From tree matching to sparse graph alignment

Luca Ganassali, Laurent Massoulié

HAL Id: hal-02941092
https://hal.archives-ouvertes.fr/hal-02941092

Submitted on 16 Sep 2020

From tree matching to sparse graph alignment

Luca Ganassali   LUCA.GANASSALI@INRIA.FR
INRIA, DI/ENS, PSL Research University, Paris, France.

Laurent Massoulié   LAURENT.MASSOULIE@INRIA.FR
MSR-Inria Joint Centre, INRIA, DI/ENS, PSL Research University, Paris, France.

Abstract

In this paper we consider alignment of sparse graphs, for which we introduce the ***Neighborhood Tree Matching Algorithm*** (NTMA). For correlated Erdős-Rényi random graphs, we prove that the algorithm returns – in polynomial time – a positive fraction of correctly matched vertices, and a vanishing fraction of mismatches. This result holds with average degree of the graphs in $O(1)$ and correlation parameter s that can be bounded away from 1, conditions under which random graph alignment is particularly challenging. As a byproduct of the analysis we introduce a matching metric between trees and characterize it for several models of correlated random trees. These results may be of independent interest, yielding for instance efficient tests for determining whether two random trees are correlated or independent.

Keywords: graph alignment, tree matching, Erdős-Rényi random graphs

Introduction

Graph alignment consists in finding an injective mapping (matching) $\mathcal{S} \subset V(G_1) \times V(G_2)$ between the vertex sets of two graphs G_1 and G_2 such that, for any two matched pairs (i, u), $(j, v) \in \mathcal{S}$, then occurrence of edge $\{i, j\}$ in G_1 tends to correspond to occurrence of edge $\{u, v\}$ in G_2. When this correspondence is exact for all pairs (i, u), (j, v) of matches, then the subgraphs of G_1, G_2 induced by their nodes appearing in $\mathcal{S}$ are isomorphic. In this sense, graph alignment is the search for approximate graph isomorphisms. It has many applications, among which: social network de-anonymization (Narayanan and Shmatikov (2008), Narayanan and Shmatikov (2009)), analysis of protein-protein interaction graphs (Kazemi et al. (2016), Feizi et al. (2019)), natural language processing (Bayati et al. (2013)), medical image processing (Lombaert et al. (2013)).

   A recent thread of research (Pedarsani and Grossglauser (2011), Dai et al. (2019b) Cullina and Kiyavash (2016), Ding et al. (2018), Cullina et al. (2019), Fan et al. (2019a), Fan et al. (2019b)) has investigated the fundamental limits to feasibility of graph alignment in the context of a natural generative model of correlated graphs, namely the correlated Erdős-Rényi random graph model $ERC(n, p, s)$. Specifically, it consists of two random graphs G_1 and G_2 on node set $[n] = \{1, \dots, n\}$ obtained as follows. First generate two aligned graphs G_1, G_2' with adjacency matrices A_1, A_2' such that for each node pair $\{i, j\}$, one has

$$
\mathbb{P}\left(A_1(i, j) = A_2'(i, j) = 1\right) = ps,
$$
$$
\mathbb{P}\left(A_1(i, j) = 1, A_2'(i, j) = 0\right) = \mathbb{P}\left(A_1(i, j) = 0, A_2'(i, j) = 1\right) = p(1 - s),
$$
$$
\mathbb{P}\left(A_1(i, j) = A_2'(i, j) = 0\right) = 1 - p(2 - s),
$$

and this independently over node pairs $\{i, j\}$. Graph G_2's adjacency matrix is then $A_2 = M_\sigma A_2' (M_\sigma)^\top$, where M_σ is the matrix associated to a permutation σ chosen uniformly at random from $\mathcal{S}_n$.

Researchers have strived to determine for which parameter values (p, s), assuming $n \gg 1$, one can recover the unknown permutation σ, and therefore the alignment G_2' of G_2 with G_1. As in other high-dimensional inference tasks such as community detection, one expects such goal to be either poly-time achievable, achievable though not in poly-time, or impossible to achieve. The corresponding regions of parameter space are usually referred to as the "easy", "hard" or "Information-theoretically (IT) impossible" phases for the problem considered.

Cullina and Kiyavash (2016) have shown that it is possible to recover σ if and only if $nps - \log(n) \underset{n \to \infty}{\longrightarrow} +\infty$, thereby characterizing the "IT-impossible" phase. Ding et al. (2018) have proposed a polynomial-time 'degree profile matching' algorithm, and proven it to recover σ under the conditions $np \geq \log^\alpha(n)$, $1 - s \leq \log^{-\beta}(n)$ for suitable constants $\alpha, \beta > 0$, thereby identifying a subset of the "easy" phase. More recently, Fan et al. (2019a), Fan et al. (2019b) have proposed a spectral method, and proven it to recover σ under the same conditions.

The result of Cullina and Kiyavash (2016) shows that there is no hope of recovering σ, or in other words, of perfectly re-aligning G_1 and G_2, in the case of sparse graphs, that is graphs with average degree np of order 1. Nevertheless, their result does not rule out the possibility of partially recovering the unknown permutation σ. For the applications mentioned earlier, it is at the same time natural to assume that the graphs involved are sparse, and potentially useful to recover only a fraction of the unknown matches $(i, \sigma(i))$.

Objectives and main result

This motivates the present work, whose goal is to show that ***partial alignment of sparse correlated graphs is feasible***, and to introduce a polynomial-time algorithm for producing such partial alignments. Our main result is the proposal of the so-called ***Neighborhood Tree Matching Algorithm*** (NTMA hereafter) together with the following

Theorem I *Consider the correlated Erdős-Rényi model $ERC(n, p, s)$, where $p = \lambda/n$. For some $\lambda_0 > 1$, for all $\lambda \in (1, \lambda_0]$, there exists $s^*(\lambda) < 1$ such that, provided $s \in (s^*(\lambda), 1]$, the NTMA returns a matching $\mathcal{S}$ verifying the following properties with high probability:*

$$|\mathcal{S} \cap \{(i, \sigma(i)), \ i \in [n]\}| = \Omega(n), \quad |\mathcal{S} \setminus \{(i, \sigma(i)), \ i \in [n]\}| = o(n). \tag{1}$$

In words, our algorithm returns a set of node alignments which contains a negligible fraction of mismatches, and $\Omega(n)$ good matches. Our result covers values of λ arbitrarily close to 1, and thus applies to very sparse graphs. For $\lambda < 1$, Erdős-Rényi graphs in our correlated model have connected components of size at most logarithmic in n, so that there is no hope of recovering a positive fraction of correct matches. This result can be interpreted as follows. For partial graph alignment of sparse Erdős-Rényi correlated random graphs, there is an "easy phase" that includes the parameter range $\{(\lambda, s) : \lambda \in (1, \lambda_0], \ s \in (s^*(\lambda), 1]\}$.

Paper organization

Description of the Neighborhood Tree Matching Algorithm and the proof strategy for establishing Theorem I are given in Section 2. Our algorithm relies essentially on a tree matching operation.

To pave the way for Section 2, we therefore introduce in Section 1 a notion of matching weight between trees that is key for our algorithm, and can be computed efficiently in a recursive manner. We further obtain probabilistic guarantees on the matching weights between random trees drawn according to some Galton-Watson branching processes. These are instrumental in the proof of Theorem I. However these may be of independent interest. Indeed we introduce in Section 1 a natural hypothesis testing problem on pairs of random trees, for which we obtain a successful test based on computation of tree matching weights.

Related work

Most relevant to the present work are the papers Pedarsani and Grossglauser (2011), Dai et al. (2019b) Cullina and Kiyavash (2016), Ding et al. (2018), Fan et al. (2019a), Fan et al. (2019b) already mentioned, which also focus on graph alignment in the context of the correlated Erdős-Rényi model. The main differences between the present paper and these is our focus on sparse random graph models, with average degree $\lambda = O(1)$, our treatment of correlation coefficients s bounded away below 1, and our aim of partial rather than full alignment. Article Cullina et al. (2019) addresses a notion of partial alignment stronger than ours, and hence requires conditions under which graphs are not sparse. Makarychev et al. (2014) show that graph alignment is NP-hard to solve, even approximately. This justifies the search for custom algorithms in a variety of scenarios. The main methods proposed are: ***Percolation methods*** based on some initial *seeds*, i.e. matched node pairs provided a priori (Kazemi et al. (2016)). We remark that the matchings returned by our algorithm could be used as seeds, and then processed e.g. using percolation matching to eventually obtain an improved matching. ***Spectral methods*** are considered in Feizi et al. (2019), Fan et al. (2019b); ***Degree profile matching*** is introduced in Ding et al. (2018); ***Quadratic programming*** approaches are proposed in Zaslavskiy et al. (2009). ***Message passing methods*** are introduced in Bayati et al. (2013). These are structurally similar to our neighborhood tree alignment approach, which is implemented in a recursive manner and can be seen as a message-passing method. Our algorithm is however different, and comes with novel theoretical guarantees. Graph alignment is a special case of the quadratic assignment problem, reviewed in Pardalos et al. (1993). Database alignment is an important variant of graph alignment, studied in Dai et al. (2019a).

