to \infty$. Moreover the law of T corresponds to the first d generations of a GW branching process: the root i has a number l of descendents drawn from the probability law $\mathrm{Po}(\lambda; l)$, each of them having an independent number of offsprings with the same law, and so on and so forth until the d-th generation has been reached (or until the branching process gets extinct). We shall denote $\mathbb{P}_0^{(d)}[T]$ the probability that a given tree T is generated in this way, that admits the following recursive decomposition:

$$\mathbb{P}_0^{(d)}[T] = \mathrm{Po}(\lambda; l) \prod_{j=1}^{l} \mathbb{P}_0^{(d-1)}[T_j] \ , \tag{1}$$

for a tree T whose root has l descendents which are themselves the roots of the l subtrees $(T_1, \ldots, T_l)$, and with the convention $\mathbb{P}_0^{(0)} = 1$. To be precise, here and in the following all trees are understood to be rooted and labelled, and we consider two trees as equal if they are related by a relabelling that preserves the parent-offspring relationships.

Without entering into a formal proof of this local convergence property let us sketch its justification, which will be useful to address the generalization to correlated pairs of graphs. As the edges of an ER graph are independently present with probability λ/n, the degree of a vertex i has a binomial distribution with parameters $(n - 1, \lambda/n)$, that converges to $\mathrm{Po}(\lambda)$ as $n \to \infty$. Once the edges absent and present around i have been revealed, one can continue the exploration process by exposing the edges adjacent to the neighbors of i, let us call them $\{i_1, \ldots, i_l\}$. The number of neighbors of i_1 distinct from i is binomial with parameters $(n - 2, \lambda/n)$, that again

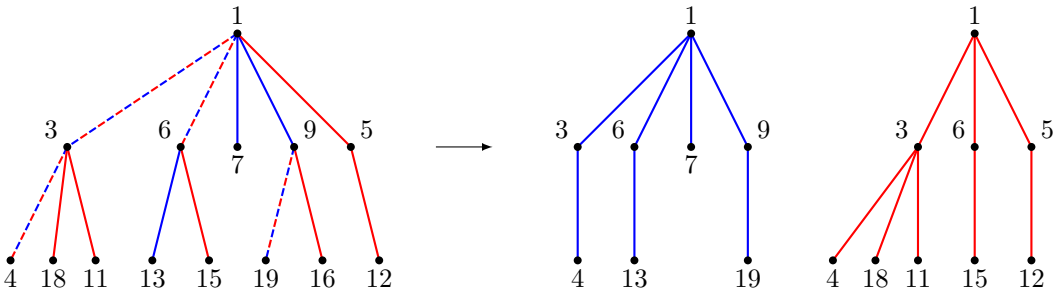

Figure 1. Left: an example of the colored GW tree $\mathcal{T}$ obtained by a local exploration of the aligned graphs represented by $\mathbf{G}$, starting from the vertex $i = 1$, up to distance 2. Right: the corresponding neighborhoods of $i = 1$ in the two graphs $\mathbf{A}$ and $\mathbf{C}$, $(T, T') = (b(\mathcal{T}), r(\mathcal{T}))$. Note that the red and bicolored edges below the vertex 9 do not appear in the red tree, because this vertex has been reached by a blue edge in $\mathcal{T}$. The neighborhoods of i in $\mathbf{A}$ and $i' = \boldsymbol{\pi}_\star(i)$ in $\mathbf{B}$ are isomorphic to these two trees, with a random relabelling of the vertices.

converges to $\mathrm{Po}(\lambda)$ as $n \to \infty$. Moreover the probability that there is an edge between any two vertices in $\{i_1, \dots, i_l\}$ is of order $1/n$, as there is a finite number of possible edges between them, each being present with probability λ/n. This reasoning can be extended to the exploration of the neighborhood of i up to any fixed distance d, as the number of revealed vertices remains finite while $n \to \infty$, hence all binomial of parameters $(n - o(n), \lambda/n)$ converge to $\mathrm{Po}(\lambda)$, and the probability of an edge being present among a fixed number of vertices being of order $1/n$.

2.2.2. Aligned pairs of graphs Let us now come back to the correlated graph model, and discuss the local properties of the pair $(\mathbf{A}, \mathbf{C})$, i.e., the aligned graphs before the reshuffling of their vertices through the random permutation. It is convenient to represent this pair of graphs by a single graph $\mathbf{G}$ whose edge bear a three-valued label represented as a color. Consider that between the vertices i and j there is in $\mathbf{G}$:

- a blue edge if $\{i, j\}$ is present in $\mathbf{A}$ but not in $\mathbf{C}$;
- a red edge if $\{i, j\}$ is present in $\mathbf{C}$ but not in $\mathbf{A}$;
- a bicolored edge if $\{i, j\}$ is present in both $\mathbf{A}$ and $\mathbf{C}$;
- no edge otherwise.

It is clear that this colored graph $\mathbf{G}$ contains exactly the same information as the pair $(\mathbf{A}, \mathbf{C})$. Suppose now that one performs a local exploration of $\mathbf{G}$ starting from a vertex i, up to a distance d, or in other words that one explores simultaneously both graphs $\mathbf{A}$ and $\mathbf{C}$ from i, following edges that are present in at least one of the two graphs. The arguments invoked to justify the local convergence of a single ER graph can be immediately generalized to this case, and show that with high probability when $n \to \infty$ with d fixed the resulting neighborhood in $\mathbf{G}$ is a colored GW tree (or multi-type branching process) that we shall denote $\mathcal{T}$. More precisely, this tree can be built recursively, each vertex having a number of offsprings linked to it by a blue (resp., red, bicolored) drawn as independent Poisson random variables of parameter $\lambda(1 - s)$ (resp., $\lambda(1 - s)$, λs), see the left panel of Fig. 1 for an illustration. Thanks to the memoryless property of Poissonian distributions each subtree has the same law as $\mathcal{T}$.

2.2.3. Pairs of local neighborhoods with aligned roots We move now to the pair $(\mathbf{A}, \mathbf{B})$ of disaligned graphs, and consider the following question, whose motivation will be unveiled later on: what is the joint law of (T, T'), where T (resp. T') is the depth d neighborhood of a vertex

i in the graph $\mathbf{A}$ (resp. of i' in $\mathbf{B}$), when i and i' are aligned vertices (i.e. when $i' = \boldsymbol{\pi}_\star(i)$), with d fixed and $n \to \infty$? This question is obviously related to the exploration process on $\mathbf{G}$ described above, but with some important differences. Indeed T (resp. T') is built by following only the blue and bicolored (resp., red and bicolored) edges of $\mathbf{G}$, hence it is obtained from the colored GW tree $\mathcal{T}$ by keeping only the blue and bicolored (resp. red and bicolored) edges that form the connected component of the root. We can thus define a map $(T, T') = (b(\mathcal{T}), r(\mathcal{T}))$ that transforms the colored tree $\mathcal{T}$ into a pair of monochromatic trees (T, T'), see the right panel and the caption of Fig. 1 for an illustration. In other words once a blue (resp., red) edge has been crossed in the colored GW tree one can draw its descendence as a blue (resp. red) GW tree with a Poisson offspring distribution of parameter λ. Moreover in the pair (T, T') the original alignment of the vertices is completely lost except for the root vertex. From these observations one can establish a recursive relation for the law of (T, T'), to be denoted $\mathbb{P}_1^{(d)}[T, T']$. Denoting l (resp., l') the degree of the root of T (resp., T') and $T_1, \dots, T_l$ (resp., $T'_1, \dots, T'_{l'}$) the subtrees rooted at its descendents, one has

$$\begin{aligned}\mathbb{P}_1^{(d)}[T,T'] = \sum_{l_{\rm b},l_{\rm r},l_{\rm bi}=0}^{\infty} &\mathrm{Po}(\lambda(1-s);l_{\rm b})\,\mathrm{Po}(\lambda(1-s);l_{\rm r})\,\mathrm{Po}(\lambda s;l_{\rm bi})\,\mathbb{I}(l=l_{\rm b}+l_{\rm bi})\;\mathbb{I}(l'=l_{\rm r}+l_{\rm bi})\\ &\sum_{Q_1,\dots,Q_{l_{\rm b}}}\sum_{R_1,\dots,R_{l_{\rm r}}}\sum_{S_1,S'_1,\dots,S_{l_{\rm bi}},S'_{l_{\rm bi}}}\prod_{i=1}^{l_{\rm b}}\mathbb{P}_0^{(d-1)}[Q_i]\prod_{i=1}^{l_{\rm r}}\mathbb{P}_0^{(d-1)}[R_i]\prod_{i=1}^{l_{\rm bi}}\mathbb{P}_1^{(d-1)}[S_i,S'_i]\\ &\frac{1}{l!l'!}\sum_{\pi,\pi'}\mathbb{I}((T_1,\dots,T_l)=\pi(Q_1,\dots,Q_{l_{\rm b}},S_1,\dots,S_{l_{\rm bi}}))\mathbb{I}((T'_1,\dots,T'_{l'})=\pi'(R_1,\dots,R_{l_{\rm r}},S'_1,\dots,S'_{l_{\rm bi}}))\ ,\end{aligned}$$

where $\mathbb{I}(E)$ is the indicator function of the event E, π (resp., π') is a permutation of its l (resp., l') arguments, and we use the convention $\mathbb{P}_1^{(0)}[T, T'] = 1$. In this expression $l_{\rm b}$, $l_{\rm r}$ and $l_{\rm bi}$ are the number of blue, red and bicolored edges emerging from the root of the colored GW tree, the Q_i's (resp., R_i's) are the blue (resp., red) usual GW tree rooted at the offsprings reached by a blue (resp., red) edges, and the pairs of trees (S_i, S'_i) are those rooted at offsprings reached by a bicolored edge. The uniform average over the permutations π and π' arise from the ignorance of the vertex correspondance between the two graphs apart from the aligned root. This expression can be slightly simplified by noting that the relevant information contained in the permutations π and π' are the indices of the subtrees of T and T' assigned to the correlated pairs of trees (S_i, S'_i). This yields

$$\begin{aligned}\mathbb{P}_1^{(d)}[T,T'] = \sum_{l_{\rm bi}=0}^{\min(l,l')} &\mathrm{Po}(\lambda(1-s);l-l_{\rm bi})\,\mathrm{Po}(\lambda(1-s);l'-l_{\rm bi})\,\mathrm{Po}(\lambda s;l_{\rm bi})\\ &\frac{1}{\binom{l}{l_{\rm bi}}\binom{l'}{l_{\rm bi}}l_{\rm bi}!}\sum_{I,I',\sigma}\prod_{i\in I}\mathbb{P}_1^{(d-1)}[T_i,T'_{\sigma(i)}]\prod_{i\in[l]\setminus I}\mathbb{P}_0^{(d-1)}[T_i]\prod_{i\in[l']\setminus I'}\mathbb{P}_0^{(d-1)}[T'_i]\ ,\end{aligned}\tag{2}$$

where I (resp I') is a subset of $[l]$ (resp., of $[l']$) of $l_{\rm bi}$ elements, and σ a bijection from I to I'.

3. The inference problem

3.1. Estimators

The inference problem naturally associated with the correlated graph ensemble consists in aligning the graphs $\mathbf{A}$ and $\mathbf{B}$, in other words in retrieving the permutation $\boldsymbol{\pi}_\star$, and therefore

the original labeling in $\mathbf{C}$, from the sole observation of the two graphs $\mathbf{A}$ and $\mathbf{B}$. We will study this problem in a Bayesian setting, assuming that the procedure followed for the construction of these graphs is completely known to the observer. All the information available on $\boldsymbol{\pi}_\star$ given two observed adjacency matrices A and B is thus contained in its posterior probability distribution, that can be expressed thanks to the Bayes formula as

$$\mathbb{P}(\boldsymbol{\pi}_\star = \pi | \mathbf{A} = A, \mathbf{B} = B) = \frac{\mathbb{P}(\boldsymbol{\pi}_\star = \pi, \mathbf{A} = A, \mathbf{B} = B)}{\mathbb{P}(\mathbf{A} = A, \mathbf{B} = B)} \propto \mathbb{P}(\boldsymbol{\pi}_\star = \pi, \mathbf{A} = A, \mathbf{B} = B) \ , \tag{3}$$

where here and in the following the symbol $\propto$ implies the presence of a π-independent prefactor.

As in all inference problems the notion of optimal estimator depends on the properties required for the estimator, and on the quantitative measure of its distance to the groundtruth signal. In the present context where the groundtruth $\boldsymbol{\pi}_\star$ is a permutation of n elements, and where the estimator $\widehat{\pi} = \widehat{\pi}(A, B)$ has to be computed from the observed graphs A and B, one can envision different possible choices:

- If one requires the estimator to be a permutation and if the objective is to minimize the probability that it differs from $\boldsymbol{\pi}_\star$, then the optimal choice is $\widehat{\pi}(A, B) = \arg\max_\pi \mathbb{P}(\boldsymbol{\pi}_\star = \pi | \mathbf{A} = A, \mathbf{B} = B)$. As we concentrate in this paper on the constant degree regime where the exact recovery of $\boldsymbol{\pi}_\star$ is impossible this choice is not relevant here.
- One can view the groundtruth $\boldsymbol{\pi}_\star$ as an $n \times n$ matrix with $\{0, 1\}$ elements, a 1 in the i, i' entry encoding the fact that $\boldsymbol{\pi}_\star(i) = i'$. More precisely we can define the ground truth permutation matrix as $\boldsymbol{\Pi}_{\star ii'} = \mathbb{I}[\boldsymbol{\pi}_\star(i) = i']$, which is thus constrained to have exactly one 1 per row and per column. In this perspective one can consider an estimator $\widehat{\boldsymbol{\Pi}}$ that is a $\{0, 1\}$ matrix, without row and column sum constraints, and measure its accuracy in terms of the Hamming distance between $\boldsymbol{\Pi}_\star$ and $\widehat{\boldsymbol{\Pi}}$ (viewed as strings of n^2 bits). The optimal estimator is then

$$\widehat{\boldsymbol{\Pi}}(A, B)_{i,i'} = \begin{cases} 1 & \text{if} \quad \mathbb{P}(\boldsymbol{\pi}_\star(i) = i' | \mathbf{A} = A, \mathbf{B} = B) > \frac{1}{2} \\ 0 & \text{if} \quad \mathbb{P}(\boldsymbol{\pi}_\star(i) = i' | \mathbf{A} = A, \mathbf{B} = B) \leq \frac{1}{2} \end{cases} . \tag{4}$$

  Note that $P_{i,i'} = \mathbb{P}(\boldsymbol{\pi}_\star(i) = i' | \mathbf{A} = A, \mathbf{B} = B)$ is a bistochastic matrix, with both row and column sums equal to 1 (they correspond to sums of probabilities of disjoint events whose union is sure to occur), hence each row and column of P contains at most one element strictly larger than $1/2$. As a consequence the matrix estimator $\widehat{\boldsymbol{\Pi}}$ contains at most one nonzero entry per row and per column, but can leave some row i (or column i') equal to 0, i.e., do not propose any estimate for the vertex matched to i (or to i'). We shall come back on this estimator in Appendix B.2.
- The choice on which we will concentrate in most of the following is to require the estimator $\widehat{\pi} = \widehat{\pi}(A, B)$ to be a function from $[n]$ to $[n]$ (not necessarily a permutation), and to measure its quality in terms of the overlap with the groundtruth,

$$\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[\widehat{\boldsymbol{\pi}}(i) = \boldsymbol{\pi}_\star(i)] \ , \tag{5}$$

  which gives the fraction of vertices in $\mathbf{A}$ that are correctly assigned their matching vertices in $\mathbf{B}$. The optimal estimator, in the sense of maximizing this average overlap, is achieved by taking

$$\widehat{\pi}(A, B)(i) = \arg\max_{i'} \mathbb{P}(\boldsymbol{\pi}_\star(i) = i' | \mathbf{A} = A, \mathbf{B} = B) \ . \tag{6}$$

In intuitive terms this corresponds to compute a $n \times n$ matrix of "scores" giving the posterior probability of the event that i was matched to i' given the observations of the graphs A and B, and for each i choosing the i' with the highest score.
Two other alternative estimators will be described and evaluated for comparison and complementarity in Appendix B.2.

3.2. A local approximation of the posterior

The optimal estimators discussed above rely on finding the maximum of the posterior probability of $\boldsymbol{\pi}_\star$, or on computing the probability that the posterior gives to the event $\boldsymbol{\pi}_\star(i) = i'$. Unfortunately these tasks are computationally intractable: the joint distribution $\mathbb{P}(\boldsymbol{\pi}_\star = \pi, \mathbf{A} = A, \mathbf{B} = B)$ takes the form

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{\pi}_\star = \pi, \mathbf{A} = A, \mathbf{B} = B) = \\
&= \frac{1}{n!} \prod_{i<j} \left[\left(\frac{\lambda s}{n}\right)^{A_{ij} B_{\pi(i)\pi(j)}} \left(\frac{\lambda}{n}(1-s)\right)^{A_{ij}(1-B_{\pi(i)\pi(j)}) + (1-A_{ij}) B_{\pi(i)\pi(j)}} \left(1 - \frac{\lambda}{n}(2-s)\right)^{(1-A_{ij})(1-B_{\pi(i)\pi(j)})} \right] \\
&= \frac{1}{n!} \left(1 - \frac{\lambda}{n}(2-s)\right)^{\frac{n(n-1)}{2}} \left(\frac{\frac{\lambda}{n}(1-s)}{1 - \frac{\lambda}{n}(2-s)}\right)^{\sum_{i<j}(A_{ij}+B_{ij})} \left(\frac{ns}{\lambda(1-s)^2}\left(1 - \frac{\lambda}{n}(2-s)\right)\right)^{\sum_{i<j} A_{ij} B_{\pi(i)\pi(j)}} , \quad (7)
\end{aligned}$$

hence the posterior probability that is obtained, up to its normalization, by keeping only the terms that depend on π in the joint law, reads:

$$\mathbb{P}(\boldsymbol{\pi}_\star = \pi | \mathbf{A} = A, \mathbf{B} = B) \propto \left(\frac{ns}{\lambda(1-s)^2}\left(1 - \frac{\lambda}{n}(2-s)\right)\right)^{\sum_{i<j} A_{ij} B_{\pi(i)\pi(j)}} . \quad (8)$$

Maximising Eq. (8) corresponds thus to solving a quadratic assignment problem, that is notoriously a NP-hard problem (see [10] for a review), and the computation of marginal probabilities of (8) is at least as difficult.