Notations

For a graph G, denote by $V(G)$ its set of vertices, $E(G)$ its set non-oriented edges, and $\overrightarrow{E}(G) := \{(i,j), \{i,j\} \in E(G)\}$ its set of oriented edges. We use the notations $u \sim v$ if $\{u,v\} \in E(G)$ and $u \to v$ if $(u,v) \in \overrightarrow{E}(G)$. The usual graph distance in G will be denoted δ_G. For $v \in V(G)$, let $\mathcal{N}_G(v)$ denote the neighborhood of v in G, and $\deg_G(v)$ its degree. For $d \geq 1$ we also define $\mathcal{B}_G(v,d)$ the set of vertices at (graph) distance at most d from v, and $\mathcal{S}_G(v,d) := \mathcal{B}_G(v,d) \setminus \mathcal{B}_G(v,d-1)$ the set of vertices at distance d from v. For a rooted tree $\mathcal{T}$, we let $\rho(\mathcal{T})$ denote its root node. For any $i \in V(\mathcal{T}) \setminus \{\rho(\mathcal{T})\}$, we let $\pi_\mathcal{T}(i)$ denote the parent of node i in $\mathcal{T}$. For $d \geq 1$, we note $\mathcal{B}_d(\mathcal{T}) = \mathcal{B}_\mathcal{T}(\rho(\mathcal{T}),d)$ and $\mathcal{L}_d(\mathcal{T}) = \mathcal{S}_\mathcal{T}(\rho(\mathcal{T}),d)$. We omit the dependencies in G or $\mathcal{T}$ of these notations when there is no ambiguity.

1. Tree matching

In this section, we introduce the matching weight between rooted trees and the related matching rate. We then establish bounds on the matching rate for specific models of random trees. We also give an application to a hypothesis testing problem on the independence between two trees.

1.1. Matching weight of two rooted trees

Definition 1.1 *For any $d \geq 0$, let $\mathcal{A}_d$ denote the collection of rooted trees whose leaves are all of depth d. Given two rooted trees $\mathcal{T}$ and $\mathcal{T}'$, let $\mathcal{M}_d(\mathcal{T}, \mathcal{T}')$ denote the collection of trees $t \in \mathcal{A}_d$ such that there exist injective embeddings $f : V(t) \to V(\mathcal{T})$, $f' : V(t) \to V(\mathcal{T}')$ that preserve the rooted tree structure, i.e. such that*

$$f(\rho(t)) = \rho(\mathcal{T}), \qquad f'(\rho(t)) = \rho(\mathcal{T}'),$$
$$\forall i \in V(t) \setminus \{\rho(t)\}, \quad f(\pi_t(i)) = \pi_{\mathcal{T}}(f(i)), \quad f'(\pi_t(i)) = \pi_{\mathcal{T}'}(f'(i)).$$

*The **matching weight of $\mathcal{T}$ and $\mathcal{T}'$ at distance** d is then defined as:*

$$\mathcal{W}_d(\mathcal{T}, \mathcal{T}') := \sup_{t \in \mathcal{M}_d(\mathcal{T}, \mathcal{T}')} |\mathcal{L}_d(t)|, \tag{2}$$

i.e. the size, measured in number of leaves, of the largest tree in $\mathcal{M}_d(\mathcal{T}, \mathcal{T}')$.

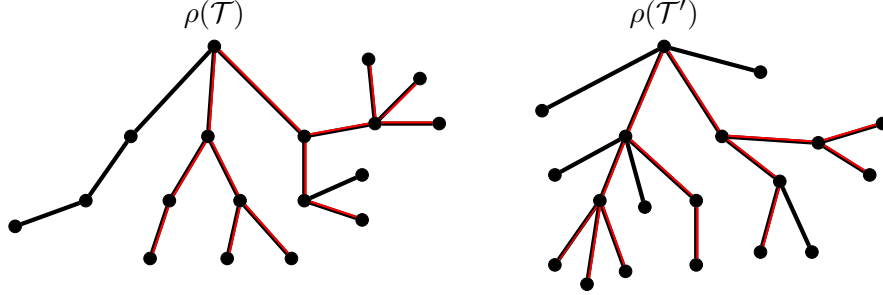Figure 1: Example of two trees $\mathcal{T}, \mathcal{T}'$ with $\mathcal{W}_3(\mathcal{T}, \mathcal{T}') = 7$, where an optimal $t \in \mathcal{A}_3$ is drawn in red.

1.2. Recursive computation of $\mathcal{W}_d$

We shall need the following notations and definitions. For a tree $\mathcal{T}$, for $i \in V(\mathcal{T})$, and $d \geq 0$, $\mathcal{T}_d^{(i)}$ is the sub-tree of $\mathcal{T}$ re-rooted at i, containing all vertices at distance less than d of i. For $i, j \in V(\mathcal{T})$ such that $j \to i$, $\mathcal{T}_d^{(i \leftarrow j)}$ denotes the sub-tree of $\mathcal{T}$ re-rooted at i, containing all vertices at distance less than d of i but where vertex j has been removed. By definition two vertices not connected by a path are at distance ∞. $\mathcal{T}_d^{(i \leftarrow j)}$ is thus the tree of depth at most d reached by oriented edge $j \to i$.
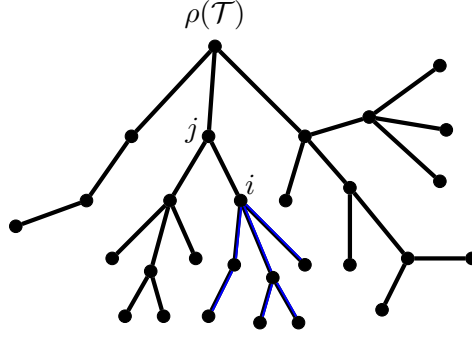
4

Figure 2: We here show an example of a tree $\mathcal{T}$ and its corresponding $\mathcal{T}_2^{(i \leftarrow j)}$ highlighted in blue.

Definition 1.2 *For a given pair of trees $\mathcal{T}$ and $\mathcal{T}'$, for all pair of vertices $(i, u) \in V(\mathcal{T}) \times V(\mathcal{T}')$, the **matching weight of** (i, u) **at depth** d is then defined as:*

$$\mathcal{W}_d(i, u) := \sup_{t \in \mathcal{M}_d\left(\mathcal{T}_d^{(i)}, \mathcal{T}_d'^{(u)}\right)} |\mathcal{L}_d(t)|. \tag{3}$$

*Moreover, for all pairs of vertices $(i, u), (j, v) \in V(\mathcal{T}) \times V(\mathcal{T}')$ such that $j \rightarrow i$ and $v \rightarrow u$, the **matching weight of edges** $j \rightarrow i$ **and** $v \rightarrow u$ **at distance** d is then defined as:*

$$\mathcal{W}_d(i \leftarrow j, u \leftarrow v) := \sup_{t \in \mathcal{M}_d\left(\mathcal{T}_d^{(i \leftarrow j)}, \mathcal{T}_d'^{(u \leftarrow v)}\right)} |\mathcal{L}_d(t)|. \tag{4}$$

Remark 1.1 *This definition is compatible with the first one in the context of tree matching: one has $\mathcal{W}_d(\rho(\mathcal{T}), \rho(\mathcal{T}')) = \mathcal{W}_d(\mathcal{T}, \mathcal{T}')$. Note that $\mathcal{W}_0(i, u) = 1$ and $\mathcal{W}_1(i, u) = \max(\deg(i), \deg(u))$. Similarly, $\mathcal{W}_0(i \leftarrow j, u \leftarrow v) = 1$ and $\mathcal{W}_1(i \leftarrow j, u \leftarrow v) = \max(\deg(i), \deg(u)) - 1$.*

Now fix $\mathcal{T}$ and $\mathcal{T}'$. From these definitions, for all $d \geq 1$, $(i, u), (j, v) \in V(\mathcal{T}) \times V(\mathcal{T}')$ such that $j \rightarrow i$ and $v \rightarrow u$, by doing a first step conditioning, we obtain the following recursion formulae:

$$\mathcal{W}_d(i \leftarrow j, u \leftarrow v) = \sup_{\mathfrak{m} \in \mathcal{M}(\mathcal{N}_\mathcal{T}(i) \backslash \{j\}, \mathcal{N}_{\mathcal{T}'}(u) \backslash \{v\})} \sum_{(k,w) \in \mathfrak{m}} \mathcal{W}_{d-1}(k \leftarrow i, w \leftarrow u), \tag{5}$$

where $\mathcal{M}(\mathcal{E}, \mathcal{F})$ is the set of all maximal injective (or one-to-one) mappings $\mathfrak{m} : \mathcal{E}_0 \subseteq \mathcal{E} \rightarrow \mathcal{F}$, where maximal means that they are not restrictions of another one-to-one mapping $\widetilde{\mathfrak{m}} : \mathcal{E}_1 \rightarrow \mathcal{F}$ with $\mathcal{E}_1$ such that $\mathcal{E} \supseteq \mathcal{E}_1 \supset \mathcal{E}_0$. In the same way we have

$$\mathcal{W}_d(i, u) = \sup_{\mathfrak{m} \in \mathcal{M}(\mathcal{N}_\mathcal{T}(i), \mathcal{N}_{\mathcal{T}'}(u))} \sum_{(k,w) \in \mathfrak{m}} \mathcal{W}_{d-1}(k \leftarrow i, w \leftarrow u). \tag{6}$$

Thus matching weights at depth d can be obtained by computing weights at depth $d-1$ and solving a linear assignment problem (LAP). Recursive formulae (5) and (6) yield simple recursive algorithms (see Algorithms 3 and 2 in A) to compute all matching weights at depth d.