As a consequence we will turn now to approximations of the posterior probability. One possibility, that was investigated in [24, 25], is to write (8) as a factor graph and to derive the Belief Propagation (BP) algorithm associated to it. This is, however, a rather problematic strategy: even if the graphs $\mathbf{A}$ and $\mathbf{B}$ are locally tree-like, the constraint that π is a permutation has to be implemented by factor nodes with a dense structure and a proliferation of short loops that are quite detrimental for the quality of the BP approximation, unless some side information is provided with the ground-truth values of $\boldsymbol{\pi}_\star(i)$ revealed for a fraction of vertices i. We will follow therefore a different path, exploiting the local tree-like structure of the graphs.

The idea of our computation is to discard a part of the available information and to compute the probability of the event $\boldsymbol{\pi}_\star(i) = i'$ not under the full posterior given the observation of $\mathbf{A}$ and $\mathbf{B}$, but under a truncated posterior where one only observes the local neighborhoods of i and i'. To be more precise, let us recall the notation for the truncated matrix $\mathbf{A}_{i,d}$ (resp. $\mathbf{B}_{i',d}$) corresponding to the adjacency matrix of the subgraph of $\mathbf{A}$ (resp. of $\mathbf{B}$) induced by the vertices at distance at most d from i (resp. i'). The truncated posterior probability of $\boldsymbol{\pi}_\star(i) = i'$ can be rewritten with Bayes formula as

$$\begin{aligned}
\mathbb{P}(\boldsymbol{\pi}_\star(i) = i' | \mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T') &= \frac{\mathbb{P}(\boldsymbol{\pi}_\star(i) = i', \mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T')}{\mathbb{P}(\mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T')} \\
&= \frac{\mathbb{P}(\mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T' | \boldsymbol{\pi}_\star(i) = i') \mathbb{P}(\boldsymbol{\pi}_\star(i) = i')}{\mathbb{P}(\mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T')}
\end{aligned}$$

$$= \frac{1}{n} \frac{\mathbb{P}(\mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T' | \boldsymbol{\pi}_\star(i) = i')}{\mathbb{P}(\mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T')} \ , \tag{9}$$

where in the last line we used the fact the prior probability of the event $\boldsymbol{\pi}_\star(i) = i'$ is $1/n$. We claim that in the large n limit with d fixed the observed neighborhoods T and T' are trees with high probability, and that the fraction in (9) converges to the ratio of the probabilities introduced in the local properties of the random graphs in Section 2.2, namely

$$\lim_{n\to\infty} \frac{\mathbb{P}(\mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T' | \boldsymbol{\pi}_\star(i) = i')}{\mathbb{P}(\mathbf{A}_{i,d} = T, \mathbf{B}_{i',d} = T')} = \frac{\mathbb{P}_1^{(d)}[T, T']}{\mathbb{P}_0^{(d)}[T]\mathbb{P}_0^{(d)}[T']} \ , \tag{10}$$

with $\mathbb{P}_0^{(d)}$ and $\mathbb{P}_1^{(d)}$ defined in equations (1) and (2) respectively. As a matter of fact the numerator of the left hand side is precisely the quantity we investigated in Section 2.2.3, and in the denominator the permutation $\boldsymbol{\pi}_\star$ is averaged out, hence the vertex i' can be seen as uniformly chosen in the aligned graph $\mathbf{C}$. With a probability $1 - O(1/n)$ the depth d neighborhoods of i and i' do not intersect in $\mathbf{G}$, hence T and T' are asymptotically independent and drawn from their marginal probabilities. From the expressions given in equations (1) and (2) we can derive a recursive expression of this ratio, that we will denote $L^{(d)}(T, T')$. After some simplifications of the Poisson probabilities one obtains indeed

$$L^{(d)}(T, T') = \frac{\mathbb{P}_1^{(d)}[T, T']}{\mathbb{P}_0^{(d)}[T]\mathbb{P}_0^{(d)}[T']} = \sum_{l_{\rm bi}=0}^{\min(l,l')} \mathrm{e}^{\lambda s}(1-s)^{l+l'} \left(\frac{s}{\lambda(1-s)^2}\right)^{l_{\rm bi}} \sum_{I,I',\sigma} \prod_{i\in I} L^{(d-1)}(T_i, T'_{\sigma(i)}) \ , \tag{11}$$

with $L^{(0)}(T, T') = 1$, and where we recall that as in equation (2) $T_1, \dots, T_l$ are the subtrees of T rooted at the l offsprings of the root of T, $T'_1, \dots, T'_{l'}$ is a similar decomposition of T', I (resp. I') is a subset of $[l]$ (resp. of $[l']$) of $l_{\rm bi}$ elements, and σ a bijection from I to I'.

Note that $L^{(d)}$ is the ratio of the probabilities for the generation of the pair (T, T') in two different ensembles, a correlated one with $\mathbb{P}_1^{(d)}$ and an uncorrelated one with the product of the $\mathbb{P}_0^{(d)}$. According to the Neyman-Pearson lemma this likelihood ratio can be used to design the optimal estimators for the hypothesis testing problem where an observer is handed a sample (T, T') and must decide whether the pair was generated in a correlated way from $\mathbb{P}_1^{(d)}$ (alternative hypothesis) or as a pair of independent copies from the marginal law $\mathbb{P}_0^{(d)}$ (null hypothesis). The optimal answer to this question is the alternative if and only if $L^{(d)}(T, T') \geq \alpha$, where α is a threshold that depends on the compromise to be made between false positive and false negative errors. In [18] this hypothesis testing problem has been studied per se, and translated to the graph alignment problem, here we arrived at it through a slightly different road, namely the study of the truncated posterior.

To simplify notations in the following we abstract Eq. (11) by introducing a function f that takes as inputs two integers l and l' and an $l \times l'$ array $L_{i,i'}$ of real numbers and computes

$$f(l, l'; \{L_{i,i'}\}) = \sum_{l_{\rm bi}=0}^{\min(l,l')} \mathrm{e}^{\lambda s}(1-s)^{l+l'} \left(\frac{s}{\lambda(1-s)^2}\right)^{l_{\rm bi}} \sum_{I,I',\sigma} \prod_{i\in I} L_{i,\sigma(i)} \ , \tag{12}$$

with the same definitions for I, I' and σ as in (11), and with the convention $f = \mathrm{e}^{\lambda s}(1-s)^{l+l'}$ if $\min(l, l') = 0$. In the limit of perfect correlation $s \to 1$ this function becomes

$$f(l, l'; \{L_{i,i'}\}) = \mathbb{I}(l = l')\mathrm{e}^{\lambda}\lambda^{-l} \sum_{\sigma} \prod_{i=1}^{l} L_{i,\sigma(i)} \ . \tag{13}$$

3.3. A message passing algorithm for the graph alignment problem

We shall now present the algorithm for the alignment of correlated Erdős-Rényi random graphs that follows naturally from the above considerations. Given the observations of two graphs A and B and a positive integer parameter d we compute an estimator $\widehat{\pi} = \widehat{\pi}(A, B)$ which is a function (not necessarily bijective) of the vertex set of A to the one of B, with the goal to maximize the average overlap of this correspondence with the groundtruth. As the information theoretical optimal procedure given in (6) is not computationally tractable we approximate the posterior distribution by its truncated version defined in (9), which corresponds to define a $n \times n$ score matrix $L^{(d)}_{ii'} = L^{(d)}(A_{i,d}, B_{i',d})$ and set $\widehat{\pi}(i) = \arg\max_{i'} L^{(d)}_{ii'}$, the constant multiplicative prefactor $1/n$ in (9) being irrelevant here. Thanks to the recursive nature of $L^{(d)}$ these scores can be computed via a message-passing procedure; to specify it we need first to introduce some additional notations. For a vertex i of A we write ∂i for the set of vertices adjacent to i, and $d_i = |\partial i|$. Similar notations apply to vertices i' of B; to avoid any confusion we will always use primed indices for vertices in B, and consider the vertex set $\{1, \dots, n\}$ of A as distinct from $\{1', \dots, n'\}$, the vertex set of B. We now introduce a set of messages $L^{(t)}_{ii' \to jj'}$ for all vertices i of A, all vertices i' of B, all $j \in \partial i$, and all $j' \in \partial i'$. The discrete 'time' index t corresponds to a number of iterations, or depth; the interpretation of $L^{(t)}_{ii' \to jj'}$ is the likelihood ratio for the depth t neighborhoods of i and i', deprived of the branch linking them to j and j' respectively. We can thus summarize our algorithm as follows:

(i) all messages are initialized to $L^{(0)}_{ii' \to jj'} = 1$.

(ii) for each $t = 1, \dots, d-1$, they are updated according to

$$L^{(t)}_{ii' \to jj'} = f(d_i - 1, d_{i'} - 1; \{L^{(t-1)}_{kk' \to ii'} : k \in \partial i \setminus j \, , \, k' \in \partial i' \setminus j'\}) \, , \tag{14}$$

where the function f was defined in Eq. (12).

(iii) the scores are computed as

$$L^{(d)}_{ii'} = f(d_i, d_{i'}; \{L^{(d-1)}_{jj' \to ii'} : j \in \partial i \, , \, j' \in \partial i'\}) \, . \tag{15}$$

(iv) finally the estimator $\widehat{\pi}$ is computed as $\widehat{\pi}(i) = \arg\max_{i'} L^{(d)}_{ii'}$, with ties broken uniformly at random if several i' achieve the same maximal score.

Let us make a series of comments before presenting the numerical results we obtained with this algorithm:

- Cycles certainly occur in random graphs, that are only locally tree-like; one may thus wonder about the meaning of $L^{(d)}_{ii'}$ when the depth d neighborhoods of i and i' are not both trees. A moment of thought reveals that the iterative procedure described in the algorithm actually computes the likelihood ratio of the trees of non-backtracking walks of length at most d starting at i and i' (also known as the computational tree). This coincides with the usual neighborhood when the latter is a tree, and otherwise "unwraps" the cycles according to this non-backtracking rule. Note also that there is only a finite number (on average) of cycles of finite length in random graphs with fixed average degrees. The neighborhood of depth d of most vertices is acyclic even if d grows (logarithmically) with n, see for instance [15] for a precise statement of the coupling between random graphs and random trees on logarithmic scales.
- The interpretation put forward in [18] of the scores as likelihood ratios for an hypothesis testing problem between two perfectly aligned versus two independent trees may suggest an

issue when using these scores $L_{ii'}^{(d)}$ between partially correlated trees, i.e. when $i' \neq \boldsymbol{\pi}_\star(i)$ but with i and $\boldsymbol{\pi}_\star{}^{-1}(i')$ at a graph distance smaller than $2d$ in $\mathbf{G}$, since in that case the joint law of the neighborhoods is neither the null nor the alternative of the hypothesis testing problem. We emphasize however that our derivation of the algorithm follows from the truncation of the posterior probability of the hidden permutation, and does not rely on the hypothesis testing interpretation. Moreover we have checked that this potential issue does not spoil the behavior of the algorithm, the scores between partially aligned neighborhoods being substantially smaller than between perfectly aligned ones, see Appendix B.1 for more details on this point.

- The number of messages $L_{ii'\to jj'}$ is 4 times the product of the number of edges in the two graphs, i.e., $O(n^2)$ in the sparse regime considered here. The algorithm requires therefore a total number of message updates of order $O(d\,n^2)$. However, the number of operations required for the computation of the function f in (12) grows very fast (factorially) with $\min\{l, l'\}$, because of the sum over the permutations σ. As the maximal degree in an ER graph grows slowly with n, as $O\left(\frac{\ln n}{\ln\ln n}\right)$, the asymptotic scaling with n of the computational cost of the algorithm remains polynomial in n. Nevertheless from a practical point of view this factorial growth shall restrict our study to random graphs with rather small average degree (for instance if $l = l' = 10$ the computation of f would involve a sum over more than 10^8 terms).
- Another implementation remark concerns the very large value some messages can acquire for growing d; to alleviate this problem we actually stored the logarithms of the messages.
- An implementation of the algorithm in C language can be found at https://github.com/giovannipiccioli/graph_alignment.

4. Numerical results for the graph alignment algorithm

This Section shall be devoted to a presentation of the results we have obtained by numerical simulations of the algorithm presented above. Let us recall that the parameters of the problem are n, the number of vertices of the pair of graphs to be aligned, λ, their average degree, s, their correlation, and d, the parameter of the algorithm that controls the depth at which the neighborhoods of the vertices are inspected to decide which pairs of nodes to match. For each choice of these parameters we compute $\mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)]$, where the overlap between the groundtruth permutation $\boldsymbol{\pi}_\star$ and the estimate returned by the algorithm $\widehat{\boldsymbol{\pi}}$ has been defined in Eq. (5), and the expectation is over the randomness in the generation of the pair $(\mathbf{A}, \mathbf{B})$ of graphs, and possibly over the tie-breaking procedure of the algorithm; to simplify the notation we keep implicit the dependency of this average overlap upon the parameters (n, λ, s, d). We estimate numerically this expectation by an empirical average over several independent samples. Our ultimate goal would be to determine the algorithmic phase diagram in the (λ, s) plane in the large size limit $n \to \infty$, in other words to determine the values of these parameters for which the algorithm achieves asymptotically a partial recovery of the hidden permutation; in formula this corresponds to $\liminf_{n\to\infty} \mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)] > 0$, for a suitable choice of the depth parameter d, possibly n-dependent. We expect this property to be monotonous in s, as increasing the correlation between the two graphs increase the amount of information available for the inference of $\boldsymbol{\pi}_\star$, we would like thus to determine the algorithmic threshold $s_{\mathrm{algo}}(\lambda)$ defined as the smallest value of s for which the algorithm achieves asymptotically partial recovery for the parameters (λ, s). One knows from previous works that partial recovery is only achievable in some regions of the (λ, s) plane; in particular it has been proven in [17] that the asymptotic average overlap of any estimator (be it efficiently computable or not) is upper bounded by $c(\lambda s)$, the largest non-negative solution

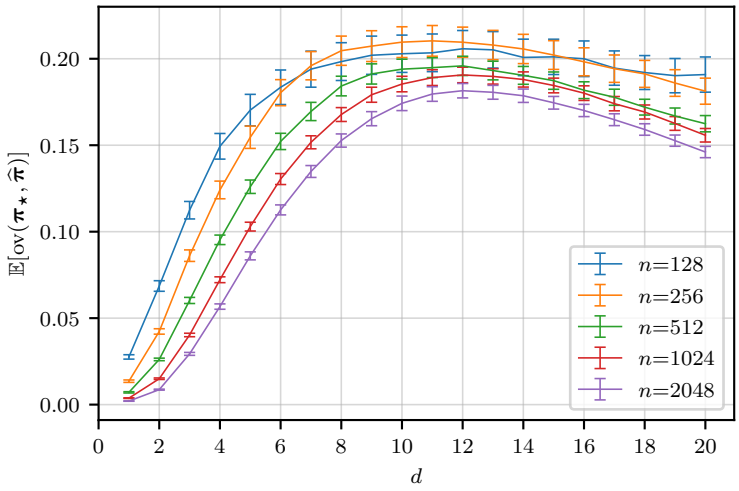

Figure 2. The average overlap as a function of the depth d for average degree $\lambda = 1.4$, correlation $s = 0.81$, and various values of system size n. Each point is averaged over 100 realizations of the two graphs $\mathbf{A}$, $\mathbf{B}$. The overlap between the estimator $\widehat{\boldsymbol{\pi}}$ and the ground truth $\boldsymbol{\pi}_\star$ exhibits a clear maximum for an intermediate value of the depth d, which increases slowly with the system size n.

of $1 - c = \mathrm{e}^{-\lambda s c}$, which corresponds to the fraction of vertices in the largest component of the intersection graph between $\mathbf{A}$ and $\mathbf{C}$, an Erdős-Rényi random graph of average degree λs. As $c(\lambda s) = 0$ whenever $\lambda s \leq 1$, this implies that partial recovery is impossible in this case, hence the lower bound on the algorithmic threshold $s_{\mathrm{algo}}(\lambda) > 1/\lambda$. As we shall see the determination of the algorithmic phase transition is a very challenging numerical task, we have nevertheless some partial answers to this question.

4.1. The scaling of the depth parameter d

Let us first discuss the choice of the depth parameter d of the algorithm, that we have left unspecified up to now. On the one hand we would like to take d as large as possible: the algorithm has been derived by replacing the full posterior distribution (8), that exploits all the information contained in the realization of $(\mathbf{A}, \mathbf{B})$, by its truncated version (9) which only depends on $(\mathbf{A}_{\cdot,d}, \mathbf{B}_{\cdot,d})$, thus discarding a part of the available information. Larger values of d corresponds to a less drastic loss of information, the truncated posterior getting closer to the information-theoretical optimal full posterior. On the other hand, for a finite value of n we should not take d arbitrarily large: the computation that underlies the algorithm is based on the assumption that the neighborhoods explored up to depth d are trees, which is not true when d exceeds $O(\ln n)$, the scaling of the minimal length of a cycle from a typical vertex in a sparse random graph. When a cycle is encountered the algorithm unwraps it according to the non-backtracking rule, hence producing spurious terms that corrupt the estimator. As a consequence one expects that for a given choice of the parameters (n, λ, s) there will be an optimal value of d that reaches a compromise between these two conflicting requirements, for which the average overlap will be maximal. This is confirmed by the results presented in Fig. 2, which shows the average overlap as a function of d, for one choice of (λ, s) and several values of n: these curves exhibit indeed a maximum at an optimal value $d_*(n, \lambda, s)$.

In order to make statements about the asymptotic behavior of the algorithm in the limit $n \to \infty$ one should now understand what is the scaling of the optimal depth $d_*(n, \lambda, s)$ in this limit. For the range of values of n displayed in Fig. 2 one sees a mild increase of d_* with n, one could thus be tempted to assume that it reaches a finite value when n diverges. A moment of thought reveals that this expectation is wrong, and, as a matter of fact, for any value of (λ, s) the average overlap is at most of order $1/n$, and hence vanishes, if the large n limit is taken with any fixed finite value d. To justify this claim let us consider a vertex i in $\mathbf{A}$, its image $i' = \boldsymbol{\pi}_\star(i)$ in $\mathbf{B}$ through the ground-truth permutation, and let us denote T_i and $T'_{i'}$ their respective neighborhoods of depth d in the two graphs. There are two failure mechanisms in the algorithm that lead to a wrong estimation of the vertex matched to i, i.e. to $\widehat{\pi}(i) \neq i'$: either a vertex $j' \neq i'$ achieves a strictly higher score than the correct assignment, $L^{(d)}_{i,j'} > L^{(d)}_{i,i'}$, or $L^{(d)}_{i,i'}$ achieves the maximum score in $L^{(d)}_{i,\cdot}$ but i' is not the unique maximizer. In the latter case the probability that $\widehat{\pi}(i) = i'$ is one divided by the degeneracy of the maximum, as the algorithm picks a uniformly random maximizer of the score in case of ties. It turns out that if $n \to \infty$ with d fixed the number of vertices j' that achieves the same score as the correct match i' is extensive (proportional to n): $T'_{i'}$ being a fixed finite tree, the probability $\mathbb{P}_0^{(d)}[T'_{i'}]$ that the depth d neighborhood of a randomly chosen vertex j' is isomorphic to it is of order 1, hence the cardinality of $\{j' \,:\, L^{(d)}_{i,j'} = L^{(d)}_{i,i'}\}$ is of order n. This concludes the justification of our claim that the average overlap is at most of order $1/n$ in the large n limit with d fixed: even if i' achieves the maximum in $L^{(d)}_{i,\cdot}$ there will be an extensive number of vertices achieving it, hence i' will be picked with a probability inversely proportional to this degeneracy.

This reasoning is confirmed by the numerical results presented in Fig. 3, where we plot the average overlap as a function of n for a fixed (and small) value of the depth, $d = 2$. The left panel, for some choices of (λ, s), shows indeed a behavior of the form $\mathbb{E}[\mathrm{ov}(\boldsymbol{\pi}_\star, \widehat{\boldsymbol{\pi}})] \propto n^\alpha$ with $\alpha = -1$. The right panel, for slightly different parameters (λ, s), exhibit a power-law behavior with $\alpha > -1$; we interpret this apparent contradiction with the reasoning above as signalling that the values of n investigated (which are the largest ones we could reach within a reasonable amount of computation time) are too small to be representative of the asymptotic behavior of the limit $n \to \infty$. Indeed the probability $\mathbb{P}_0^{(d)}[T'_{i'}]$ is certainly of order 1, but can be numerically very small; as long as $n\mathbb{P}_0^{(d)}[T'_{i'}] \ll 1$ the typical number of confounding vertices is on average much smaller than 1, and hence typically 0.

Having ruled out the possibility that the optimal depth $d_*(n, \lambda, s)$ remains constant as $n \to \infty$ leaves open the question of its scaling with n in this limit. We conjecture that $d_*(n, \lambda, s) = \widetilde{d}_*(\lambda, s) \ln n + o(\ln n)$, with a constant $\widetilde{d}_*$ possibly dependent on (λ, s). Indeed $d = \Theta(\ln n)$ is the largest possible scaling of d that ensures that most of the depth-d neighborhoods of the graph are trees, compromising between the two opposed requirements on d discussed above. Moreover with this scaling of d the typical degeneracy $n\mathbb{P}_0^{(d)}[T'_{i'}]$ of the scores remains bounded, invalidating the previous "pigeonhole" argument used in the limit $n \to \infty$ with d fixed, and opens the possibility of asymptotic partial recovery in some regions of the (λ, s) plane. Unfortunately it does not seem possible to test numerically this conjecture: on the range of n that is accessible to our simulations the function $\ln n$ has very small variations that do not allow for an accurate fit.

Another element of information on the values of depth d for which the truncated posterior behaves as expected in Bayes-optimal inference is presented in Appendix B.3 where we test the validity of the so-called Nishimori condition for the truncated estimator. We observe that the Nishimori condition is violated for larger values of the depth d. This implies that the presented message passing algorithm will be suboptimal for those values of d.

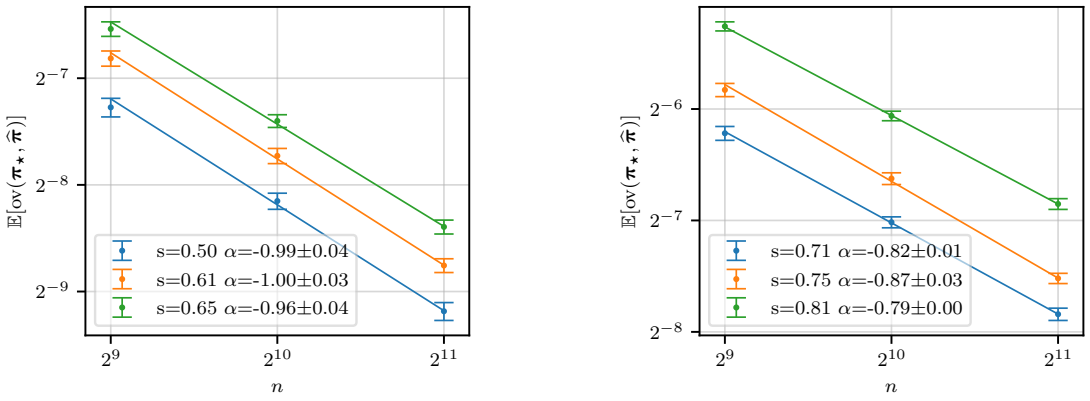

Figure 3. The average overlap between the estimated and ground truth configurations $\mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)]$ as a function of the system size n for a fixed depth $d = 2$, average degree $\lambda = 1.4$ and for the left panel correlations $s = 0.5$, $s = 0.61$, $s = 0.65$, for the right panel $s = 0.71$, $s = 0.75$, $s = 0.81$. Every point is an average over 100 samples. The lines correspond to power-laws fits of the form $\mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)] = c\, n^\alpha$ with c and α two fitting constants, logarithmic scales being used on both axis. In the left panel the exponent α is compatible with the value -1 of the analytic argument, while on the right $\alpha > -1$, which reveals strong finite size effects and a preasymptotic behavior.

4.2. The performances of the algorithm across the (λ, s) plane

One can adopt two attitudes in facing this difficulty in the choice of d: if one is interested in the performances reachable in practice by the algorithm, then one has only to consider relatively small values of n, at most a few thousands, for which d_* varies only mildly, and study the overlap for a fixed, reasonably large value of d. This is what we have done to produce the curves of Fig. 4, choosing here $d = 10$. As anticipated above these curves are increasing functions of s, confirming that more correlated graphs are easier to align. For some values of the parameters we obtain average overlaps which are higher than the upper bound derived in [17] in terms of the fraction of vertices in the giant component of the bicolored subgraph of $\mathbf{G}$, an Erdős-Rényi random graph of average degree λs. This of course is not a contradiction, the bound is valid asymptotically in the $n \to \infty$ limit while our numerical results are obtained at finite n, for which the size of the largest components of the bicolored subgraph of $\mathbf{G}$ have strong fluctuations (it would thus be useful to derive an information-theoretic upper bound on the average overlap valid for all n, or at least compute the finite size corrections to the asymptotic one). It shows nevertheless that the sizes of a few thousands that one can reach numerically suffer from strong finite size effects and are still far from the asymptotic behavior; as a matter of fact we argued above that in the limit $n \to \infty$ taken with d fixed (which is the case on Fig. 4) the average overlap vanishes, a fact which is certainly impossible to deduce from a naive extrapolation of the data in Fig. 4.

If on the other hand one is interested in making conjectures on the $n \to \infty$ limit behavior of the algorithm from the finite n results, one needs to find a meaningful way to extrapolate them, taking into account the necessary dependency of d on n. To avoid the problematic choice of the prefactor in front of $\ln n$ in d we adopted the following pragmatic strategy: for every value of (n, λ, s) investigated we computed the average overlap for all values of d with $1 \leq d \leq 20$, and selected the one that maximized the average overlap, defined above as $d_*(n, \lambda, s)$. This eliminates one parameter and in principle allows for an extrapolation at large n for fixed (λ, s); the results presented in figures 5 and 6 have been obtained by following this procedure.

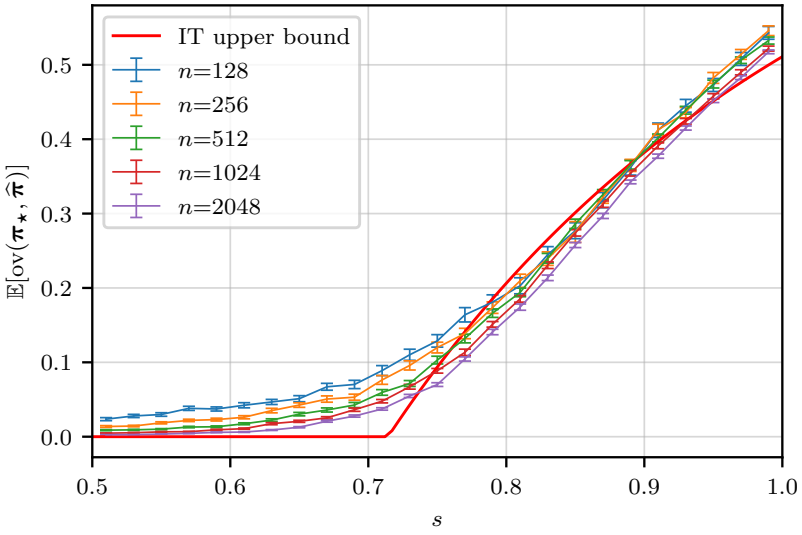

Figure 4. The average overlap $\mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)]$ as a function of the correlation s for average degree $\lambda = 1.4$, depth $d = 10$ and several values of system sizes n. The curve denoted 'IT upper bound' refers to the upper bound of [17] on the overlap given by the fraction of vertices in the giant component of an Erdős-Rényi random graph of average degree λs.

The top left panel of figure 5, corresponding to $\lambda = 1.1$, shows average overlaps exceeding considerably the upper bound of [17], even in the regime $\lambda s < 1$ where the latter vanishes (the top right panel for $\lambda = 1.4$ displays a similar but less marked phenomenon): effective finite-size partial recovery is achievable even when the asymptotic one is impossible. As explained above this is a finite size effect, with the gap between the numerical results and the $n \to \infty$ bound closing at a very slow rate of order $1/\ln(n)$ (doubling n produces a constant downward shift in the overlap curves for the ranges investigated). On the contrary the average overlaps in the bottom panels of figure 5, corresponding to $\lambda = 1.9$ and $\lambda = 2.9$, are well below the upper bound, and suggest that the algorithmic threshold $s_{\mathrm{algo}}(\lambda)$ is markedly above the lower bound $1/\lambda$. In principle the determination of $s_{\mathrm{algo}}(\lambda)$ should simply follow from an extrapolation of these curves in the $n \to \infty$ limit, the extrapolation being 0 for $s < s_{\mathrm{algo}}(\lambda)$ and strictly positive for $s > s_{\mathrm{algo}}(\lambda)$; unfortunately, because of the relatively small sizes we could reach, of the slowly vanishing finite-size effects and in absence of additional analytic arguments to constrain the fitting form we did not manage to find a reliable and stable extrapolation procedure. Nevertheless it is tempting from a visual inspection of the bottom panel of Fig. 5 to propose an estimation of $s_{\mathrm{algo}}(\lambda)$ as the value of s for which the average overlap markedly grows away from 0, for the largest available size (here $n = 2048$). For both these two values of λ this rough estimate is slightly above $s \approx 0.6$; this very weak dependency on λ is confirmed by the data presented in the left panel of Fig. 6, which shows on the same plot various values of λ for $n = 2048$. All the curves collapse to zero for $s < 0.6$, indicating that the algorithm does not achieve partial recovery in this regime. Even if one does not see a sharp transition on these finite-size results all the curves seem to start to grow above roughly the same value $s \approx 0.6$. To confirm this finding we performed some numerical experiments for larger values of λ, up to $\lambda = 6.8$, for which we were limited to much smaller sizes ($n = 64$) because of the factorial growth of the computational cost with the degree of the vertices; these results, shown in the right panel of Fig. 6, exhibit also

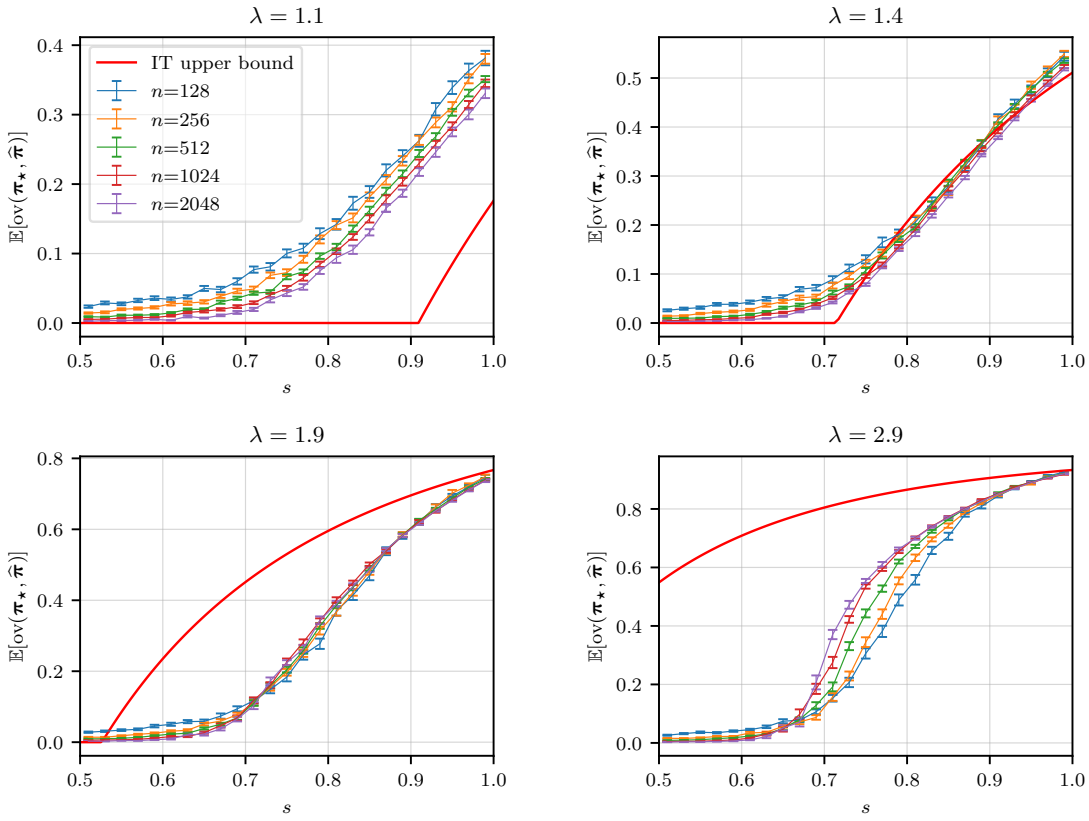

Figure 5. The average overlap $\mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)]$ computed at the optimal depth, as a function of the correlation s, with average degree $\lambda = 1.1$ (top left panel), $\lambda = 1.4$ (top right panel), $\lambda = 1.9$ (bottom left panel), $\lambda = 2.9$ (bottom right panel), for several values of system size n, the color code being the same in all panels.

a notable growth of the average overlap around $s \approx 0.6$. To be slightly more quantitative we introduce a small arbitrary threshold R and define an effective transition point (or more precisely a crossover) as the minimal value of s for which the overlap is larger than R, which thus depends on R, n and λ. These effective transition lines in the (λ, s) phase diagram are presented in Fig. 7 for two values of n (2048 in the left panel and 64 in the right one) and several values of R. Once again the finite size effects lead to partial recovery in the information theoretic impossible regime. Moreover for $n = 64$ and large values of λ one observes that the crossover line enters the information theoretic feasible phase, thus hinting at the existence of a hard phase, i.e. a phase where the problem is information theoretically feasible but the algorithm fails to perform partial recovery.

The conclusion of this Section is that the numerical experiments show for practical sizes of the order $n \approx 10^3$ and $0.9 \leq \lambda \leq 2.9$ an observed algorithmic threshold for partial recovery almost independent on λ, around $s \approx 0.6$.