5

Remark 1.2 *The complexity of computing all matching weights at depth d can be obtained as follows. We use dynamic programming and store the $\mathcal{W}_k(i \leftarrow j, u \leftarrow v)$ in a array of size the number of pairs (e, e') where e and e' are two oriented edges in $\mathcal{T}, \mathcal{T}'$ (that is, $4 \times |\mathcal{T}| \times |\mathcal{T}'|$). Each time we increase k, we solve one LAP for each pair (e, e'). The size of the small matrix on which the LAP is done does not exceed $d_{\max}$, the maximal degree in $\mathcal{T}$ and $\mathcal{T}'$. The Hungarian algorithm solves LAP with cubic time complexity. The time complexity is thus $O\left(d \times |\mathcal{T}| \times |\mathcal{T}'| \times d_{\max}^3\right)$. As $d_{\max} = O(\log n)$ with high probability, the total complexity is $O(dn^2 \log^3 n)$ where n bounds the number of nodes in $\mathcal{T}$ and $\mathcal{T}'$.*

However for small values of n, the recursive algorithms 3 and 2 are faster, although of complexity $O\left(d_{\max}^{2d}\right)$, which is not polynomial for $d = \Omega(\log n)$.

1.3. Matching rate of random trees

Definition 1.3 *Consider two random trees $\mathcal{T}, \mathcal{T}'$. Their **matching rate** is defined as*

$$\gamma(\mathcal{T}, \mathcal{T}') := \inf\left\{\gamma : \exists m, c, d_0 > 0, \forall x \geq 0, \forall d \geq d_0, \mathbb{P}\left(\mathcal{W}_d(\mathcal{T}, \mathcal{T}') \geq mx\gamma^d\right) \leq e^{-(x-c)_+}\right\}. \tag{7}$$

This quantity captures the geometric rate of growth of matching weights with depth d. A simpler alternative definition could be $\widetilde{\gamma}(\mathcal{T}, \mathcal{T}') := \inf\left\{\gamma : \mathbb{P}\left(\mathcal{W}_d(\mathcal{T}, \mathcal{T}') \geq \gamma^d\right) \underset{d\to\infty}{\longrightarrow} 0\right\}$. However definition (7) better suits our purpose.

Remark 1.3 *By definition, note that for any $\gamma > \gamma(\mathcal{T}, \mathcal{T}')$, $\mathbb{P}\left(\mathcal{W}_d(\mathcal{T}, \mathcal{T}') \geq \gamma^d\right)$ converges to 0 very fast, like $O\left(\exp\left(-c(\gamma)^d\right)\right)$ with $c(\gamma) > 1$, so that $\widetilde{\gamma}(\mathcal{T}, \mathcal{T}') \leq \gamma(\mathcal{T}, \mathcal{T}')$.*

1.4. Models of random trees

We now describe three models of random trees that are relevant to sparse graph alignment.

$GW(\lambda)$: We consider two independent Galton-Watson trees $\mathcal{T}$ and $\mathcal{T}'$ with offspring distribution $\text{Poi}(\lambda)$, $\lambda > 0$. We denote $(\mathcal{T}, \mathcal{T}') \sim GW(\lambda)$.

$GW(\lambda, s, \delta)$: For $\delta \geq 1$, consider a labeled tree $\mathcal{T}$ rooted at ρ and a tree $\mathcal{T}'$ rooted at ρ'. ρ' is also a node of $\mathcal{T}$, at distance δ from its root ρ. The two trees are generated as follows. First, nodes in $\mathcal{T}$ on the path from ρ to the parent of ρ' in $\mathcal{T}$ have, besides their child leading to ρ', $\text{Poi}(\lambda)$ children in $\mathcal{T}$, themselves having offspring in $\mathcal{T}$ given by independent Galton-Watson trees with offspring $\text{Poi}(\lambda)$. Then, the intersection between $\mathcal{T}$ and $\mathcal{T}'$ is a Galton-Watson tree with offspring $\text{Poi}(\lambda s)$, with $\lambda > 0$ and $s \in [0, 1]$. Then, to each node in the intersection tree, we attach children in $\mathcal{T} \setminus \mathcal{T}'$ and children in $\mathcal{T}' \setminus \mathcal{T}$, each number being independent $\text{Poi}(\lambda(1 - s))$ variables. These children in turn have offspring in the corresponding tree given by independent Galton-Watson trees with offspring $\text{Poi}(\lambda)$. See figure 3 for an illustration. We denote $(\mathcal{T}, \mathcal{T}') \sim GW(\lambda, s, \delta)$.

$GW(\lambda, s)$: It is the previous model with $\delta = 0$, so that the two correlated trees $\mathcal{T}$ and $\mathcal{T}'$ have same root ρ. We denote $(\mathcal{T}, \mathcal{T}') \sim GW(\lambda, s)$.
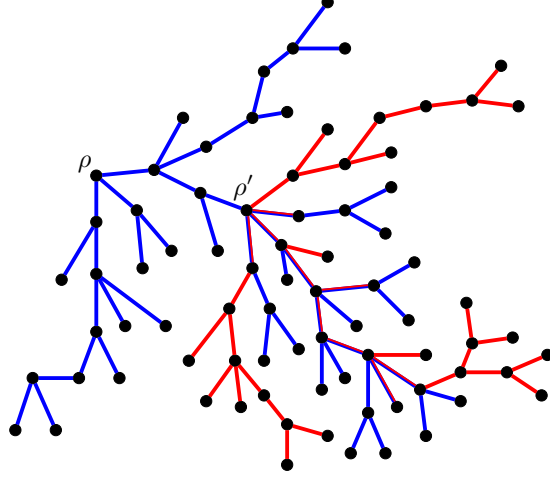
Figure 3: Random trees $\mathcal{T}$ (blue) and $\mathcal{T}'$ (red) from model $GW(\lambda, s, \delta)$ with $\delta = 3$.

We now turn to the analysis of matching rates for these models.

Proposition 1.1 *Let $\lambda > 1$ and $s \in [0, 1]$ such that $\lambda s > 1$. For $(\mathcal{T}, \mathcal{T}') \sim GW(\lambda, s)$, letting $\gamma(\lambda, s) := \gamma(\mathcal{T}, \mathcal{T}')$, we have:*

$$\gamma(\lambda, s) \geq \lambda s.$$

Proof Let $\mathcal{T}_\cap$ be the intersection tree between $\mathcal{T}$ and $\mathcal{T}'$. Branching process theory implies that $(\lambda s)^{-d} \big| \mathcal{L}_d(\mathcal{T}_\cap) \big|$ converges almost surely to a random variable Z as $d \to \infty$, such that $\mathbb{P}(Z > 0) = 1 - p$, with p the extinction probability of the branching tree $\mathcal{T}_\cap$. Since $p < 1$ when $\lambda s > 1$, and for every small enough $\varepsilon > 0$, $\lim_{d \to \infty} \mathbb{P}\left(\mathcal{W}_d(\mathcal{T}, \mathcal{T}') \geq (\lambda s(1 - \varepsilon))^d\right) \geq 1 - p$, the result follows. ∎

1.5. Matching rate of independent Galton-Watson trees

Theorem 1.1 *Let $\mathcal{T}$, $\mathcal{T}'$ be two independent Galton-Watson trees from the model $GW(\lambda)$. Let $\gamma(\lambda) := \gamma(\mathcal{T}, \mathcal{T}')$. There exists $\lambda_0 > 1$ such that for all $\lambda \in (1, \lambda_0]$, we have*

$$\gamma(\lambda) < \lambda. \tag{8}$$

Evaluations of $\gamma(\lambda)$ by simulations, confirming the Theorem, are provided in Appendix B.1.

Proof outline The full proof of Theorem 1.1 is detailed in the appendix (C.1), but we here give the key steps.

We introduce some notations. First, for a tree t, let $r_d(t)$ denote the tree obtained by suppressing nodes at depths greater than d, and then iteratively pruning leaves of depth strictly less than d. When computing $\mathcal{W}_d(t, t')$, the only informative sub-trees are precisely $r_d(t)$ and in $r_d(t')$, one of these being empty if t or t' doesn't survive up to depth d. In the rest of the paper, we define $\mathcal{T}_d$ the random variable $r_d(\mathcal{T})$ where $\mathcal{T}$ is conditioned to survive up to depth d.

7

Consider $(\mathcal{T}, \mathcal{T}') \sim GW(\lambda)$. We let $\mathcal{E}_d$ (respectively, $\mathcal{E}_d'$) denote the event that tree $\mathcal{T}$ (respectively, $\mathcal{T}'$) becomes extinct before d generations, i.e. $\mathcal{L}_d(\mathcal{T}) = \varnothing$ (respectively, $\mathcal{L}_d(\mathcal{T}') = \varnothing$). We let $p_d = \mathbb{P}(\mathcal{E}_d) = \mathbb{P}(\mathcal{E}_d')$. It is well known that it satisfies the recursion

$$p_0 = 0, \ p_d = e^{-\lambda(1 - p_{d-1})}.$$

We now establish the following lemma on the structure of $\mathcal{T}_d$:

Lemma 1.1 *For any $\lambda > 1$, $\mathcal{T}_d$ can be constructed by first sampling the number of children D of the root $\rho(\mathcal{T})$ according to distribution*

$$\mathbb{P}(D = k) = \mathbf{1}_{k>0} \frac{\mathbb{P}(\mathrm{Poi}(\lambda(1 - p_{d-1})) = k)}{\mathbb{P}(\mathrm{Poi}(\lambda(1 - p_{d-1})) > 0)} =: q_{d,k},$$

and then attaching D independent copies of $\mathcal{T}_{d-1}$ to the D children of $\rho(\mathcal{T})$.

Assume $\varepsilon = \lambda - 1$ to be small enough . Fix $r \in (0, 1)$, let $\gamma = 1 + r\varepsilon$. We first show using exponential moments that there exist $m, c > 0$ and $d_0 > 0$ such that for all $x > 0$

$$\mathbb{P}\left(\mathcal{W}_{d_0}\left(\mathcal{T}_{d_0}, \mathcal{T}_{d_0}'\right) \geq mx\right) \leq e^{-x+c}.$$

Then we define the random variables

$$X_d := \gamma^{-(d-d_0)} m^{-1} \mathcal{W}_d\left(\mathcal{T}_d, \mathcal{T}_d'\right).$$

Then, considering the number D of children of the root in $\mathcal{T}_d$ (resp. D' in $\mathcal{T}_d'$), using the previous lemma, one can establish, for all $x > 0$, a recursive formula of the following form

$$\mathbb{P}\left(X_d \geq x\right) \leq \sum_{k,\ell \geq 1} q_{d,k} q_{d,\ell} \mathbb{P}\left(\exists \mathfrak{m} \in \mathcal{M}\left([k], [\ell]\right), \sum_{(i,u) \in \mathfrak{m}} X_{d-1,i,u} \geq \gamma x\right),$$

where the $X_{d-1,i,u}$ are i.i.d. copies of X_{d-1}. The union bound yields

$$\mathbb{P}\left(X_d \geq x\right) \leq \sum_{k,\ell \geq 1} q_{d,k} q_{d,\ell} \min\left(1, (k \vee \ell)^{\underline{k \wedge \ell}} \times \mathbb{P}\left(\sum_{i=1}^{k \wedge \ell} X_{d-1,i,u} \geq \gamma x\right)\right),$$

where $m^{\underline{p}} := m(m-1)\dots(m-p+1) = \frac{m!}{(m-p)!}$. This inequality enables, with a few more technical steps (see C.1), to propagate recursively the inequality

$$\mathbb{P}\left(X_d \geq x\right) \leq e^{-(x-c)_+}.$$

IMPLICATIONS FOR A HYPOTHESIS TESTING PROBLEM

Let a pair of trees $(\mathcal{T}, \mathcal{T}')$ be distributed according to $GW(\lambda)$ under the null hypothesis $\mathcal{H}_0$, and according to $GW(\lambda, s)$ under the alternative hypothesis $\mathcal{H}_1$. They are thus independent under $\mathcal{H}_0$, and correlated under $\mathcal{H}_1$. Consider the following test:

$$\text{Decide } \mathcal{H}_0 \text{ if } \mathcal{W}_d(\mathcal{T}, \mathcal{T}') < \gamma^d, \ \mathcal{H}_1 \text{ otherwise.}$$

Assume that $\gamma(\lambda) < \gamma < \lambda s$. Then in view of Remark 1.3 and Theorem 1.1 one has for some $c(\gamma) > 1$:

$$\mathbb{P}\left(\text{decide } \mathcal{H}_1 \big| \mathcal{H}_0\right) = O\left(e^{-c(\gamma)^d}\right),$$

thus a super-exponential decay of the probability of false positive (first type error). Conversely, in view of Proposition 1.1, noting $\mathcal{T}_\cap$ the intersection tree under $\mathcal{H}_1$, one has

$$\mathbb{P}\left(\text{decide } \mathcal{H}_0 \big| \mathcal{H}_1, \text{non-extinction of } \mathcal{T}_\cap\right) = o_d(1).$$

The false negative probability of this test thus also goes to zero, provided the intersection tree survives. As we will see in the next section, this hypothesis testing problem on a pair of random trees is related to our original graph alignment problem much as the so-called tree reconstruction problem, reviewed in Mossel (2001), is related to community detection in sparse random graphs (see e.g. Bordenave et al. (2015)).

1.6. Matching rate of intersecting trees

Theorem 1.2 *Let $(\mathcal{T}, \mathcal{T}') \sim GW(\lambda, s, \delta)$ with $\delta \geq 1$ and $s \in [0, 1]$. Let $\gamma(\lambda, s, \delta) := \gamma(\mathcal{T}, \mathcal{T}')$. There exists $\lambda_0 > 1$ such that for all $\lambda \in (1, \lambda_0]$ we have*

$$\sup_{\delta \geq 1} \gamma(\lambda, s, \delta) < \lambda. \tag{9}$$

Evaluations of $\gamma(\lambda, s, \delta)$ by simulations, confirming the Theorem, are provided in Appendix B.1.

Proof outline   The full proof of Theorem 1.2 is detailed in the appendix (C.2), but we here give the key steps. The proof will again be by induction on d, the initial step being established with the same argument as in the proof of Theorem 1.1. $\varepsilon = \lambda - 1$ is assumed to be small enough. We fix $r \in (0, 1)$, and we let $\gamma = 1 + r\varepsilon'$. We now work with the random variables

$$X'_d := \gamma^{-(d-d_0)} m^{-1} \mathcal{W}_d\left(\mathcal{T}_d, \mathcal{T}'_d\right),$$

conditionally on the event that the path from ρ to ρ' survives down to depth d in $\mathcal{T}$. Then, considering D the number of children of ρ in $\mathcal{T}_d$, D' the number of children of ρ' in $\mathcal{T}'_d$ that are in the intersection tree $\mathcal{T}_d \cap \mathcal{T}'_d$, and D'' the number of children of ρ' in $\mathcal{T}'_d \setminus \mathcal{T}_d$, we establish for all $x > 0$ a recursive formula of the following form

$$\mathbb{P}\left(X'_d \geq x\right) \leq \sum_{k, \ell \geq 1} \mathbb{P}\left(D' + D'' = k, D = \ell\right) \min\left(1, (k \vee \ell)^{k \wedge \ell} \mathbb{P}\left(X'_{d-1} + \sum_{i=1}^{k \wedge \ell - 1} X_{d-1, i, u} \geq \gamma x\right)\right),$$

where the $X_{d-1, i, u}$ are i.i.d. copies of X_{d-1} as defined in the proof of Theorem 1.1. Again, with a few more technical steps (see C.2), we are able to propagate recursively the inequality

$$\mathbb{P}\left(X'_d \geq x\right) \leq e^{-(x-c)_+}.$$

2. Sparse graph alignment

We now describe our main algorithm and its theoretical guarantees. For simplicity we assume that the underlying permutation σ is the identity.

2.1. Neighborhood Tree Matching Algorithm (NTMA), main result

The main intuition for the NTMA algorithm is as follows. In order to distinguish matched pairs of nodes (i, u), we consider their neighborhoods at a certain depth d, that are close to Galton-Watson trees. In the case where the two vertices are actual matches, the largest common subtree measured in terms of children at depth (exactly) d is w.h.p. of size $\geq (\lambda s)^d$. However, when the two nodes i and u are sufficiently distant, previous study of matching rates shows that the growth rate of largest common subtree will be $< \lambda s$. The natural idea is thus to apply the test comparing $\mathcal{W}_d(i, u)$ to γ^d for some well-chosen γ to decide whether i is matched to u.

But as the reader may have noticed, testing $\mathcal{W}_d(i, u) > \gamma^d$ is not enough, because two-hop neighbors would dramatically increase the number of incorrectly matched pairs, making the performance collapse. To fix this, we use the **dangling trees trick**: instead of just looking at their neighborhoods, we look for the downstream trees from two distinct neighbors $j \neq j'$ of i, and $v \neq v'$ of u. The trick is now to compare both $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u)$ and $\mathcal{W}_{d-1}(j' \leftarrow i, v' \leftarrow u)$ to γ^{d-1}. This way, even if $i \neq u$ and i and u are close by, the pairs of rooted trees that can be considered will lead to one of the four cases considered and illustrated on Figure 10, that are settled in the proof of Theorem 2.2.

Our algorithm is as follows, where matching tree weights $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u)$ are defined in (4):

Algorithm 1 Neighborhood Tree Matching Algorithm for sparse graph alignment
---
Input: Two graphs G_1 and G_2 of size n, average degree λ, depth d, parameter γ.
Output: A set of pairs $\mathcal{S} \subset V(G_1) \times V(G_2)$.
$\mathcal{S} \leftarrow \varnothing$
for $(i, u) \in V(G_1) \times V(G_2)$ **do**
    if $\mathcal{B}_{G_1}(i, d)$ and $\mathcal{B}_{G_2}(u, d)$ contain no cycle, and $\exists j \neq j' \in \mathcal{N}_{G_1}(i), \exists v \neq v' \in \mathcal{N}_{G_2}(u)$ such that $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u) > \gamma^{d-1}$ and $\mathcal{W}_{d-1}(j' \leftarrow i, v' \leftarrow u) > \gamma^{d-1}$ **then**
        | $\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, u)\}$
    end
end
return $\mathcal{S}$

Remark 2.1 *For $d = \lfloor c \log n \rfloor$, in view of Remark 1.2, with high probability the complexity of NTMA is*

$$O\left(|V(G_1)| |V(G_2)| (\log n)^2 n^{2c \log \lambda} d_{\max}^2\right) + O\left(|E(G_1)| |E(G_2)| (\log n) d_{\max}^3\right),$$

where $d_{\max}$ is the maximum degree in G_1 and G_2. In the context of Theorems 2.1 and 2.2 the complexity is then $O\left((\log n)^4 n^{5/2}\right)$.

The two results to follow will readily imply Theorem I.