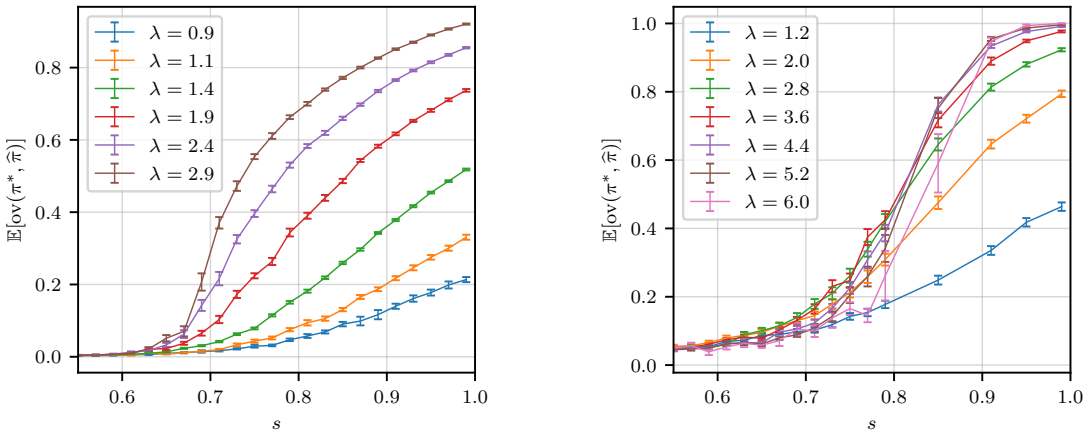

Figure 6. The average overlap $\mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)]$ computed at the optimal depth as a function of the correlation s for several values of the average degree λ. (Left) Results obtained for system size $n = 2048$: all the curves are zero below $s \approx 0.6$. For $s > 0.6$, the overlap increases faster in curves with greater values of λ. (Right) Results obtained for $n = 64$: we consider higher values of λ, up to $\lambda = 6$. When s tends to 1 curves with higher λ converge to higher values of the overlap.

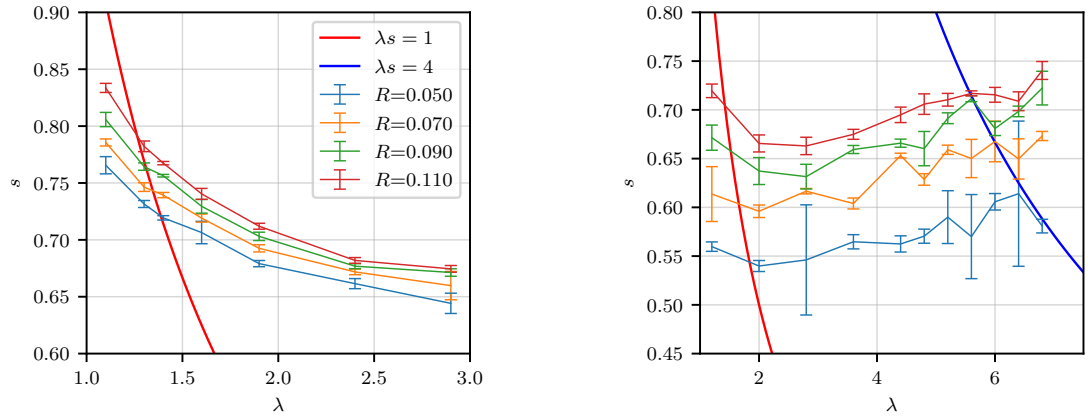

Figure 7. The crossover lines in the (λ, s) plane, defined as the minimal value of the correlation s for which the overlap exceeds a small threshold R. (Left) $n = 2048$. (Right) $n = 64$: the crossover position is approximately constant for large values of λ. We compare these lines with the lower and upper bounds on the information theoretic threshold for the possibility of asymptotic partial recovery, namely $\lambda s = 1$ and $\lambda s = 4$.

5. The tree problem

We have seen in the previous Section that the determination of the algorithmic threshold $s_{\rm algo}(\lambda)$ from the finite n numerical simulations on pairs of graphs suffered from strong finite n corrections that limited its accuracy. In this Section we shall follow a different road, that amounts in some sense to work directly with $n = \infty$; nevertheless we will have to face some other numerical difficulties, related to finite d effects.

The computation of the probability of the event $\boldsymbol{\pi}_\star(i) = i'$ under the truncated posterior made naturally appear the ratio $L^{(d)}(T,T') = \frac{\mathbb{P}_1^{(d)}[T,T']}{\mathbb{P}_0^{(d)}[T]\mathbb{P}_0^{(d)}[T']}$ of probabilities for the generation of a pair of trees in a correlated and uncorrelated ensemble of random trees. According to the Neyman-Pearson lemma this likelihood ratio leads, by thresholding, to the family of optimal estimators in the hypothesis testing problem where an observer has to decide from which of these two laws a pair of trees has been generated. Forgetting temporarily the original graph alignment problem we concentrate for the moment on this hypothesis testing problem on trees (note that the parameter n does not appear in the latter). This problem has been studied per se in [18], which emphasized the importance in this case of the notion of one-sided tests, namely families of d-dependent estimators that asymptotically for large d have a vanishing probability of error under the null hypothesis, and a probability of error bounded away from 1 under the alternative. Conditions for the existence of such tests have been derived in terms of the Kullback-Leibler divergence between the two distributions,

$$\mathrm{KL}_d = D(\mathbb{P}_1^{(d)} || \mathbb{P}_0^{(d)} \otimes \mathbb{P}_0^{(d)}) = \sum_{T,T'} \mathbb{P}_1^{(d)}[T,T'] \ln \left(\frac{\mathbb{P}_1^{(d)}[T,T']}{\mathbb{P}_0^{(d)}[T]\mathbb{P}_0^{(d)}[T']} \right) = \sum_{T,T'} \mathbb{P}_1^{(d)}[T,T'] \ln L^{(d)}(T,T') \,. \tag{16}$$

It was indeed shown in [18] that KL_d is a non-decreasing sequence in d, that diverges to $+\infty$ when $d \to +\infty$ if and only if one-sided tests do exist for the corresponding values of (λ, s). This leads to the definition of a threshold $s_{\rm c}(\lambda)$ for the tree problem, such that KL_d diverges with d if and only if $s > s_{\rm c}(\lambda)$ (we assume again that this property is monotonous in s). Some upper and lower bounds on $s_{\rm c}(\lambda)$ have also been proven in [18]. The lower bound $s_{\rm c}(\lambda) \geq s_{\rm lb}(\lambda) = 1/\lambda$, which parallels the one on $s_{\rm algo}$ discussed before, is valid for all $\lambda \geq 1$; for $\lambda \gtrsim 5.5$ the theorem 5 in [18] provides an improved lower bound that behaves asymptotically as $1/\sqrt{\lambda}$ for $\lambda \to \infty$. Upper bounds on $s_{\rm c}(\lambda)$ are also provided by theorems 3 and 4 in [18], by showing that KL_d diverges with d for some values of (λ, s); explicit formulas for these bounds are cumbersome to write, see Fig. 9 for a graphical representation of one of them. For $\lambda \in [1, 1.178]$ the upper and lower bounds coincide, implying that $s_{\rm c}(\lambda) = 1/\lambda$ in this interval.

We have performed a numerical study of this hypothesis testing problem on trees, through the computation of the Kullback-Leibler divergence KL_d defined in Eq. (16). We estimated this quantity with an elementary, brute-force procedure, by generating a large number of pairs of trees (T,T') with the law $\mathbb{P}_1^{(d)}$, computing $L^{(d)}(T,T')$ using the recursions (11) from the leaves towards the root, and performing the empirical average over the samples. The generation step was done by first drawing the multi-type Galton-Watson tree $\mathcal{T}$ described in Section 2.2.2 and then projecting it to the blue and red connected components of the root, $(T,T') = (b(\mathcal{T}), r(\mathcal{T}))$. The numerical results obtained in this way are presented in Fig. 8 for $\lambda = 1.5$. One obtains as expected that KL_d is a non-decreasing function of d, with a behavior at large d that suggests a saturation for small s and a divergence for large s. As the range of d that can be investigated is rather small (the limiting factor is the storage of the pair of trees, the memory cost is $O(\lambda^{2d})$ because of the number of vertices that grows exponentially with d, with a rate that increases with λ) one cannot locate affirmatively a transition in a very precise way. Nevertheless it is tempting

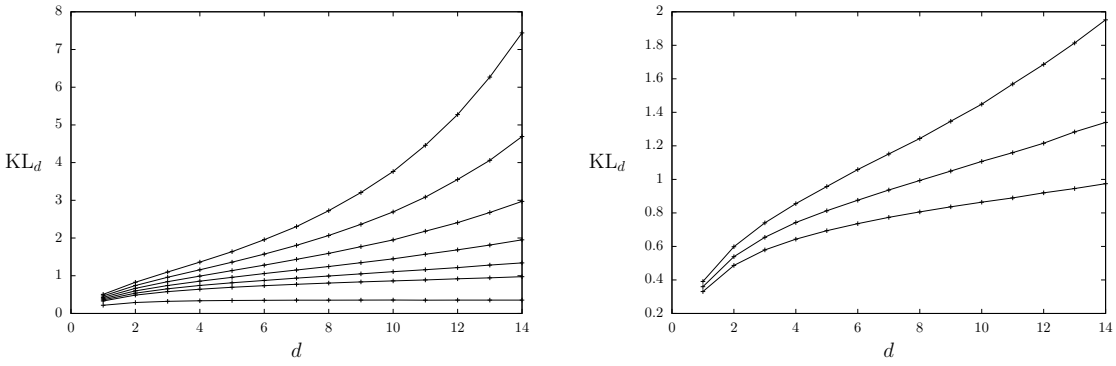

Figure 8. Left panel: numerical estimates of the Kullback-Leibler divergence KL_d defined in Eq. (16) for the hypothesis testing problem on trees, as a function of d for $\lambda = 1.5$, and from bottom to top $s = 0.6$, $s = 0.7$, $s = 0.72$, $s = 0.74$, $s = 0.76$, $s = 0.78$, and $s = 0.8$. Averages are taken over 10^5 to 10^6 samples depending on the values of d, error bars are of the order of the symbol size. The right panel shows the curves corresponding to $s = 0.7$, $s = 0.72$ and $s = 0.74$ on a different range to better appreciate the transition region.

to conjecture that $s_{\mathrm{c}}(\lambda = 1.5) \in [0.7, 0.74]$, as the curves bend upwards (resp. downwards) for larger (resp. smaller) values of s (see in particular the right panel of Fig. 8), and the divergence is argued in [18] to be exponential in d above the transition (we mention as a side remark the open problem of the continuity of this transition, i.e. whether $\lim_{s \to s_{\mathrm{c}}^-} \lim_{d \to \infty} \mathrm{KL}_d$ is finite or not). This procedure to determine s_{c} is somehow subjective and cannot yield very accurate estimates of s_{c}, but we did not found a better way to perform the large d extrapolation. We repeated the same analysis for a few different values of λ; for larger values of λ we could only reach $d = 10$, whereas for $\lambda = 1.2$ we computed KL_d up to $d = 20$. The results are summarized as a phase diagram in the (λ, s) plane in Figure 9, along with the lower bound $\lambda s = 1$ and the upper bound from Theorem 3 in [18].

Let us now come back to the original graph alignment problem and its connection with the tree problem. We first underline the fact that the numerically determined values of $s_{\mathrm{c}}(\lambda)$ plotted in Fig. 9 are compatible with the estimate $s_{\mathrm{algo}}(\lambda) \approx 0.6$ obtained from the finite n simulations of the previous Section in this range of λ, and we believe that despite the problematic large d extrapolation the determination of $s_{\mathrm{c}}(\lambda)$ is more accurate than the one of $s_{\mathrm{algo}}(\lambda)$. Furthermore we expect that these two thresholds coincide, namely that $s_{\mathrm{c}}(\lambda) = s_{\mathrm{algo}}(\lambda)$. As a matter of fact it has been proven in [18] that if one-sided tests for the tree problem exist for a given value of (λ, s), then a polynomial-time algorithm achieves partial recovery for the corresponding graph problem; however, this is proven for a different algorithm than the one presented above, less efficient but easier to analyze. The intuitive connection between the graph and the tree problem goes as follows. For a given vertex i the algorithm recovers successfully its matched vertex $i' = \boldsymbol{\pi}_\star(i)$ with a positive probability if the score $L_{i,i'}^{(d)}$ is a maximizer of $L_{i,\cdot}^{(d)}$, and if the degeneracy of this maximum is finite. Forgetting the cycles in the graph and some correlations between overlapping neighborhoods one can picture this vector of n random variables as containing one sample of $L^{(d)}$ under the law $\mathbb{P}_1^{(d)}$, and $n-1$ samples drawn with the law $\mathbb{P}_0^{(d)}$. For the single random variable corresponding to the aligned pair to be larger than the extreme value of the $n-1$ other ones in the large n limit the laws of the likelihood ratio under the null and the alternative have to strongly differ one from the other, with L being typically much larger under $\mathbb{P}_1$ than under $\mathbb{P}_0$, as expressed by the divergence of KL_d.

The connection between graph and tree problems is of course a recurrent theme that

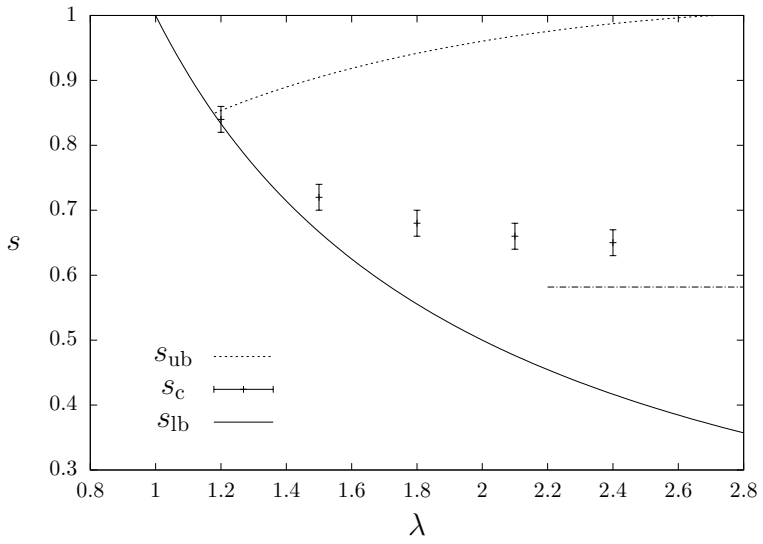

Figure 9. The numerically determined threshold $s_{\rm c}(\lambda)$ for the hypothesis testing problem on random trees, along with its lower bound $s_{\rm lb}(\lambda) = 1/\lambda$ and its upper bound $s_{\rm ub}(\lambda)$ that follows from theorem 3 in [18]. The meaning of the horizontal dot-dashed line is explained in the conclusions.

appeared in the previous literature under many guises. Without attempting to be exhaustive let us give a few examples that will be useful to discuss the situation of the present case. Under the name of the objective method it was shown in [26] that the minimal cost of the matching of a weighted complete graph converges in the large size limit to a quantity that can be computed from an infinite tree. The cavity method [27] allows to study statistical mechanics models defined on random graphs by exploiting their local convergence to trees. One way to interpret this method is to consider the factor graph associated to the interactions of the original graph model, and to study the latter via the so-called Belief Propagation (BP) algorithm, a message-passing procedure to compute approximations of the local marginals and of the global partition function (via the Bethe free-energy formula), that would be exact if the model was defined on a tree, and that is asymptotically exact for models on graphs locally converging to trees, provided some correlation decay conditions are fulfilled (for simplicity we only discuss here the Replica Symmetric version of the cavity method, see [27] for a discussion of Replica Symmetry Breaking when this decorrelation condition is violated). In these two examples the limiting object is characterized by a random variable $\boldsymbol{X}$ (the message passed between adjacent nodes), that obeys a fixed point equation of the form $\boldsymbol{X} \stackrel{\rm d}{=} g(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$, where the equality is in distribution, and the $\boldsymbol{X}_i$'s are i.i.d. copies of $\boldsymbol{X}$. This type of fixed point condition is known as a Recursive Distributional Equation (RDE), which can be easily solved numerically by a so-called population dynamics procedure. In the context of inference problems let us also mention the case of the Stochastic Block Model (SBM), where one has to recover a hidden signal made of labels placed on the vertices of a graph, the observations being the edges of the graph, whose probability of presence depends on the labels of the two vertices at its endpoints [28–30]. In the sparse regime of the SBM, with constant average degree, the posterior distribution of the labels given the observed graph can be written as a factor graph which converges locally to a tree (treating the information from the absent edges in an average way), and which can be studied with BP and the cavity method. This connects the possibility of efficient recovery of the labels on the graph model to

the possibility of (robust) reconstruction on the associated tree problem [31]. Moreover the limit for the information theoretical possibility of recovery (without consideration of computational efficiency) is deduced from the mutual information between the labels and the observed edges, which itself is expressed in terms of the solution of the RDE via the Bethe free-energy formula.