Theorem 2.1 *Let $(G_1, G_2) \sim ERC(n, \lambda/n, s)$ be two $s-$correlated Erdős-Rényi graphs such that $\lambda s > 1$. Let $d = \lfloor c \log n \rfloor$ with $c \log(\lambda(2 - s)) < 1/2$. Then for $\gamma \in (1, \lambda s)$, with high probability,*

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{(i,i) \in \mathcal{S}\}} = \Omega(1). \tag{10}$$

In other words, a non vanishing fraction of nodes is correctly recovered by NTMA (1).

Theorem 2.2 *Let $(G_1, G_2) \sim ERC(n, \lambda/n, s)$ be two $s-$correlated Erdős-Rényi graphs. Assume that $\gamma_0(\lambda) := \max\left(\gamma(\lambda), \sup_{\delta \geq 1} \gamma(\lambda, s, \delta)\right) < \lambda s$, and that $d = \lfloor c \log n \rfloor$ with $c \log \lambda < 1/4$. Then for $\gamma \in (\gamma_0(\lambda), \lambda s)$, with high probability,*

$$\mathrm{err}(n) := \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{\exists u \neq i, (i,u) \in \mathcal{S}\}} = o(1), \tag{11}$$

i.e. only at most a vanishing fraction of nodes are incorrectly matched by NTMA (1).

Remark 2.2 *The set $\mathcal{S}$ returned by the NTMA is not necessarily a matching. Let $\mathcal{S}'$ be obtained by removing all pairs (i, u) of $\mathcal{S}$ such that i or u appears at least twice. Theorems 2.1 and 2.2 guarantee that $\mathcal{S}'$ still contains a non-vanishing number of correct matches and a vanishing number of incorrect matches. Theorem I easily follows. Simulations of NTMA–2, a simple variant of of NTMA, are reported in Appendix B.2. These confirm our theory, as the algorithm returns many good matches and few mismatches.*

2.2. Proof strategy

We start by stating Lemmas, adapted from Massoulié (2013) and Bordenave et al. (2015) and proven in Appendix D, that are instrumental in the proofs of Theorems 2.1 and 2.2.

Lemma 2.1 (Control of the sizes of the neighborhoods) *Let $G \sim ER(n, \lambda/n)$, $d = \lfloor c \log n \rfloor$ with $c \log \lambda < 1$. For all $\gamma > 0$, there is a constant $C = C(\gamma) > 0$ such that with probability $1 - O\left(n^{-\gamma}\right)$, for all $i \in [n]$, $t \in [d]$:*

$$|\mathcal{S}_G(i, t)| \leq C(\log n)\lambda^t. \tag{12}$$

Lemma 2.2 (Cycles in the neighborhoods in an ER graph) *Let $G \sim ER(n, \lambda/n)$, $d = \lfloor c \log n \rfloor$ with $c \log \lambda < 1/2$. There exists $\varepsilon > 0$ such that for any vertex $i \in [n]$, one has*

$$\mathbb{P}\left(\mathcal{B}_G(i, d) \text{ contains a cycle}\right) = O\left(n^{-\varepsilon}\right). \tag{13}$$

Lemma 2.3 (Two logarithmic neighborhoods are typically size-independent) *Let $G \sim ER(n, \lambda/n)$ with $\lambda > 1$, $d = \lfloor c \log n \rfloor$ with $c \log \lambda < 1/2$. Then there exists $\varepsilon > 0$ such that for any fixed nodes $i \neq j$, the variation distance between the joint law of the neighborhoods $\mathcal{L}\left((\mathcal{S}_G(i, t), \mathcal{S}_G(j, t))_{t \leq d}\right)$ and the product law $\mathcal{L}\left((\mathcal{S}_G(i, t))_{t \leq d}\right) \otimes \mathcal{L}\left((\mathcal{S}_G(j, t))_{t \leq d}\right)$ tends to 0 as $O\left(n^{-\varepsilon}\right)$ for some $\varepsilon > 0$ when $n \to \infty$.*

Lemma 2.4 (Coupling the $|\mathcal{S}_G(i, t)|$ with a Galton-Watson process) *Let $G \sim ER(n, \lambda/n)$, $d = \lfloor c \log n \rfloor$ with $c \log \lambda < 1/2$. For a fixed $i \in [n]$, the variation distance between the law of $(|\mathcal{S}_G(i, t)|)_{t \leq d}$ and the law of $(Z_t)_{t \leq d}$ where $(Z_t)_t$ is a Galton-Watson process of offspring distribution $\mathrm{Poi}(\lambda)$ tends to 0 as $O\left(n^{-\varepsilon}\right)$ when $n \to \infty$.*

PROOF OF THEOREMS 2.1 AND 2.2

Proof [Proof of Theorem 2.1] Define the joint graph $G_\cup = G_1 \cup G_2$. For $i \in [n]$, let M_i denote the event that the algorithm matches i in G_1 with i in G_2, i.e. on which $\mathcal{B}_{G_1}(i,d)$ and $\mathcal{B}_{G_2}(i,d)$ contain no cycle, and $\exists j \neq j' \in \mathcal{N}_{G_1}(i), \exists v \neq v' \in \mathcal{N}_{G_2}(i)$ such that $\mathcal{W}_{d-1}((j,v),(i,u)) > \gamma^{d-1}$ and $\mathcal{W}_{d-1}((j',v'),(i,u)) > \gamma^{d-1}$. Denote by $C_{\cup,i,d}$ the event that there is no cycle in $\mathcal{B}_{G_\cup}(i,d)$.

Arguing as in the proof of Lemma 2.4, the two neighborhoods $\mathcal{B}_{G_1}(i,d)$ and $\mathcal{B}_{G_2}(i,d)$ can be coupled with trees distributed as $GW(\lambda,s)$ of Section 1. However, we will instead consider the intersection graph $G_\cap = G_1 \cap G_2$. Obviously, $G_\cap \sim ER(n, \lambda s/n)$. By Lemma 2.4, the random variables $|\mathcal{S}_{G_\cap}(i,t)|$ can be coupled with a Galton-Watson process with offspring distribution $\text{Poi}(\lambda s)$ up to depth $t = d$. Let P_i denote the event that this coupling succeeds. Since $\lambda s > 1$, there is a probability $2\alpha > 0$ that this process survives up to depth d and that the first generation has at least two children. Note S this event. On event S, the matching given by the identity on the intersection tree implies the existence of two neighbors $j \neq j' \in \mathcal{N}_{G_1}(i)$ and $v \neq v' \in \mathcal{N}_{G_2}(i)$ such that with high probability $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u) > \gamma^{d-1}$ and $\mathcal{W}_{d-1}(j' \leftarrow i, v' \leftarrow u) > \gamma^{d-1}$, by standard martingale arguments, as in Proposition 1.1. This gives the lower bound for $\mathbb{P}(M_i)$:

$$\mathbb{P}(M_i) \geq \mathbb{P}(C_{\cup,i,d} \cap P_i \cap S) \geq 2\alpha - o(1) > \alpha > 0.$$

It is easy to see that $G_\cup \sim ER(n, \lambda(2-s)/n)$. For $i \neq j \in [n]$, define $I_{i,j}$ the event on which the two neighborhoods of i and j in $G_\cup$ coincide with their independent couplings up to depth d. By lemma 2.3, $\mathbb{P}(I_{i,j}) = 1 - o(1)$. Then for $0 < \varepsilon < \alpha$ Markov's inequality yields

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\{(i,i)\in\mathcal{S}\}} < \alpha - \varepsilon\right) \leq \mathbb{P}\left(\sum_{i=1}^{n}(\mathbb{P}(M_i) - \mathbf{1}_{M_i}) > \varepsilon n\right) \tag{14}$$

$$\leq \frac{1}{n^2\varepsilon^2}\left(n\text{Var}(\mathbf{1}_{M_1}) + n(n-1)\text{Cov}(\mathbf{1}_{M_1}, \mathbf{1}_{M_2})\right) \tag{15}$$

$$\leq \frac{\text{Var}(\mathbf{1}_{M_1})}{n\varepsilon^2} + \frac{1 - \mathbb{P}(I_{1,2})}{\varepsilon^2} \to 0. \tag{16}$$

∎

Proof strategy for Theorem 2.2. Consider two distinct nodes i and u. We place ourselves on the event of high probability that $\mathcal{B}_{G_\cup}(i,2d)$ has no cycle. On this event, the two neighborhoods $\mathcal{B}_{G_1}(i,d)$ and $\mathcal{B}_{G_2}(u,d)$ can be coupled with two trees rooted at i, u respectively. We then distinguish several cases that are shown on Figure 10, that require detailed analysis, and which all show that for i fixed, one has

$$\mathbb{P}(\exists u \neq i, \ (i,u) \in \mathcal{S}) = o(1).$$

The full proof is deferred to Appendix D.5.

3. Conclusion

We have introduced NTMA, an algorithm we proved to succeed at partial alignment of sparse correlated random graphs. While our Theorem applies to a limited range of average degrees λ, we conjecture that rates $\gamma(\lambda)$ and $\gamma(\lambda, s, \delta)$ are strictly less than λ for all $\lambda > 1$ and $s < 1$ and thus NTMA in fact succeeds for a much broader parameter range. This will be the object of future work.

Acknowledgments

References

A. D. Barbour and Louis H. Y. Chen. *An Introduction to Stein's Method.* co-published with Singapore University, 2005. doi: 10.1142/5792. URL https://www.worldscientific.com/doi/abs/10.1142/5792.

Mohsen Bayati, David F. Gleich, Amin Saberi, and Ying Wang. Message-passing algorithms for sparse network alignment. *TKDD*, 7(1):3:1–3:31, 2013. doi: 10.1145/2435209.2435212. URL https://doi.org/10.1145/2435209.2435212.

Charles Bordenave, Marc Lelarge, and Laurent Massoulié. Non-backtracking spectrum of random graphs: Community detection and non-regular ramanujan graphs. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1347–1357, 2015. doi: 10.1109/FOCS.2015.86. URL https://doi.org/10.1109/FOCS.2015.86.

Daniel Cullina and Negar Kiyavash. Improved achievability and converse bounds for Erdős-Rényi graph matching. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science, Antibes Juan-Les-Pins, France, June 14-18, 2016*, pages 63–72, 2016. doi: 10.1145/2896377.2901460. URL https://doi.org/10.1145/2896377.2901460.

Daniel Cullina, Negar Kiyavash, Prateek Mittal, and H. Vincent Poor. Partial recovery of Erdős-Rényi graph alignment via k-core alignment. *POMACS*, 3(3):54:1–54:21, 2019. doi: 10.1145/3366702. URL https://doi.org/10.1145/3366702.

Osman Emre Dai, Daniel Cullina, and Negar Kiyavash. Database alignment with gaussian features. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 3225–3233, 2019a. URL http://proceedings.mlr.press/v89/dai19b.html.

Osman Emre Dai, Daniel Cullina, Negar Kiyavash, and Matthias Grossglauser. Analysis of a canonical labeling algorithm for the alignment of correlated erdős-rényi graphs. In *Abstracts of the 2019 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems, Phoenix, AZ, USA, June 24-28, 2019*, pages 97–98, 2019b. doi: 10.1145/3309697.3331505. URL https://doi.org/10.1145/3309697.3331505.

Jian Ding, Zongming Ma, Yihong Wu, and Jiaming Xu. Efficient random graph matching via degree profiles. *CoRR*, abs/1811.07821, 2018. URL http://arxiv.org/abs/1811.07821.

Zhou Fan, Cheng Mao, Yihong Wu, and Jiaming Xu. Spectral graph matching and regularized quadratic relaxations I: the gaussian model. *CoRR*, abs/1907.08880, 2019a. URL http://arxiv.org/abs/1907.08880.

Zhou Fan, Cheng Mao, Yihong Wu, and Jiaming Xu. Spectral graph matching and regularized quadratic relaxations II: erdős-rényi graphs and universality. *CoRR*, abs/1907.08883, 2019b. URL http://arxiv.org/abs/1907.08883.

Soheil Feizi, Gerald Quon, Mariana Mendoza, Muriel Medard, Manolis Kellis, and Ali Jadbabaie. Spectral alignment of graphs. *IEEE Transactions on Network Science and Engineering*, 2019.

Ehsan Kazemi, Seyed Hamed Hassani, Matthias Grossglauser, and Hassan Pezeshgi Modarres. PROPER: global protein interaction network alignment through percolation matching. *BMC Bioinformatics*, 17:527:1–527:16, 2016. doi: 10.1186/s12859-016-1395-9. URL https://doi.org/10.1186/s12859-016-1395-9.

Achim Klenke and Lutz Mattner. Stochastic ordering of classical discrete distributions, 2009.

Herve Lombaert, Jon Sporring, and Kaleem Siddiqi. Diffeomorphic spectral matching of cortical surfaces. In *Information Processing in Medical Imaging - 23rd International Conference, IPMI 2013, Asilomar, CA, USA, June 28-July 3, 2013. Proceedings*, pages 376–389, 2013. doi: 10.1007/978-3-642-38868-2_32. URL https://doi.org/10.1007/978-3-642-38868-2_32.

Konstantin Makarychev, Rajsekar Manokaran, and Maxim Sviridenko. Maximum quadratic assignment problem: Reduction from maximum label cover and lp-based approximation algorithm. *ACM Trans. Algorithms*, 10(4):18:1–18:18, 2014. doi: 10.1145/2629672. URL https://doi.org/10.1145/2629672.

Laurent Massoulié. Community detection thresholds and the weak Ramanujan property. *arXiv e-prints*, art. arXiv:1311.3085, Nov 2013.

Elchanan Mossel. Survey: Information flow on trees. In *Graphs, Morphisms and Statistical Physics, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, March 19-21, 2001*, pages 155–170, 2001. doi: 10.1090/dimacs/063/12. URL https://doi.org/10.1090/dimacs/063/12.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proc. of the 29th IEEE Symposium on Security and Privacy*, pages 111–125. IEEE Computer Society, May 2008. doi: 10.1109/SP.2008.33. URL http://www.cs.utexas.edu/~shmat/shmat_oak08netflix.pdf.

Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *30th IEEE Symposium on Security and Privacy (S&P 2009), 17-20 May 2009, Oakland, California, USA*, pages 173–187, 2009. doi: 10.1109/SP.2009.22. URL https://doi.org/10.1109/SP.2009.22.

Panos M. Pardalos, Franz Rendl, and Henry Wolkowicz. The quadratic assignment problem: A survey and recent developments. In Panos M. Pardalos and Henry Wolkowicz, editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–42. DIMACS/AMS, 1993. ISBN 978-0-8218-6607-8. URL http://dblp.uni-trier.de/db/conf/dimacs/dimacs16.html#PardalosRW93.

Pedram Pedarsani and Matthias Grossglauser. On the privacy of anonymized networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1235–1243, 2011. doi: 10.1145/2020408.2020596. URL https://doi.org/10.1145/2020408.2020596.

Mikhail Zaslavskiy, Francis R. Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2227–2242, 2009. doi: 10.1109/TPAMI.2008.245. URL https://doi.org/10.1109/TPAMI.2008.245.

Appendix A. Algorithms for matching weights

We here describe algorithms to compute recursively matching weights $\mathcal{W}_d(i \leftarrow j, u \leftarrow v)$ and $\mathcal{W}_d(i, u)$.

Algorithm 2 $\mathcal{W}_d(i, u)$

---
if $d = 0$ **then**
⎢   **return** 1
else
⎢ $\mathcal{E} \leftarrow \mathcal{N}_{\mathcal{T}}(i)$
⎢ $\mathcal{F} \leftarrow \mathcal{N}_{\mathcal{T}'}(i)$ **for** $(k, w) \in \mathcal{E} \times \mathcal{F}$ **do**
⎢   ⎢   Compute $\mathcal{W}_{d-1}(k \leftarrow i, w \leftarrow u)$
⎢   **end**
⎢   Solve the LAP problem $w^* := \sup_{\mathfrak{m} \in \mathcal{M}(\mathcal{E}, \mathcal{F})} \sum_{(k,w) \in \mathfrak{m}} \mathcal{W}_{d-1}(k \leftarrow i, w \leftarrow u)$
⎢   **return** w^*
end

Algorithm 3 $\mathcal{W}_d(i \leftarrow j, u \leftarrow v)$

---
if $d = 0$ **then**
⎢   **return** 1
else
⎢ $\mathcal{E} \leftarrow \mathcal{N}_{\mathcal{T}}(i) \setminus \{j\}$
⎢ $\mathcal{F} \leftarrow \mathcal{N}_{\mathcal{T}'}(i) \setminus \{v\}$ **for** $(k, w) \in \mathcal{E} \times \mathcal{F}$ **do**
⎢   ⎢   Compute $\mathcal{W}_{d-1}(k \leftarrow i, w \leftarrow u)$
⎢   **end**
⎢   Solve the LAP problem $w^* := \sup_{\mathfrak{m} \in \mathcal{M}(\mathcal{E}, \mathcal{F})} \sum_{(k,w) \in \mathfrak{m}} \mathcal{W}_{d-1}(k \leftarrow i, w \leftarrow u)$
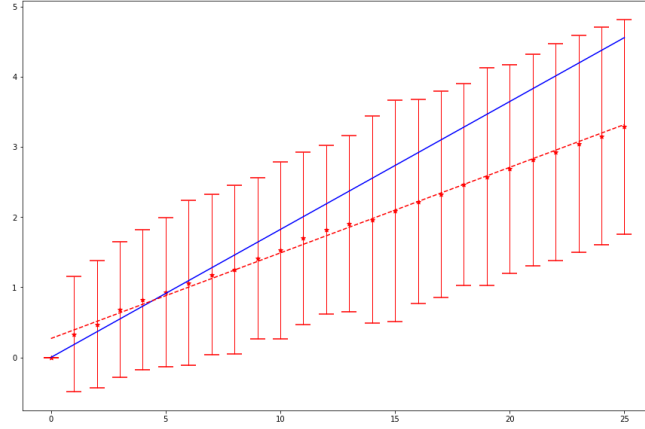⎢   **return** w^*
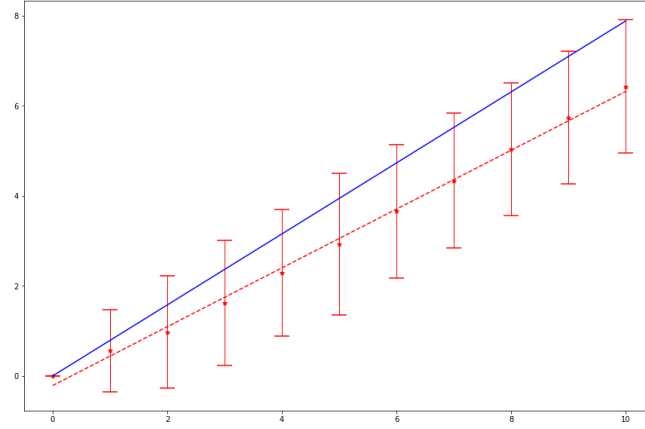**end**