We would like to emphasize that the connection between the graph and the tree problem encountered in the present paper exhibits important differences with the well-known examples we have just recalled:

- The message passing algorithm has not been obtained through the BP approximation of the factor graph encoding the full posterior of the problem. We replaced instead the posterior by a truncated probability law, that depends on the pair of vertices i, i' for which we estimate the probability that $\boldsymbol{\pi}_\star(i) = i'$. Once this truncation has been performed the rest of the derivation is asymptotically exact when d is finite with $n \to \infty$. The consequences of this observation are on the one hand that one cannot invoke some correlation decay property to justify the result, and on the other hand that one cannot use the limiting tree problem to compute some thermodynamic quantities like the normalization of the posterior, and deduce from it the information theoretical limits of partial recovery.
- The limiting tree problem is actually a problem involving pairs of trees, not a single tree like for instance in the cavity treatment of the SBM problem. As a consequence the message-passing equations of Eq. (14) involve messages passed between two pairs of vertices, and not along the edges of a graph as in the usual BP algorithms.
- The computations of the observables in the limiting tree problem, notably the KL_d of Eq. (16), cannot be computed with the usual population dynamics algorithm. As a matter of fact in the recursion of (11) the array $L_{i,i'} = L^{(d-1)}(T_i, T'_{i'})$ is strongly correlated (all the elements of the i-th row share the same tree T_i), hence one cannot write a RDE of the form $\mathbf{L}^{(d)} \stackrel{\mathrm{d}}{=} f(\{\mathbf{L}^{(d-1)}_{i,i'}\})$ with i.i.d. copies of a single random variable $\mathbf{L}^{(d-1)}$ in the right hand side. In particular the limit $d \to \infty$ cannot be described explicitly as the solution of a simple fixed-point distributional equation.

6. Conclusions

Let us conclude by summarizing our main findings and proposing some possible perspectives for further research. It would be desirable to obtain a more accurate determination of the phase transition giving the limit of the successful partial recovery of the hidden permutation by the message-passing algorithm, and to extend it to larger values of λ. This improvement could rely on better numerical procedures to perform the extrapolation at large n of the simulations on the graph, or at large d for the tree problem numerical experiments. From a more analytical point of view one could hope to either improve the bounds of [18] on $s_c(\lambda)$ to make them tighter, or to look for simplifications in some limits of the parameters. In the latter perspective, the rather modest dependency of s_{c} and s_{algo} on λ in the investigated range, and the shape of the curve in Fig. 9, could lead to the conjecture that $s_{\mathrm{c}}(\lambda)$ reaches a strictly positive value in the limit $\lambda \to \infty$. Let us further mention what is at the moment an intriguing numerical coincidence; the authors of [32] studied the detection problem associated to graph alignment, namely the hypothesis testing question of, given a pair of graphs, distinguishing their generation probability between the correlated Erdős-Rényi law and the product of two independent Erdős-Rényi laws with the same marginals as the correlated one. They defined an estimator based on the correlation of the number of trees embedded as subgraphs in the observed pairs of graphs, and characterized the range of parameters for which this estimator achieves asymptotically a vanishing probability of error under both hypotheses; in the constant-degree regime this happens for all $\lambda > 0$ whenever

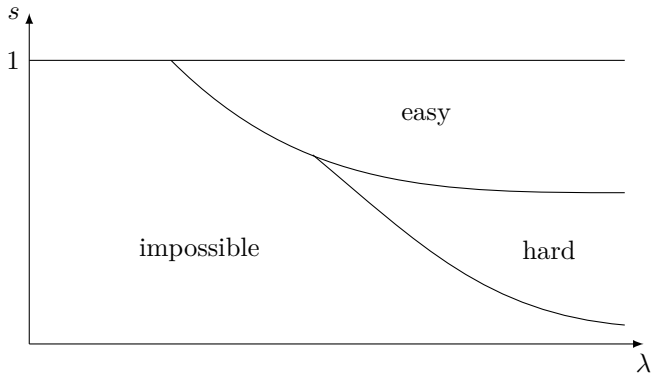

Figure 10. A sketch of the conjectured phase diagram for the alignment problem of correlated Erdős-Rényi random graphs.

$s > \sqrt{\alpha}$, where α is the Otter's constant [33] that governs the rate of growth of the number of unlabelled trees with the number of vertices (the results of [32] actually cover also denser regimes with degrees diverging with n). The numerical value of this threshold is $\sqrt{\alpha} \approx 0.5817$, indicated with an horizontal dot-dashed line in Fig. 9, just slightly below the typical values of s_c we observed, which makes this constant $\sqrt{\alpha}$ a possible candidate for the conjectured limit of $s_c(\lambda)$ when $\lambda \to \infty$.

The sketch of Fig. 10 summarizes the conjectured phase diagram for the partial recovery of the hidden permutation in the constant degree regime of the correlated Erdős-Rényi ensemble. In the easy phase partial recovery is achievable by a polynomial-time algorithm, in the hard phase it is achievable but in an a priori exponential time, while in the impossible phase the information contained in the graphs is insufficient to recover a constant fraction of the hidden permutation, even without bounds on the computational power employed. The evidences in favor of this shape of the phase diagram are on the one hand the numerical results presented in this paper, the boundary of the easy phase corresponding to the threshold $s_{\text{algo}}(\lambda)$, and on the other hand the various bounds previously obtained in the literature: $\lambda s < 1$ is a sufficient condition to be in the impossible phase [17], for $\lambda s > 4$ partial recovery is information-theoretically possible [16], hence this regime corresponds to an easy or hard phase. As the lower bound on s_c given by theorem 5 in [18] crosses the line $\lambda s = 4$ for λ large enough, a hard phase must appear in this regime. On the contrary at small λ, or more precisely for $\lambda \in [1, 1.178]$ the bounds of [18] are tight, hence the initial portion of the line $\lambda s = 1$ corresponds to a transition between the impossible and the easy phase. The upper and lower bounds on the information theoretical limit of partial recovery are relatively far apart one from the other, which leaves some room for the location of the transition line between the hard and impossible phases, and of the "triple point" where the three phases meet. One can in particular wonder about the interval of λ for which the curve $\lambda s = 1$ is the boundary of the impossible phase.

The message-passing algorithm studied in this paper was obtained by truncating the posterior, discarding the information outside the neighborhoods of radius d around the considered vertices. One could have hoped that when d is large enough, in particular for $d = \Theta(\ln n)$, the information discarded becomes negligible, as the estimator $\widehat{\pi}$ is then computed from the observation of a finite fraction of the graphs. The existence of a hard phase for this algorithm shows that this expectation is wrong, some global information that is present in the full posterior distribution is lost in the local computation, for any arbitrary large radius of observation. Note

that the notion of a hard phase is a priori related to a specific algorithm; we expect, however, that the one investigated in this paper is optimal among all local procedures, and maybe more generically for all polynomial-time algorithms for partial recovery in the sparse regime.

A possible direction for future work would be to consider ensembles of correlated random graphs more generic than the Erdős-Rényi one. In Appendix A we introduce such an ensemble that allows to tune in a flexible way the degree distribution of the generated graphs, as well as the level of correlation between them, and show that the message-passing algorithm can be generalized to this case. A particularly challenging situation is the one of regular graphs: it is known from [34] that random regular graphs do not have non-trivial automorphisms, with high probability. Hence perfect recovery is asymptotically possible for a pair of regular graphs in the noiseless case (i.e. when one observes a regular graph and a reshuffled version of it), as among all the permutations only the hidden one will achieve a perfect alignment of the two graphs; this is of course only an information-theoretic statement, the exhaustive search among all the permutations being computationally inefficient. This opens several questions that, to the best of our knowledge, have not been treated previously: can partial recovery be achieved for some pairs of partially correlated regular graphs? Can these tasks (perfect or partial recovery for noiseless or noisy regular graphs) be performed in a computationally efficient way? Note that the generalized message-passing algorithm presented in the Appendix A is completely useless on regular graphs, all trees of non-backtracking walks being regular at any depth there is strictly no local information to be exploited in this way.

Among other ensembles of correlated pairs of graphs let us also mention the case of the correlated Stochastic Block Model, whose perfect recovery in the logarithmic degree regime has been studied in [20,21]. One could also investigate these problems in the constant degree regime, and in particular study the interplay between the partial recovery of the hidden permutation and of the hidden labels. Finally there exists several "seeded" versions of the alignment problem [35–37], that are often relevant in applications, in which a side information on the hidden permutation is provided to the observer in addition to the pair of graphs, with either a part of the permutation being revealed, or under the form of an affinity matrix that favors the matching of some pairs of vertices, or some indications on strictly forbidden matchings between some subset of vertices. It should be possible to adapt the message-passing algorithm to exploit this additional information, by incorporating the latter in the truncated posterior.

Acknowledgments

We thank Luca Ganassali and Marc Lelarge for useful discussions.

Appendix A. Generalized ensembles

Correlated Erdős-Rényi random graphs have Poissonian degree distributions; however, in practical applications one often encounters the problem of aligning graphs with degree distributions significantly distinct from Poissonian. It is thus desirable to have a model of correlated random graphs with some flexibility in the degree distribution. One possibility is to draw a parent graph with an arbitrary degree distribution, and then subsample its edge set twice independently (see for instance [38] for an example of such a construction with power-law degree distributions). In this Appendix we propose a model that allows for a finer control of the degree distribution of the correlated pair of graphs, and show that the message passing algorithm described in the main text for the ER case can be adapted to this generalized ensemble. We shall follow essentially the same steps as in Sections 2 and 3, namely define the correlated graph model, study its local behavior, and deduce from it an algorithmic procedure for its alignment.

Appendix A.1. Random graphs with prescribed degree distributions

For the sake of clarity, let us start by recalling some well-known results on a procedure that allows to draw random graphs with a prescribed degree distribution, called the *configuration model*. Suppose that a distribution $P(l)$, admitting a finite second moment, is given on the non-negative integers, and we are asked to generate a random graph on n vertices with an empirical degree distribution that is close to P in the large size limit. A simple way to achieve this goal is to draw n degrees $l_1, \dots, l_n$ independently from the law P, and associate to each vertex i a number l_i of half-edges. Subsequently, we draw a uniformly random pairing of the half-edges to build the edges of the random graph. Some pairings will produce self-loops and multiple edges between pairs of vertices, but it is possible to show that the probability of generating a simple graph (i.e., a graph without self-loops and multiple edges) with this procedure remains positive in the large n limit, hence a finite (on average) number of rejections will eventually lead to a simple graph with the correct degree distribution.

The local properties of such random graphs bear some similarities with the ones of the ER model: the depth-d neighborhood of an arbitrarily chosen vertex converges with high probability to a random tree T when $n \to \infty$ with d finite. To describe the law of T let us first define another distribution on the integers,

$$\widehat{P}(l) = \frac{(l+1)P(l+1)}{\sum_{l'=0}^{\infty} l' P(l')} \ , \tag{A.1}$$

which corresponds to the size-biased version of P, also called the edge perspective degree distribution in this context. We can now characterize the law of the random tree T as follows: its root has a number l of offsprings with probability $P(l)$, each of these offsprings being the root of an independent copy of a random tree $\widehat{T}$, where in $\widehat{T}$ all vertices (including the root) have offspring distribution $\widehat{P}$. The fact that the root of T has degree distribution P follows directly from the definition of the ensemble. On the other hand, all other vertices are reached by crossing an edge, and in this exploration process the probability to end up in a vertex of degree $l+1$ is proportional to the number of half-edges around such vertices, which is itself proportional to $(l+1)P(l+1)$.

Appendix A.2. Correlated random graphs with prescribed degree distributions

Let us now introduce a model for the generation of a pair $(\mathbf{A}, \mathbf{C})$ of correlated random graphs with some prescribed degree distributions. The generation process depends on the choice of a joint distribution for three non-negative integers, $q(l_{\rm b}, l_{\rm r}, l_{\rm bi})$, which we assume has finite second moments, and fulfills the property $q(l_{\rm b}, l_{\rm r}, l_{\rm bi}) = q(l_{\rm r}, l_{\rm b}, l_{\rm bi})$. For each vertex $i \in [n]$ one draws a triplet $(l_{{\rm b},i}, l_{{\rm r},i}, l_{{\rm bi},i})$ i.i.d. from the law q, in such a way that, from vertex i, a number $l_{{\rm b},i}$ (resp. $l_{{\rm r},i}$, $l_{{\rm bi},i}$) of blue (resp. red, bicolored) half-edges emerges. One then draws three uniform pairings of these three type of half-edges, in order to produce a colored graph $\mathbf{G}$ with three types of edges (if $\mathbf{G}$ is not simple, the generation process is restarted). A pair of graphs $(\mathbf{A}, \mathbf{C})$ is finally produced from $\mathbf{G}$ by keeping in $\mathbf{A}$ (resp. in $\mathbf{C}$) the blue and bicolored (resp. red and bicolored) edges of $\mathbf{G}$. The degree distribution of $\mathbf{A}$ and $\mathbf{C}$ is then easily seen to be

$$P(l) \coloneqq \sum_{l_{\rm bi}=0}^{l} \sum_{l_{\rm r}=0}^{\infty} q(l - l_{\rm bi}, l_{\rm r}, l_{\rm bi}) \ ; \tag{A.2}$$

by virtue of the property $q(l_{\rm b}, l_{\rm r}, l_{\rm bi}) = q(l_{\rm r}, l_{\rm b}, l_{\rm bi})$, the graph $\mathbf{A}$ and $\mathbf{C}$ have the same degree distribution. As we shall see below the law of $\mathbf{A}$ is not in general the same as the single configuration model with degree distribution P.

One can then create a graph $\mathbf{B}$ with $\mathbf{C}_{ij} = \mathbf{B}_{\boldsymbol{\pi}_\star(i),\boldsymbol{\pi}_\star(j)}$ where $\boldsymbol{\pi}_\star$ is a uniformly random permutation, and study the inference problem of recovering $\boldsymbol{\pi}_\star$ from the observation of $(\mathbf{A}, \mathbf{B})$. As in the ER case we will propose an approximate estimator of $\boldsymbol{\pi}_\star$ based on the local properties of the graphs, that we shall now describe. In order to do this we define the following size-biased versions of q:

$$\widehat{q}(l_{\rm b}, l_{\rm r}, l_{\rm bi}) \coloneqq \frac{(l_{\rm bi}+1)\, q(l_{\rm b}, l_{\rm r}, l_{\rm bi}+1)}{\sum_{l'_{\rm b},l'_{\rm r},l'_{\rm bi}} l'_{\rm bi}\, q(l'_{\rm b}, l'_{\rm r}, l'_{\rm bi})} \ , \tag{A.3}$$

$$\widetilde{q}(l_{\rm b}, l_{\rm r}, l_{\rm bi}) \coloneqq \frac{(l_{\rm b}+1)\, q(l_{\rm b}+1, l_{\rm r}, l_{\rm bi})}{\sum_{l'_{\rm b},l'_{\rm r},l'_{\rm bi}} l'_{\rm b}\, q(l'_{\rm b}, l'_{\rm r}, l'_{\rm bi})} \ , \tag{A.4}$$

$$\dot{q}(l_{\rm b}, l_{\rm r}, l_{\rm bi}) \coloneqq \widetilde{q}(l_{\rm r}, l_{\rm b}, l_{\rm bi}) \ . \tag{A.5}$$

Let us call $\mathcal{T}$ the colored branching process obtained by a local exploration of $\mathbf{G}$ from an arbitrary root vertex. Generalizing the construction given in the main text for the ER case, and recalling the explanations on the appearance of size-biased versions of the degree distribution given above in the single graph case, one realizes that the law of $\mathcal{T}$ can be described as follows: the root of $\mathcal{T}$ has $l_{\rm b}$ offsprings linked to it by a blue edge, $l_{\rm r}$ by a red edge and $l_{\rm bi}$ by a bicolored edge, with probability $q(l_{\rm b}, l_{\rm r}, l_{\rm bi})$. Each of these $l_{\rm b}$ (resp. $l_{\rm r}$, $l_{\rm bi}$) offsprings is the root of an independent copy of the tree $\widetilde{\mathcal{T}}$ (resp. $\dot{\mathcal{T}}$, $\widehat{\mathcal{T}}$), where the law of $\widetilde{\mathcal{T}}$ (resp. $\dot{\mathcal{T}}$, $\widehat{\mathcal{T}}$) follows exactly the same definition as the one of $\mathcal{T}$ with the replacement of the distribution q by $\widetilde{q}$ (resp. $\dot{q}$, $\widehat{q}$).

Consider now the question investigated in Section 2.2.3 in the ER case, namely the joint law of (T, T') of the neighborhood T of i in $\mathbf{A}$ and T' of i' in $\mathbf{B}$, where i and i' are aligned vertices. As in the main text $(T, T') = (b(\mathcal{T}), r(\mathcal{T}))$, where $b(\mathcal{T})$ (resp. $r(\mathcal{T})$) returns the connected component of the root of the blue and bicolored (resp. red and bicolored) edges of $\mathcal{T}$, neighborhood of i in $\mathbf{G}$. From the recursive description of the law of $\mathcal{T}$ given above we can deduce a recursive expression of the law of (T, T'). More precisely, we shall denote $\mathbb{P}_1^{(d)}[T, T']$ the law of $(b(\mathcal{T}), r(\mathcal{T}))$, $\widehat{\mathbb{P}}_1^{(d)}[T, T']$ the law of $(b(\widehat{\mathcal{T}}), r(\widehat{\mathcal{T}}))$, and $\widetilde{\mathbb{P}}_1^{(d)}[T, T']$ the law of $(b(\widetilde{\mathcal{T}}), r(\widetilde{\mathcal{T}}))$, with $\mathcal{T}$, $\widehat{\mathcal{T}}$ and $\widetilde{\mathcal{T}}$ the random colored trees defined previously, when observed up to the d-th generation. We will also denote $\mathbb{P}_0^{(d)}[T]$, $\widehat{\mathbb{P}}_0^{(d)}[T]$ and $\widetilde{\mathbb{P}}_0^{(d)}[T]$ their marginals obtained by summing over T'. The translation of the recursive definition of $\mathcal{T}$, exploiting the fact that once a monochromatic edge of $\mathcal{T}$ has been crossed one can discard the edges of the opposite color below it, and the symmetry between the colors, yields:

$$\begin{aligned}
\mathbb{P}_1^{(d)}[T, T'] = \sum_{l_{\rm b},l_{\rm r},l_{\rm bi}=0}^{\infty} & q(l_{\rm b}, l_{\rm r}, l_{\rm bi})\, \mathbb{I}(l = l_{\rm b} + l_{\rm bi})\ \mathbb{I}(l' = l_{\rm r} + l_{\rm bi}) \\
& \sum_{Q_1,\dots,Q_{l_{\rm b}}} \sum_{R_1,\dots,R_{l_{\rm r}}} \sum_{S_1,S'_1,\dots,S_{l_{\rm bi}},S'_{l_{\rm bi}}} \prod_{i=1}^{l_{\rm b}} \widetilde{\mathbb{P}}_0^{(d-1)}[Q_i] \prod_{i=1}^{l_{\rm r}} \widetilde{\mathbb{P}}_0^{(d-1)}[R_i] \prod_{i=1}^{l_{\rm bi}} \widehat{\mathbb{P}}_1^{(d-1)}[S_i, S'_i] \\
& \frac{1}{l!l'!} \sum_{\pi,\pi'} \mathbb{I}((T_1,\dots,T_l) = \pi(Q_1,\dots,Q_{l_{\rm b}},S_1,\dots,S_{l_{\rm bi}}))\, \mathbb{I}((T'_1,\dots,T'_{l'}) = \pi'(R_1,\dots,R_{l_{\rm r}},S'_1,\dots,S'_{l_{\rm bi}})) \ ,
\end{aligned}$$

where l (resp. l') is the degree of the root of T (resp. T'), and $T_1, \dots, T_l$ (resp. $T'_1, \dots, T'_{l'}$) are the subtrees rooted below it, π (resp. π') is a permutation of its l (resp. l') arguments, and we use the convention $\mathbb{P}_1^{(0)}[T, T'] = 1$. This can be simplified into a generalization of the equation

(2) of the ER case,

$$\mathbb{P}_1^{(d)}[T,T'] = \sum_{l_{\rm bi}=0}^{\min(l,l')} \frac{q(l-l_{\rm bi},l'-l_{\rm bi},l_{\rm bi})}{\binom{l}{l_{\rm bi}}\binom{l'}{l_{\rm bi}} l_{\rm bi}!} \sum_{I,I',\sigma} \prod_{i\in I} \widehat{\mathbb{P}}_1^{(d-1)}[T_i,T'_{\sigma(i)}] \prod_{i\in[l]\setminus I} \widetilde{\mathbb{P}}_0^{(d-1)}[T_i] \prod_{i\in[l']\setminus I'} \widetilde{\mathbb{P}}_0^{(d-1)}[T'_i] \ , \tag{A.6}$$

where I (resp. I') is a subset of $[l]$ (resp. of $[l']$) of $l_{\rm bi}$ elements, and σ a bijection from I to I'. The expressions of $\widehat{\mathbb{P}}_1^{(d)}[T,T']$ and $\widetilde{\mathbb{P}}_1^{(d)}[T,T']$ are of the same form, the only modification being the replacement of the law q by $\widehat{q}$ and $\widetilde{q}$ in the right hand side of (A.6).

Summing over T' yields a recursive expression of the marginal law,

$$\mathbb{P}_0^{(d)}[T] = \sum_{l_{\rm bi}=0}^{l} \frac{q(l-l_{\rm bi},l_{\rm bi})}{\binom{l}{l_{\rm bi}}} \sum_{I} \prod_{i\in I} \widehat{\mathbb{P}}_0^{(d-1)}[T_i] \prod_{i\in[l]\setminus I} \widetilde{\mathbb{P}}_0^{(d-1)}[T_i] \ , \tag{A.7}$$

where here and in the following a law q, $\widehat{q}$ and $\widetilde{q}$ with only two arguments is understood to be the the marginal when the number of red edges has been discarded, e.g.

$$q(l_{\rm b},l_{\rm bi}) = \sum_{l_{\rm r}=0}^{\infty} q(l_{\rm b},l_{\rm r},l_{\rm bi}) \ . \tag{A.8}$$

As in the joint case $\widehat{\mathbb{P}}_0^{(d)}[T]$ and $\widetilde{\mathbb{P}}_0^{(d)}[T]$ are given by (A.7) with the law q in the right hand side replaced by $\widehat{q}$ and $\widetilde{q}$ respectively.

For a generic choice of the distribution q the size-biased versions $\widehat{q}(l_{\rm b},l_{\rm bi})$ and $\widetilde{q}(l_{\rm b},l_{\rm bi})$ differ, hence $\widehat{\mathbb{P}}_0^{(d)} \neq \widetilde{\mathbb{P}}_0^{(d)}$; in this case (A.7) cannot be written as the product of the degree distribution $P(l)$ multiplied by the product of the probabilities of the subtrees, which shows that the law of $\mathbf{A}$ is not the same as the one of the configuration model with the degree distribution $P(l)$ defined in (A.2). If instead q is such that $\widehat{q}(l_{\rm b},l_{\rm bi}) = \widetilde{q}(l_{\rm b},l_{\rm bi})$, then by induction on d one finds that $\widehat{\mathbb{P}}_0^{(d)} = \widetilde{\mathbb{P}}_0^{(d)}$, hence one can simplify (A.7) and its counterpart for $\widehat{\mathbb{P}}_0^{(d)}$ into

$$\mathbb{P}_0^{(d)}[T] = \left(\sum_{l_{\rm bi}=0}^{l} q(l-l_{\rm bi},l_{\rm bi})\right) \prod_{i=1}^{l} \widehat{\mathbb{P}}_0^{(d-1)}[T_i] \ , \qquad \widehat{\mathbb{P}}_0^{(d)}[T] = \left(\sum_{l_{\rm bi}=0}^{l} \widehat{q}(l-l_{\rm bi},l_{\rm bi})\right) \prod_{i=1}^{l} \widehat{\mathbb{P}}_0^{(d-1)}[T_i] \ , \tag{A.9}$$

which corresponds to the branching process description of the configuration model with the degree distribution $P(l)$ of (A.2). A short computation reveals indeed that $\sum_{l_{\rm bi}=0}^{l} \widehat{q}(l-l_{\rm bi},l_{\rm bi})$ is the size bias of $\sum_{l_{\rm bi}=0}^{l} q(l-l_{\rm bi},l_{\rm bi})$ in this case.

These simplifications occur in particular when q is a product of three Poisson distributions for the three types of degrees: as a Poisson distribution is equal to its size-bias one has then $q = \widehat{q} = \widetilde{q} = \dot{q}$ and the equations (A.7) and (A.6) reduce to their counterparts (1) and (2) derived in the ER model.

Appendix A.3. Generalization of the message passing algorithm

Consider now the inference problem of recovering the unknown permutation $\boldsymbol{\pi}_\star$ from the observation of $(\mathbf{A},\mathbf{B})$ generated with the correlated graph ensemble for an arbitrary distribution q. The message passing algorithm described in the main text for the correlated ER ensemble can be naturally extended to handle this case. As a matter of fact the reasoning linking the truncated posterior probability $\boldsymbol{\pi}_\star(i) = i'$ to the likelihood ratio of the neighborhoods (T,T') of the vertices i and i' formalized in (9) and (10) applies verbatim to the generalized ensemble, thanks to its

similar local convergence properties towards random trees. A slight complication occurs because in general the null hypothesis $\mathbb{P}_0^{(d)}[T]$ does not factorize as a product over its subtrees, hence one cannot write directly a recursion relation on $L^{(d)}(T,T')$, but in any case both $\mathbb{P}_0^{(d)}[T]$ and $\mathbb{P}_1^{(d)}[T,T']$ can be computed recursively thanks to (A.6) and (A.7)). To describe in a more compact way the algorithm let us first define a function f_1 that takes as input two integers l and l', an $l \times l'$ array M, a l-dimensional vector u and a l'-dimensional vector u', as

$$f_1(l,l';M,u,u') = \sum_{l_{\rm bi}=0}^{\min(l,l')} \frac{q(l-l_{\rm bi},l'-l_{\rm bi},l_{\rm bi})}{\binom{l}{l_{\rm bi}}\binom{l'}{l_{\rm bi}} l_{\rm bi}!} \sum_{I,I',\sigma} \prod_{i\in I} M_{i,\sigma(i)} \prod_{i\in[l]\setminus I} u_i \prod_{i\in[l']\setminus I'} u'_i \ , \tag{A.10}$$

where the meaning of I, I' and σ are the same as in (A.6). The function $\widehat{f}_1(l,l';M,v,v')$ is defined by the same expression where q is replaced by $\widehat{q}$. We further define a function f_0 with arguments an integer l and two l-dimensional vectors u and v, according to

$$f_0(l;u,v) = \sum_{l_{\rm bi}=0}^{l} \frac{q(l-l_{\rm bi},l_{\rm bi})}{\binom{l}{l_{\rm bi}}} \sum_{I} \prod_{i\in I} u_i \prod_{i\in[l]\setminus I} v_i \ , \tag{A.11}$$

along with $\widehat{f}_0$ and $\widetilde{f}_0$ defined similarly with q replaced by $\widehat{q}$ and $\widetilde{q}$ respectively.

The generalized algorithm uses the following sets of messages:

- $m^{(t)}_{ii'\to jj'}$ for all vertices i of A, all vertices i' of B, all $j\in\partial i$, and all $j'\in\partial i'$;
- $\widehat{m}^{(t)}_{i\to j}$ and $\widetilde{m}^{(t)}_{i\to j}$ for all vertices i of A and all $j\in\partial i$;
- $\widehat{m}^{(t)}_{i'\to j'}$ and $\widetilde{m}^{(t)}_{i'\to j'}$ for all vertices i' of B and all $j'\in\partial i'$.

All of them are initialized to 1 at $t=0$, and computed by induction for $t=1,\dots,d-1$ as:

$$m^{(t)}_{ii'\to jj'} = \widehat{f}_1(d_i-1,d_{i'}-1;\{m^{(t-1)}_{kk'\to ii'}\},\{\widetilde{m}^{(t-1)}_{k\to i}\},\{\widetilde{m}^{(t-1)}_{k'\to i'}\} : k\in\partial i\setminus j, k'\in\partial i'\setminus j'\}) \ , \tag{A.12}$$

$$\widehat{m}^{(t)}_{i\to j} = \widehat{f}_0(d_i-1;\{\widehat{m}^{(t-1)}_{k\to i}\},\{\widetilde{m}^{(t-1)}_{k\to i}\}\colon k\in\partial i\setminus j) \ , \tag{A.13}$$

$$\widetilde{m}^{(t)}_{i\to j} = \widetilde{f}_0(d_i-1;\{\widehat{m}^{(t-1)}_{k\to i}\},\{\widetilde{m}^{(t-1)}_{k\to i}\}\colon k\in\partial i\setminus j) \ , \tag{A.14}$$

$$\widehat{m}^{(t)}_{i'\to j'} = \widehat{f}_0(d_{i'}-1;\{\widehat{m}^{(t-1)}_{k'\to i'}\},\{\widetilde{m}^{(t-1)}_{k'\to i'}\}\colon k'\in\partial i'\setminus j') \ , \tag{A.15}$$

$$\widetilde{m}^{(t)}_{i'\to j'} = \widetilde{f}_0(d_{i'}-1;\{\widehat{m}^{(t-1)}_{k'\to i'}\},\{\widetilde{m}^{(t-1)}_{k'\to i'}\}\colon k'\in\partial i'\setminus j') \ . \tag{A.16}$$

All messages are assumed to be initialized to 1. Finally the scores are given by

$$L^{(d)}_{ii'} = \frac{f_1(d_i,d_{i'};\{m^{(d-1)}_{jj'\to ii'}\},\{\widetilde{m}^{(d-1)}_{j\to i}\},\{\widetilde{m}^{(d-1)}_{j'\to i'}\}\colon j\in\partial i\,,\,j'\in\partial i'\})}{f_0(d_i;\{\widehat{m}^{(d-1)}_{j\to i}\},\{\widetilde{m}^{(d-1)}_{j\to i}\}\colon j\in\partial i)\ f_0(d_{i'};\{\widehat{m}^{(d-1)}_{j'\to i'}\},\{\widetilde{m}^{(d-1)}_{j'\to i'}\}\colon j'\in\partial i')} \ , \tag{A.17}$$

from which the estimator $\widehat{\pi}$ is constructed by matching each vertex i with the vertex i' maximizing $L^{(d)}_{ii'}$, which implements exactly the same strategy than in the ER case.

Note that the flexibility of the random graph ensemble presented in this Appendix allows in particular to generate pairs of k-regular random graphs with a law that interpolates smoothly between the completely uncorrelated case (when $q(l_{\rm b},l_{\rm r},l_{\rm bi}) = \delta_{l_{\rm b},k}\delta_{l_{\rm r},k}\delta_{l_{\rm bi},0}$) and the perfectly correlated case where the two graphs are identical (with $q(l_{\rm b},l_{\rm r},l_{\rm bi}) = \delta_{l_{\rm b},0}\delta_{l_{\rm r},0}\delta_{l_{\rm bi},k}$). However, in this regular case the message passing algorithm is completely inefficient: all trees of non-backtracking walk are regular for any value of d and around any vertex, the scores between any pair of vertices will thus be all equal, hence contain no information on the signal $\boldsymbol{\pi}_\star$.

Appendix A.4. Weighted random graphs

The matrix alignment problem, studied for instance in [12, 13, 16, 22, 23], concerns the recovery of an unknown permutation $\boldsymbol{\pi}_\star$ from the observation of a pair of matrices $(\mathbf{A}, \mathbf{B})$, where $\mathbf{B}$ is obtained from $\mathbf{C}$ through the action of $\boldsymbol{\pi}_\star$ on the row and column indices, while the matrix elements $(\mathbf{A}_{ij}, \mathbf{C}_{ij})$ are pairs of correlated random variables. It corresponds thus to the graph alignment problem for weighted complete graphs.

We shall now briefly discuss a further generalization of our approach to the case of sparse correlated weighted graphs. Let us consider a joint probability density $\rho(w, w')$, symmetric under the exchange of its arguments, and with a marginal law $\rho_{\rm m}(w) = \int \mathrm{d}\, w' \rho(w, w')$. We modify the correlated random graph ensembles as follows. From the colored random graph $\mathbf{G}$ (drawn either from the ER ensemble or from its generalization with a prescribed degree distribution) we derive a pair of weighted graphs $(\mathbf{A}, \mathbf{C})$ by, for each pair $i < j$ of vertices:

- drawing $(\mathbf{A}_{ij}, \mathbf{C}_{ij})$ from ρ if the edge $\{i, j\}$ is bicolored in $\mathbf{G}$;
- drawing $\mathbf{A}_{ij}$ from $\rho_{\rm m}$ and setting $\mathbf{C}_{ij} = 0$ if the edge $\{i, j\}$ is blue in $\mathbf{G}$;
- drawing $\mathbf{C}_{ij}$ from $\rho_{\rm m}$ and setting $\mathbf{A}_{ij} = 0$ if the edge $\{i, j\}$ is red in $\mathbf{G}$;
- setting $\mathbf{A}_{ij} = \mathbf{C}_{ij} = 0$ if the edge $\{i, j\}$ is absent from $\mathbf{G}$.

The description of the local properties of the graphs $(\mathbf{A}, \mathbf{B})$ can then be adapted to this weighted setting. In particular the law $\mathbb{P}_1^{(d)}[T, T']$ for the (now weighted) tree neighborhoods of two aligned vertices in $\mathbf{A}$ and $\mathbf{B}$ admit a recursive decomposition that follows from the inclusion of the weight distribution in (A.6), namely

$$\mathbb{P}_1^{(d)}[T, T'] = \sum_{l_{\rm bi}=0}^{\min(l,l')} \frac{q(l - l_{\rm bi}, l' - l_{\rm bi}, l_{\rm bi})}{\binom{l}{l_{\rm bi}}\binom{l'}{l_{\rm bi}} l_{\rm bi}!} \sum_{I, I', \sigma} \prod_{i \in I} \rho(w_i, w'_{\sigma(i)}) \widehat{\mathbb{P}}_1^{(d-1)}[T_i, T'_{\sigma(i)}] \prod_{i \in [l] \setminus I} \rho_{\rm m}(w_i) \widetilde{\mathbb{P}}_0^{(d-1)}[T_i] \prod_{i \in [l'] \setminus I'} \rho_{\rm m}(w'_i) \widetilde{\mathbb{P}}_0^{(d-1)}[T'_i] \,, \quad \text{(A.18)}$$

where w_i (resp., w'_i) is the weight of the edge between the root of T (resp., of T') and its i-th offspring. The message-passing algorithm can then be straightforwardly adapted to incorporate the information coming from these weights. Denoting w_{ij} and $w'_{i'j'}$ the weights on the edges of the observed graphs A and B, the following update equations can be derived

$$\begin{aligned}
m^{(t)}_{ii' \to jj'} &= \rho(w_{ij}, w'_{i'j'}) \widehat{f}_1(d_i - 1, d_{i'} - 1; \{m^{(t-1)}_{kk' \to ii'}\}, \{\widetilde{m}^{(t-1)}_{k \to i}\} \,, \{\widetilde{m}^{(t-1)}_{k' \to i'}\} : k \in \partial i \setminus j, k' \in \partial i' \setminus j'), \\
\widehat{m}^{(t)}_{i \to j} &= \rho_{\rm m}(w_{ij}) \widehat{f}_0(d_i - 1; \{\widehat{m}^{(t-1)}_{k \to i}\}, \{\widetilde{m}^{(t-1)}_{k \to i}\} : k \in \partial i \setminus j) \,, \\
\widetilde{m}^{(t)}_{i \to j} &= \rho_{\rm m}(w_{ij}) \widetilde{f}_0(d_i - 1; \{\widehat{m}^{(t-1)}_{k \to i}\}, \{\widetilde{m}^{(t-1)}_{k \to i}\} : k \in \partial i \setminus j) \,, \\
\widehat{m}^{(t)}_{i' \to j'} &= \rho_{\rm m}(w'_{i'j'}) \widehat{f}_0(d_{i'} - 1; \{\widehat{m}^{(t-1)}_{k' \to i'}\}, \{\widetilde{m}^{(t-1)}_{k' \to i'}\} : k' \in \partial i' \setminus j') \,, \\
\widetilde{m}^{(t)}_{i' \to j'} &= \rho_{\rm m}(w'_{i'j'}) \widetilde{f}_0(d_{i'} - 1; \{\widehat{m}^{(t-1)}_{k' \to i'}\}, \{\widetilde{m}^{(t-1)}_{k' \to i'}\} : k' \in \partial i' \setminus j') \,,
\end{aligned} \quad \text{(A.19)}$$

with initial conditions $\widehat{m}^{(0)}_{i \to j} = \widetilde{m}^{(0)}_{i \to j} = \rho_{\rm m}(w_{ij})$, $\widehat{m}^{(0)}_{i' \to j'} = \widetilde{m}^{(0)}_{i' \to j'} = \rho_{\rm m}(w'_{i'j'})$, $m^{(0)}_{ii' \to jj'} = \rho(w_{ij}, w'_{i'j'})$, while the computation of the scores keeps the form in (A.17), from which the estimator can be built by the row maximization procedure.