Appendix B. Simulations

B.1. Simulations for tree matching

We here present some simulations of matching rates $\gamma(\lambda)$ (figure 4) and $\gamma(\lambda, s, \delta)$ for $s = 1$ (figure 5) in order to illustrate Theorems 1.1 and 1.2 and the final conjecture. For these simulations, error bars correspond to one standard deviation.
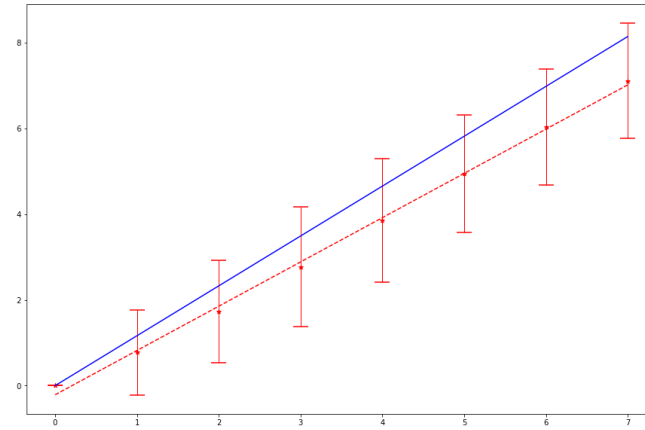
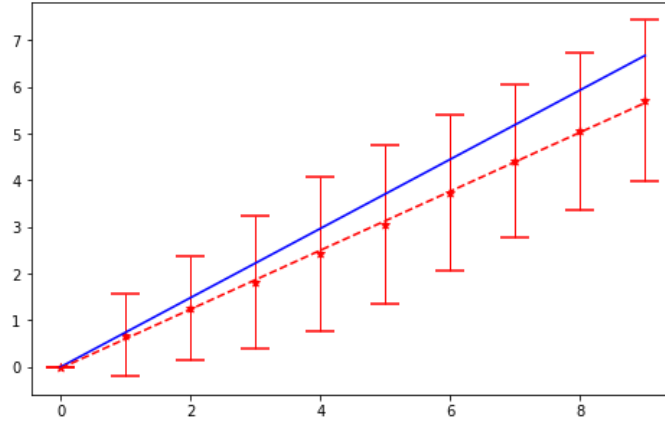(a) $\lambda = 1.2$, $\log \lambda \sim 0.18$. Red dashed slope ~ 0.12.

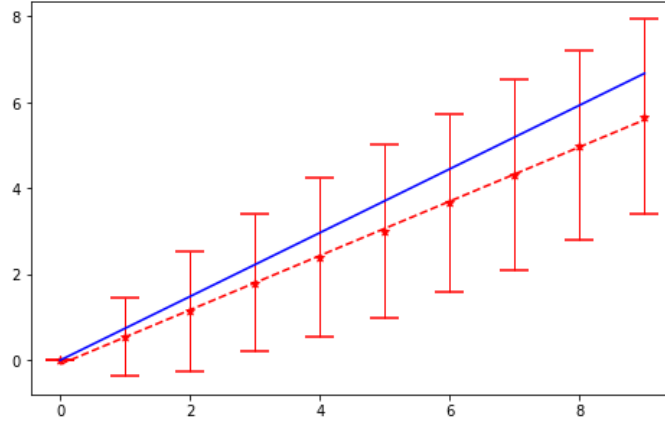(b) $\lambda = 2.2$, $\log \lambda \sim 0.79$. Red dashed slope ~ 0.65.

(c) $\lambda = 3.2$, $\log \lambda \sim 1.16$. Red dashed slope ~ 1.03.

Figure 4: Comparison of $d \log \lambda$ (blue) and $\log \mathcal{W}_d(\mathcal{T}, \mathcal{T}')$ (red) for $\mathcal{W}_d(\mathcal{T}, \mathcal{T}') \sim GW(\lambda)$ conditioned to survive (100 iterations).

(*a*) $\lambda = 2.1, \log \lambda \sim 0.74, \delta = 1$. Red dashed slope ~ 0.63.

(*b*) $\lambda = 2.1, \log \lambda \sim 0.74, \delta = 2$. Red dashed slope ~ 0.63.

(*c*) $\lambda = 2.1, \log \lambda \sim 0.74, \delta = 5$. Red dashed slope ~ 0.62.
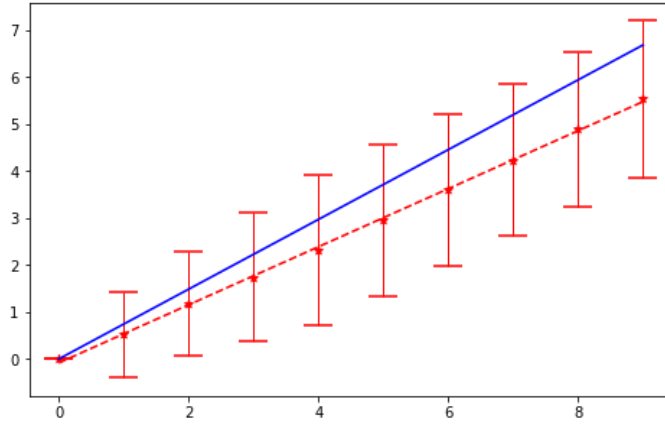
Figure 5: Comparison of $d \log \lambda$ (blue) and $\log \mathcal{W}_d(\mathcal{T}, \mathcal{T}')$ (red) for $\mathcal{W}_d(\mathcal{T}, \mathcal{T}') \sim GW(\lambda, s, \delta)$ with $s = 1$, conditioned to survive (50 iterations).

B.2. Simulations for a simple variant algorithm of NTMA

We here present some simulations of simple variant algorithm of NTMA, NTMA–2, which happens to be more efficient in practice. The Algorithm NTMA–2 is as follows.

Algorithm 4 NTMA–2

Input: Two graphs G_1 and G_2 of size n, average degree λ, depth d, parameter γ.
Output: A set of pairs $\mathcal{S} \subset V(G_1) \times V(G_2)$.
$\mathcal{S} \leftarrow \varnothing$
for $(i, u) \in V(G_1) \times V(G_2)$ **do**
    if $\mathcal{W}_d(i, u) > \gamma^d$, $\mathcal{W}_d(i, u) = \max_j \mathcal{W}_d(j, u)$ *and* $\mathcal{W}_d(i, u) = \max_v \mathcal{W}_d(i, v)$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, u)\}$
    end
end
for $(i, u) \neq (j, v) \in \mathcal{S}$ **do**
    if $i = j$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(i, y), y \in V(G_2)\}$
    end
    if $u = v$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(x, u), x \in V(G_1)\}$
    end
end
return $\mathcal{S}$

This algorithm only selects rows and columns weight maximums and match the corresponding pairs. The last part ensures that $\mathcal{S}$ is a matching. For these simulations, error bars correspond to a confidence interval for the mean value of scores. In figures 6 and 7 we compare the scores of NTMA–2 for $s = 0.95$ with the isomorphism case $s = 1.0$, for different values of n. We illustrate the fact that nearly no vertex is mismatched, whereas a non-negligible fraction of nodes is indeed recovered. In figure 8, we compare the scores of NTMA–2 for fixed n but varying s, illustrating the existence of a 'critical' parameter $s^*(\lambda)$.

## Appendix C. Detailed proofs for Section 1

C.1. Proof of Theorem 1.1

Proof [Proof of Theorem 1.1] We first state an easy corollary:

Corollary C.1 *For any $d \geq 1$, the random variable $X = |\mathcal{L}_d(\mathcal{T}_d)|$ is such that $\mathbb{E}\left[e^{\theta X}\right] < \infty$ for all $\theta > 0$.*
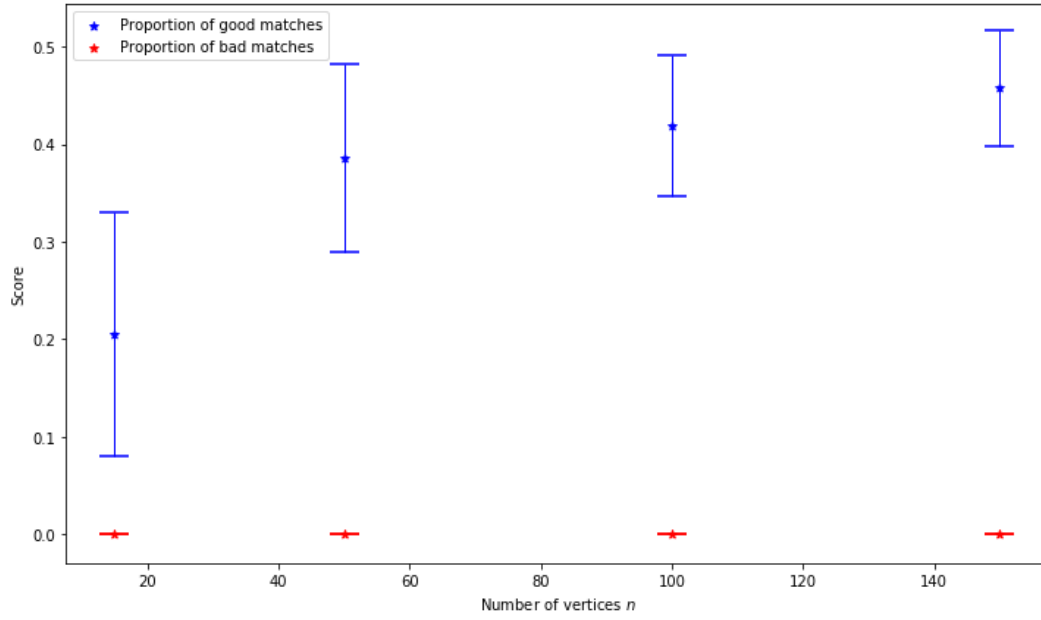
Proof This is easily seen by induction, based on the structure of $\mathcal{T}_d$ given in Lemma 1.1. ∎