Note that when $\rho(w, w') = \rho_{\rm m}(w)\rho_{\rm m}(w')$ the contributions of the weights cancel out in the computation of the score, which coincides then with the unweighted computation. Indeed in this case the weights bring no information on whether the edges were aligned or not. The case $\rho(w, w') = \rho_{\rm m}(w)\delta(w - w')$ with $\rho_{\rm m}$ absolutely continuous is the opposite limit: edges

with equal weights in A and B were certainly bicolored in G and aligned, those with different weights are certainly not (this would signal itself as zeros and formal infinities in the message passing algorithm). The level of correlation in ρ allows to tune the amount of information on the alignment provided by the weights between these two limit cases.

Appendix B. Further numerical results

In this Appendix we present a series of additional results obtained from the numerical simulations of the message-passing algorithm on correlated Erdős-Rényi random graphs.

Appendix B.1. A comparison of the scores between pairs of aligned and quasi-aligned vertices

Our derivation of the message-passing algorithm given in Eq. (9) started by the replacement of the posterior distribution by its truncated version; once this approximation had been made we wrote the truncated posterior in terms of the probability laws $\mathbb{P}_1^{(d)}$ and $\mathbb{P}_0^{(d)}$, which is asymptotically exact according to the local analysis of the correlated random graph ensemble. Finally, we constructed an estimator by maximizing the marginal posterior probability. If the full posterior distribution were used, this procedure would maximize the average overlap with the ground truth. The scores derived in this way can be interpreted as the likelihood ratios of an hypothesis testing problem between correlated and uncorrelated pairs of trees, even if this was not used in the derivation of the algorithm itself. This perspective motivates further investigations on the properties of the neighborhoods compared by the algorithm. For example, some pairs of neighborhoods in the graphs are neither drawn from the correlated law $\mathbb{P}_1^{(d)}$ nor from the uncorrelated product of the laws $\mathbb{P}_0^{(d)}$. To be more precise, let us consider a vertex i of the graph $\mathbf{A}$, its image $\boldsymbol{\pi}_\star(i)$ in $\mathbf{B}$ under the groundtruth permutation, and a neighbor of the latter $i' \in \partial\boldsymbol{\pi}_\star(i)$; we shall call in the following (i, i') a *quasi-aligned pair of vertices*. It should be clear that the law of the neighborhoods $(T_i, T'_{i'})$ is neither $\mathbb{P}_1^{(d)}$, because $i' \neq \boldsymbol{\pi}_\star(i)$, nor the product of $\mathbb{P}_0^{(d)}$, because i and i' correspond to vertices at distance 1 in the colored graph $\mathbf{G}$, hence their neighborhoods overlap in general via the bicolored edges of $\mathbf{G}$. The presence of this correlation could suggest that the scores $L_{i,\boldsymbol{\pi}_\star(i)}$ of an aligned pair and $L_{i,i'}$ with $i' \in \partial\boldsymbol{\pi}_\star(i)$ of a quasi-aligned pair are of the same order, hence that the algorithm could easily mistake one for the other. In order to investigate the possible presence of an issue due to this correlation, we estimated the typical values of the scores for aligned and quasi-aligned pairs of vertices, by computing the following averages of the logarithm of the scores:

$$A_1 = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \ln L^{(d)}_{i,\widehat{\boldsymbol{\pi}}(i)}\right] , \tag{B.1}$$

$$A_2 = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \ln L^{(d)}_{i,\boldsymbol{\pi}_\star(i)}\right] , \tag{B.2}$$

$$A_3 = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\partial\boldsymbol{\pi}_\star(i)|}\sum_{i'\in\partial\boldsymbol{\pi}_\star(i)} \ln L^{(d)}_{i,i'}\right] , \tag{B.3}$$

$$A_4 = \mathbb{E}\left[\frac{1}{n^2}\sum_{i,i'=1}^{n} \ln L^{(d)}_{i,i'}\right] . \tag{B.4}$$

The averages are intended over the graphs realizations and over pairs of vertices which are, respectively, the ones selected by the estimator, A_1, the ones in the groundtruth, A_2, the quasi-

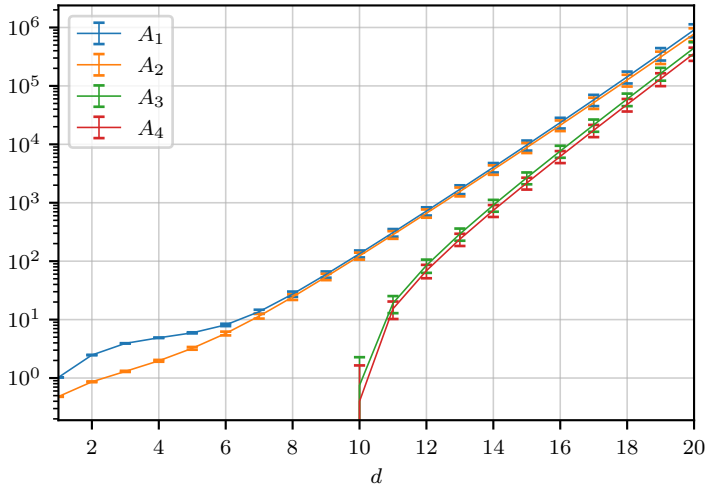

Figure B1. The various averages of logarithm of scores A_1, A_2, A_3, A_4 defined in Eqs. (B.1-B.4) as a function of the depth d, for system size $n = 2048$, average degree $\lambda = 2.9$, correlation $s = 0.79$. Each point is averaged over 15 independent realizations of the two graphs $\mathbf{A}, \mathbf{B}$.

aligned ones A_3, and finally randomly selected pairs A_4. The numerical estimation of these averages is presented in Fig. B1, which shows that $A_1 \geq A_2 \gg A_3 \approx A_4$. The fact that $A_2 \gg A_3 \approx A_4$ is very reassuring and should dissipate the concern raised above: pairs of quasi-aligned vertices have scores substantially smaller than the aligned ones, which avoids the possible confusion between them, and of the same order as arbitrarily distant vertices. An interpretation of this result proceeds as follows. The function $L(T, T')$ is, forgetting its precise definition, a function of the structure of the rooted trees T and T'. Given a tree T and two adjacent vertices a and b, the ordered structure of the same tree T rooted in a is very different from the one of the tree rooted in b (the roots themselves might have different degrees, and no common subtree might appear), hence a priori the two structures have very different images under the function $L(\cdot, T')$. Interestingly, this argument breaks down if a and b are vertices at distance 2: consider indeed the path $a - c - b$ in T, the tree rooted at a and the tree rooted at b. The two trees will share a subtree rooted in c. We have checked that indeed the average of the logarithm of the scores between i and a vertex at distance 2 from $\boldsymbol{\pi}_\star(i)$ in B is larger than A_3, but still notably smaller than A_2.

Appendix B.2. Alternative estimators

As we discussed at the beginning of Section 3, the notion of optimal estimator in inference problems depends on the choice of the distance between the estimator and the signal that is to be minimized. In the main text we chose an estimator $\widehat{\pi}$ that, for each node i of the graph A, maximizes the (approximate) probability of the event $\widehat{\pi}(i) = \boldsymbol{\pi}_\star(i)$, aiming at maximizing the average overlap between $\widehat{\boldsymbol{\pi}}$ and $\boldsymbol{\pi}_\star$, as defined in (5). We will consider here other estimators, devised to optimize other error measures.

Before giving some examples, let us introduce additional definitions that will be useful in the discussion. We shall consider partial estimators, namely functions $\widehat{\pi}$ from $[n]$ to $[n] \cup \{*\}$, where $*$ is an additional dummy symbol, such that $\widehat{\pi}(i) = *$ whenever the partial estimator does

not propose any vertex of the graph B to be matched with the vertex i of A. We shall denote $\mathcal{S}(\widehat{\pi}) = \{i \in [n] \ : \ \widehat{\pi}(i) \neq *\}$ the set of vertices that are assigned by $\widehat{\pi}$, $n(\widehat{\pi}) = |\mathcal{S}(\widehat{\pi})|$ their number, and define the overlap of a partial estimator $\widehat{\pi}$ with a permutation π_* of $[n]$ as

$$\mathrm{ov}(\widehat{\pi}, \pi_*) = \frac{1}{n(\widehat{\pi})} \sum_{i \in \mathcal{S}(\widehat{\pi})} \mathbb{I}(\widehat{\pi}(i) = \pi_*(i)) \ . \tag{B.5}$$

This counts the fraction of correct matches among the assigned ones, and coincides with the definition in Eq. (5) when $\widehat{\pi}$ assigns all the vertices.

Let us also recall the notation $P_{i,i'} = \mathbb{P}(\boldsymbol{\pi_\star}(i) = i' | \mathbf{A} = A, \mathbf{B} = B)$ introduced in Section 3, whose row and column sums are normalized to 1, and in which we keep implicit the dependency on the observed graphs A and B. As this marginal of the exact posterior is not efficiently computable we will use its approximation $\widehat{P}_{i,i'}$ obtained from the message passing algorithm. According to its derivation in terms of the truncated posterior, $\widehat{P}_{i,i'}$ should be proportional to the score matrix $L^{(d)}_{i,i'}$; to ensure one type of normalization we will define

$$\widehat{P}_{i,i'} = \frac{L^{(d)}_{i,i'}}{\sum_{j'} L^{(d)}_{i,j'}} \ , \tag{B.6}$$

which by definition satisfies the same row normalization as the exact quantity P, but because of the approximation may violate the column normalization.

Matrix estimator Let us consider the following distance (or loss function) between a (possibly partial) estimator $\widehat{\pi}$ and a permutation π_*:

$$\mathscr{L}(\widehat{\pi}, \pi_*) = \left(1 - \frac{n(\widehat{\pi})}{n}\right) + \frac{2n(\widehat{\pi})}{n} \left(1 - \mathrm{ov}(\widehat{\pi}, \pi_*)\right) \ , \tag{B.7}$$

which coincides with the Hamming distance between the matrix representation of $\widehat{\pi}$ and π_* as discussed in Eq. (4) (the Hamming distance between an empty row and a row containing exactly one 1 is 1, while it is 2 between two rows containing exactly one 1 at different positions). Minimizing $\mathscr{L}$ amounts to find a compromise between the two terms: the first one favors estimators that assign the largest possible number of vertices, but the second one grows if too many of these assignment are erroneous. The error $\mathscr{L}$ satisfies the bounds $0 \leq \mathscr{L}(\widehat{\pi}, \pi_*) \leq 2$, with $\mathscr{L} = 2$ if and only if all the vertices are assigned and are all incorrect, and $\mathscr{L} = 1$ for the null estimator which does not assign any of the vertices.

The estimator that minimizes this loss on average is built from the posterior probabilities $P_{i,i'}$ as

$$\widehat{\boldsymbol{\pi}}_{1/2}(i) = \begin{cases} i' & \text{if} \quad P_{i,i'} > \frac{1}{2} \\ * & \text{if} \quad P_{i,j'} \leq \frac{1}{2} \quad \forall \ j' \end{cases} \ ; \tag{B.8}$$

because of the normalization condition at most one i' can be selected in the first line. The numerical results presented in Fig. B2 have been obtained by replacing in this expression P by its approximation $\widehat{P}$ defined in Eq. (B.6), and selecting for each set of parameters the optimal value of the depth d (the one that minimized the average loss). The curves in Fig. B2 shows the average error $\mathscr{L}$ between this estimator and the groundtruth as a function of s, for different values of n. The error decreases below $\mathscr{L} = 1$ for values of s around 0.7, slightly above the threshold $s_{\mathrm{algo}} \approx 0.6$ we observed in the main part of the text for the estimator $\widehat{\boldsymbol{\pi}}$: as a matter of fact in between these two values of s the algorithm is not confident enough about the quality

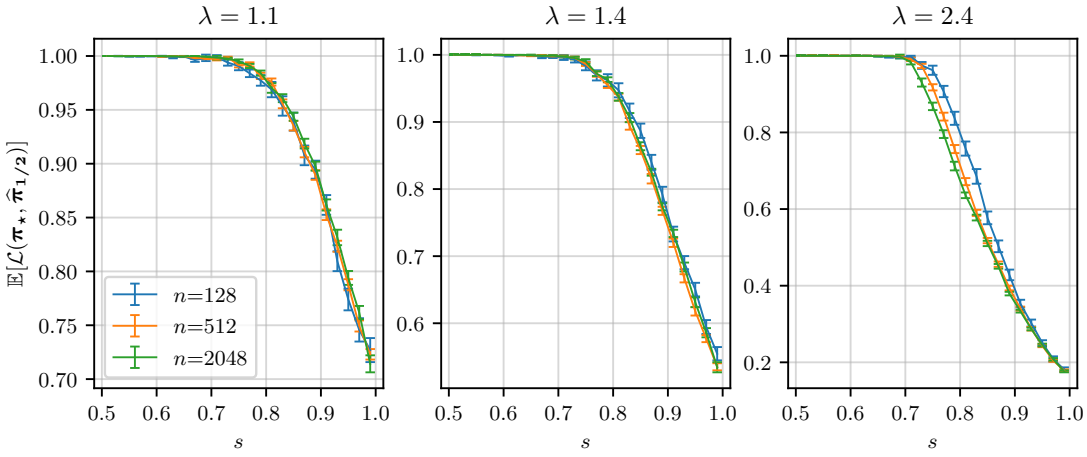

Figure B2. The average loss $\mathbb{E}[\mathcal{L}(\widehat{\boldsymbol{\pi}}_{1/2}, \boldsymbol{\pi}_\star)]$ as a function of the correlation s for different values of the system size n. In increasing order of n we averaged over $100, 100, 25$ independent realizations of the two graphs $\mathbf{A}, \mathbf{B}$.

of its predictions (in more technical terms all $\widehat{P}_{i,i'}$ are below $1/2$), hence it prefers to return the null estimator instead of some possibly erroneous matches. One can also observe that the finite size effects seem to be much weaker for this estimator than for $\widehat{\boldsymbol{\pi}}$, as the curves for various n are almost superimposed in Fig. B2, to be compared for instance with Fig. 5.

Thresholded estimator We will now discuss a generalization of the estimator $\widehat{\boldsymbol{\pi}}_{1/2}$ that amounts in some sense to tune the relative weights of the two terms in the loss function (B.7). Indeed in some applications it might be preferable to propose a large number of matched vertices, at the risk of making many mistakes, or on the contrary to return a very partial estimator but with a large overlap for the few assigned vertices. This point was formalized under the name of one-sided partial recovery in [18]; we recall that in the sparse regime we are investigating some errors are unavoidable, in particular because of an extensive number of isolated vertices in the bicolored graph.

For a given "budget" of vertices to assign one should make a choice of which vertices to put in the estimator, and it would be advisable to select the ones which are the most likely to be correctly matched. As $\max_{i'} P_{i,i'}$ is the probability (over $\boldsymbol{\pi}_\star$, conditional on $\mathbf{A}, \mathbf{B}$) that $\widehat{\boldsymbol{\pi}}(i) = \boldsymbol{\pi}_\star(i)$, the information theoretically optimal choice (if one has access to the exact posterior distribution) is thus to consider a threshold $T \in [0, 1]$ and define the following estimator:

$$\widehat{\boldsymbol{\pi}}_T(i) = \begin{cases} \arg\max_{i'} P_{i,i'} & \text{if} \quad \max_{i'} P_{i,i'} > T \\ * & \text{if} \quad \max_{i'} P_{i,i'} \leq T \end{cases}, \tag{B.9}$$

that coincides with (B.8) when $T = 1/2$, thus justifying the notation. When $T = 0$ one recovers from this formula the full estimator $\widehat{\pi}$ of the main text, and for general values of T one has $\widehat{\boldsymbol{\pi}}_T(i) = \widehat{\boldsymbol{\pi}}(i)$ for all the assigned vertices. The role of T is thus to control the number of assigned vertices in the partial estimator, larger values of T corresponding to smaller values of $n(\widehat{\pi}_T)$, the estimator concentrating on the vertices which are the most likely to be correctly matched.

In practice we used this formula replacing P by its proxy $\widehat{P}$ computed from the message-passing algorithm according to Eq. (B.6), and obtained in this way the curves of Fig. B3. On the

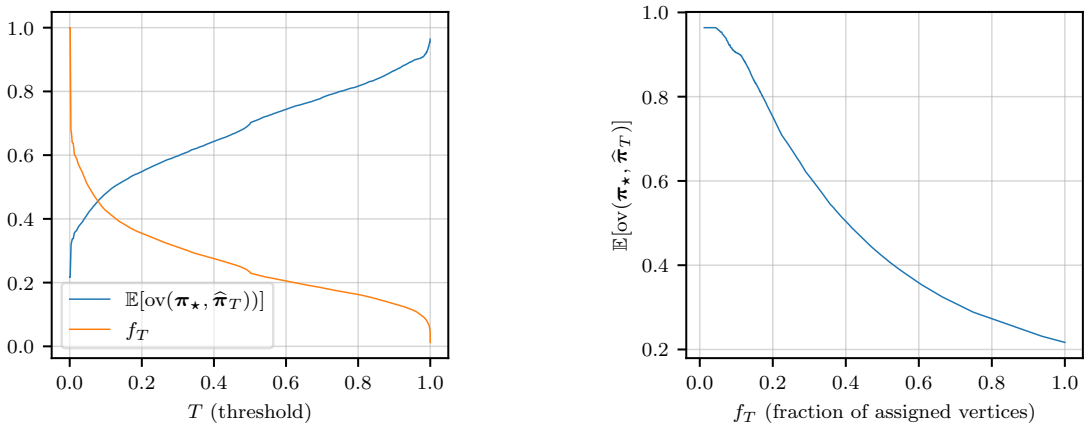

Figure B3. Performance of the threshold estimator of Eq. (B.9) for system size $n = 2048$, average degree $\lambda = 1.4$, correlation $s = 0.83$, depth $d = 10$, averages being taken over 25 independent realizations of $(\mathbf{A}, \mathbf{B})$. (Left) Average overlap and fraction of assigned vertices f_T as a function of the threshold: with larger T fewer vertices are included in the partial alignment but the overlap increases. (Right) Average overlap as function of the fraction of assigned vertices. The plot is obtained from the left panel: increasing T from zero to one corresponds to following the curve from the bottom right corner to the top left one.

left panel we present the fraction of assigned vertices $f_T = \mathbb{E}[n(\widehat{\boldsymbol{\pi}}_T)]/n$ and the corresponding overlap (computed only among the assigned vertices according to Eq. (B.5)) as a function of the threshold T. For small T most vertices are included in the partial alignment, however the overlap is small and close to that of the full estimator $\widehat{\boldsymbol{\pi}}$. When T increases the fraction of assigned vertices is reduced, but the partial overlap increases, showing that despite the approximation incurred when replacing P by $\widehat{P}$ the algorithm is indeed able to select the vertices that it manages to align correctly. In particular for $T \to 1$ only a few vertices are included, but they are almost all matched correctly (the overlap is close to 1). The left panel is a parametric representation of the same data, displaying the overlap as a function of the fraction of assigned vertices, hence the trade-off between the two opposite requirements of predicting a match for a large number of vertices, and doing this accurately.

Appendix B.3. A consistency check of the approximation

We will now present some numerical tests of the accuracy of the approximation made by replacing the posterior probabilities $P_{i,i'}$ by the expression $\widehat{P}_{i,i'}$ of Eq. (B.6). Both quantities depend implicitly on the observed pair of graphs A, B. Let us start considering the optimal estimator $\widehat{\boldsymbol{\pi}}_{\mathrm{opt}}$ defined from the exact posterior probabilities as $\widehat{\boldsymbol{\pi}}_{\mathrm{opt}}(i) = \arg\max_{i'} P_{i,i'}$. Its average overlap with the ground-truth can be written in two equivalent ways,

$$\begin{aligned}\mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}_{\mathrm{opt}}, \boldsymbol{\pi}_\star)] &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathbb{I}(\boldsymbol{\pi}_\star(i) = \arg\max_{i'} P_{i,i'})] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\sum_{j'} P_{i,j'}\mathbb{I}(j' = \arg\max_{i'} P_{i,i'})\right] \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\max_{i'} P_{i,i'}] \ ,\end{aligned} \tag{B.10}$$

where we used the fact, called *Nishimori condition* [39], that the law of $(\boldsymbol{\pi}_\star, (\mathbf{A}, \mathbf{B}))$ is the same as the law of $(\boldsymbol{\pi}', (\mathbf{A}, \mathbf{B}))$ if $\boldsymbol{\pi}'$ is drawn from the posterior law given $(\mathbf{A}, \mathbf{B})$. Note that in the last expression the ground-truth expression does not appear explicitly.

We turn now to the estimator $\widehat{\boldsymbol{\pi}}(i) = \arg\max_{i'} \widehat{P}_{i,i'}$ based on the approximation of the posterior probabilities through the message-passing algorithm, which is the only one we can use in practice, and define what we shall call the true average overlap,

$$\mathrm{ov} = \mathbb{E}[\mathrm{ov}(\widehat{\boldsymbol{\pi}}, \boldsymbol{\pi}_\star)] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\mathbb{I}(\boldsymbol{\pi}_\star(i) = \arg\max_{i'} \widehat{P}_{i,i'})] \ , \tag{B.11}$$

and the estimated average overlap,

$$\widehat{\mathrm{ov}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\max_{i'} \widehat{P}_{i,i'}] \ , \tag{B.12}$$

where in the last equation we do not use explicitly the knowledge of the ground-truth permutation, $\max_{i'} \widehat{P}_{i,i'}$ being the estimation by the algorithm itself of the probability that its prediction $\widehat{\boldsymbol{\pi}}(i)$ is correct. According to the discussion above we would have $\mathrm{ov} = \widehat{\mathrm{ov}}$ if the approximated probabilities $\widehat{P}_{i,i'}$ coincided with the exact ones $P_{i,i'}$.

We present in Fig. B4 the numerical results of the comparison of the two quantities ov and $\widehat{\mathrm{ov}}$; to account for the dependency of $\widehat{P}$ on the depth parameter d of the message-passing algorithm we plot both quantities as a function of d for fixed values of n, λ, s. One can see on the figure that they coincide for small d but differ from each other for larger values of d; as a matter of fact when d increases the estimated probability laws $\widehat{P}_{i,\cdot}$ become more and more concentrated on their mode $\widehat{\boldsymbol{\pi}}(i)$, i.e., close to $\widehat{P}_{i,i'} = \delta_{i',\widehat{\boldsymbol{\pi}}(i)}$. This means that the algorithm becomes overconfident about its predictions, its estimation of the probability that the predicted match is correct is larger than the true one. The plots on the bottom of Fig. B4 study the dependency of the separation point between the two curves on λ and n. Larger n leads to agreement up to higher values of d, while higher λ makes the two curves depart at smaller d. Let us finally underline that if $P = \hat{P}$ implies that $\mathrm{ov} = \widehat{\mathrm{ov}}$, the converse implication is far from being true: suppose for instance that $\widehat{P}_{i,i'} = 1/n$ for all i, i', i.e., that the estimation does not extract any information from the observations and that $\widehat{\boldsymbol{\pi}}(i)$ is a uniformly random vertex (the ties in the arg max being broken at random). Then from (B.11) and (B.12) it follows that $\mathrm{ov} = \widehat{\mathrm{ov}} = 1/n$. However in a hard phase where partial recovery is information theoretically possible this situation would occur with a non-trivial $P \neq \widehat{P}$.

Bibliography

[1] Conte D, Foggia P, Sansone C and Vento M 2004 Thirty years of graph matching in pattern recognition *International journal of pattern recognition and artificial intelligence* **18** 265–298

[2] Narayanan A and Shmatikov V 2008 Robust de-anonymization of large sparse datasets *2008 IEEE Symposium on Security and Privacy (sp 2008)* (IEEE) pp 111–125

[3] Pedarsani P and Grossglauser M 2011 On the privacy of anonymized networks *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* KDD '11 (New York, NY, USA: Association for Computing Machinery) p 1235–1243 ISBN 9781450308137

[4] Ramani A K and Marcotte E M 2003 Exploiting the co-evolution of interacting proteins to discover interaction specificity *Journal of molecular biology* **327** 273–284

[5] Berg J and Lässig M 2006 Cross-species analysis of biological networks by Bayesian alignment *Proceedings of the National Academy of Sciences* **103** 10967–10972

[6] Li Z, Zhang S, Wang Y, Zhang X S and Chen L 2007 Alignment of molecular networks by integer quadratic programming *Bioinformatics* **23** 1631–1639

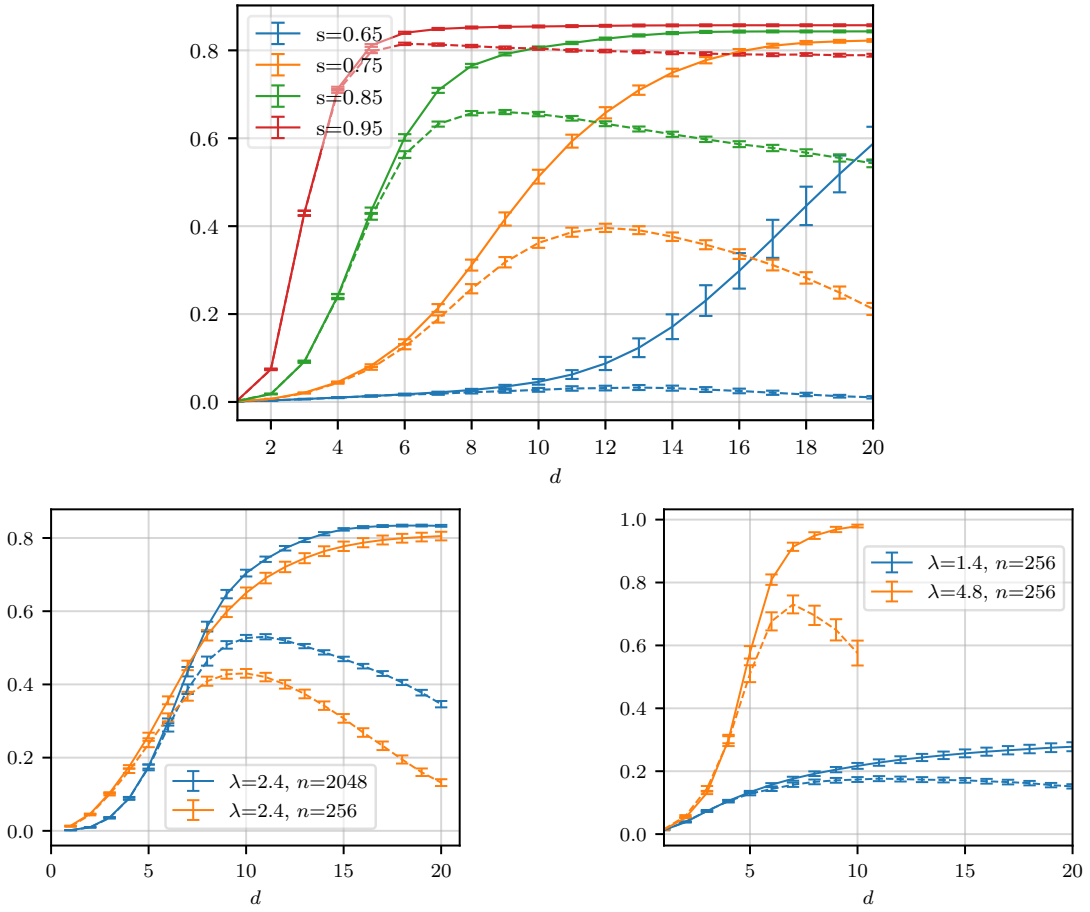

Figure B4. A comparison of the true average overlap (dashed line) of Eq. (B.11) and of the estimated overlap (solid line) of Eq. (B.12). (Top) $n = 2048$, $\lambda = 2.4$: for small d the estimated and true overlap agree, however increasing d the estimated overlap grows towards 1 showing that the algorithm is overconfident of its predictions. (Bottom left) when n is increased the value of d at which the true and estimated overlap ceases to agree also increases. (Bottom right) when λ is increased the value of d at which the two curves separates decreases, possibly because of stronger finite size effects for larger λ.

[7] Singh R, Xu J and Berger B 2008 Global alignment of multiple protein interaction networks with application to functional orthology detection *Proceedings of the National Academy of Sciences* **105** 12763–12768
[8] Nowak A, Villar S, Bandeira A S and Bruna J 2018 Revised note on learning quadratic assignment with graph neural networks *2018 IEEE Data Science Workshop (DSW)* (IEEE) pp 1–5
[9] Azizian W and Lelarge M 2020 Expressive power of invariant and equivariant graph neural networks *arXiv:2006.15646*
[10] Burkard R E, Çela E, Pardalos P M and Pitsoulis L S 1998 *The Quadratic Assignment Problem* (Boston, MA: Springer US) pp 1713–1809 ISBN 978-1-4613-0303-9
[11] Cullina D and Kiyavash N 2016 Improved achievability and converse bounds for erdos-renyi graph matching *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science* SIGMETRICS '16 (New York, NY, USA: Association for Computing Machinery) p 63–72 ISBN 9781450342667

[12] Ding J, Ma Z, Wu Y and Xu J 2021 Efficient random graph matching via degree profiles *Probability Theory and Related Fields* **179**(1) 29–115
[13] Fan Z, Mao C, Wu Y and Xu J 2020 Spectral graph matching and regularized quadratic relaxations: Algorithm and theory *Proceedings of the 37th International Conference on Machine Learning* (*Proceedings of Machine Learning Research* vol 119) ed III H D and Singh A (PMLR) pp 2985–2995
[14] Mao C, Rudelson M and Tikhomirov K 2021 Exact matching of random graphs with constant correlation *arXiv:2110.05000*
[15] Ganassali L and Massoulié L 2020 From tree matching to sparse graph alignment *Proceedings of Thirty Third Conference on Learning Theory* (*Proceedings of Machine Learning Research* vol 125) (PMLR) pp 1633–1665
[16] Wu Y, Xu J and Yu S H 2021 Settling the sharp reconstruction thresholds of random graph matching *arXiv:2102.00082*
[17] Ganassali L, Massoulié L and Lelarge M 2021 Impossibility of partial recovery in the graph alignment problem *Proceedings of Thirty Fourth Conference on Learning Theory* (*Proceedings of Machine Learning Research* vol 134) (PMLR) pp 2080–2102
[18] Ganassali L, Massoulié L and Lelarge M 2021 Correlation detection in trees for partial graph alignment *arXiv:2107.07623*
[19] Hall G and Massoulié L 2020 Partial Recovery in the Graph Alignment Problem *arxiv:2007.00533*
[20] Onaran E, Garg S and Erkip E 2016 Optimal de-anonymization in random graphs with community structure *2016 50th Asilomar Conference on Signals, Systems and Computers* pp 709–713
[21] Racz M Z and Sridhar A 2021 Correlated stochastic block models: Exact graph matching with applications to recovering communities *arxiv:2107.06767*
[22] Ganassali L, Lelarge M and Massoulié L 2019 Spectral Alignment of Correlated Gaussian matrices *arXiv:1912.00231*
[23] Ganassali L 2020 Sharp threshold for alignment of graph databases with Gaussian weights *arXiv:2010.16295*
[24] Bradde S, Braunstein A, Mahmoudi H, Tria F, Weigt M and Zecchina R 2010 Aligning graphs and finding substructures by a cavity approach *EPL (Europhysics Letters)* **89** 37009
[25] Bayati M, Gleich D F, Saberi A and Wang Y 2013 Message-passing algorithms for sparse network alignment *ACM Trans. Knowl. Discov. Data* **7**
[26] Aldous D J 2001 The $\zeta(2)$ limit in the random assignment problem *Random Structures & Algorithms* **18** 381–418
[27] Mézard M and Parisi G 2001 The bethe lattice spin glass revisited *Eur. Phys. J. B* **20** 217
[28] Decelle A, Krzakala F, Moore C and Zdeborová L 2011 Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications *Phys. Rev. E* **84**(6) 066106
[29] Moore C 2017 The computer science and physics of community detection: Landscapes, phase transitions, and hardness *Bulletin of EATCS* **1** preprint: arXiv:1702.00467
[30] Abbe E 2018 Community detection and stochastic block models: Recent developments *Journal of Machine Learning Research* **18** 1–86
[31] Mossel E, Neeman J and Sly A 2016 Belief propagation, robust reconstruction and optimal recovery of block models *Ann. Appl. Probab.* **26** 2211–2256
[32] Mao C, Wu Y, Xu J and Yu S H 2021 Testing network correlation efficiently via counting trees *arxiv:2110.11816*
[33] Otter R 1948 The number of trees *Annals of Mathematics* **49** 583–599
[34] Kim J H, Sudakov B and Vu V H 2002 On the asymmetry of random regular graphs and random graphs *Random Structures & Algorithms* **21** 216–224
[35] Yartseva L and Grossglauser M 2013 On the performance of percolation graph matching *Proceedings of the First ACM Conference on Online Social Networks* COSN '13 (New York, NY, USA: Association for Computing Machinery) p 119–130
[36] Lyzinski V, Fishkind D E and Priebe C E 2014 Seeded graph matching for correlated erdös-rényi graphs *J. Mach. Learn. Res.* **15** 3513–3540
[37] Mossel E and Xu J 2020 Seeded graph matching via large neighborhood statistics *Random Structures & Algorithms* **57** 570–611
[38] Yu L, Xu J and Lin X 2021 The power of d-hops in matching power-law graphs *arxiv:2102.12975*
[39] Zdeborová L and Krzakala F 2016 Statistical physics of inference: Thresholds and algorithms *Advances in Physics* **65** 453–552