Recall that we let $\mathcal{E}_d$ (respectively, $\mathcal{E}_d'$) denote the event that tree $\mathcal{T}$ (respectively, $\mathcal{T}'$) becomes extinct before d generations, i.e. $\mathcal{L}_d(\mathcal{T}) = \varnothing$ (respectively, $\mathcal{L}_d(\mathcal{T}') = \varnothing$). We let $p_d = \mathbb{P}(\mathcal{E}_d)$. It is well known that it satisfies the recursion

$$p_0 = 0, \; p_d = e^{-\lambda(1 - p_{d-1})},$$
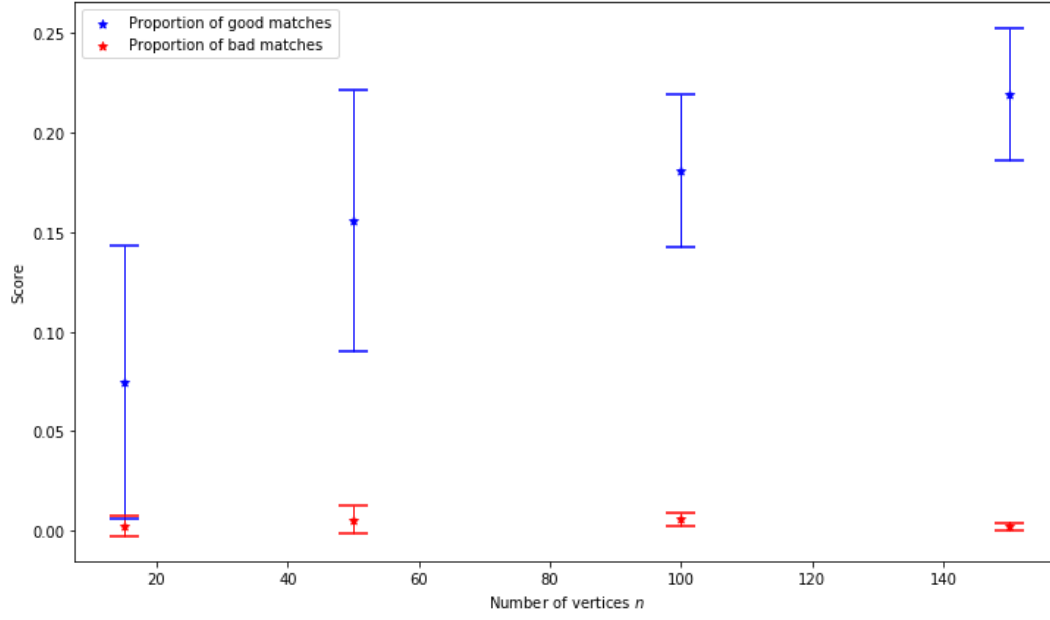
(a) $s = 0.95$.

(b) Isomorphism case , $s = 1.0$.

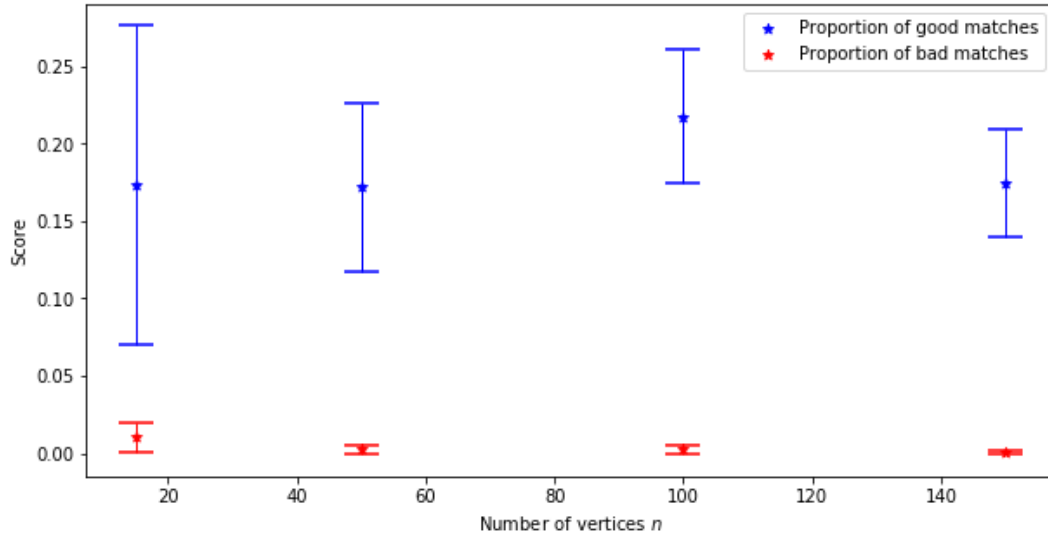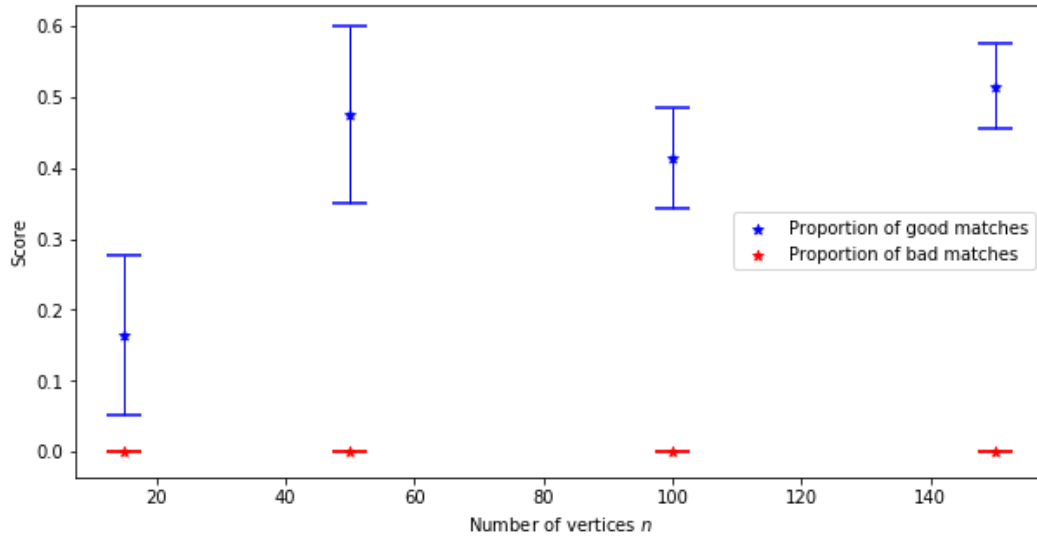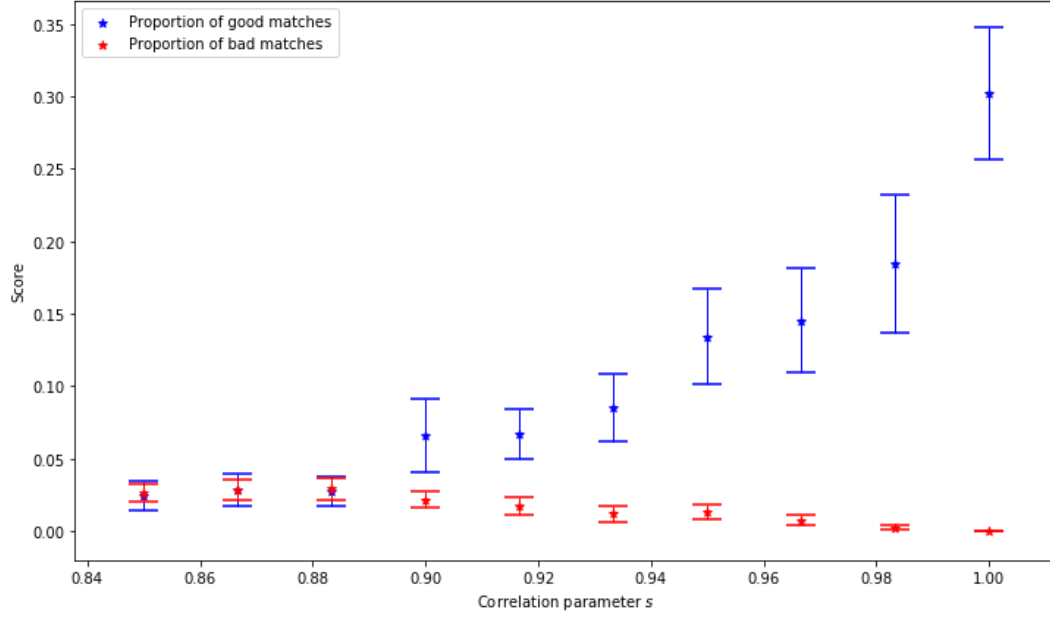Figure 6: Mean score of NTMA–2 for $\lambda = 2.1$, $d = 5$ (25 iterations per value of n).
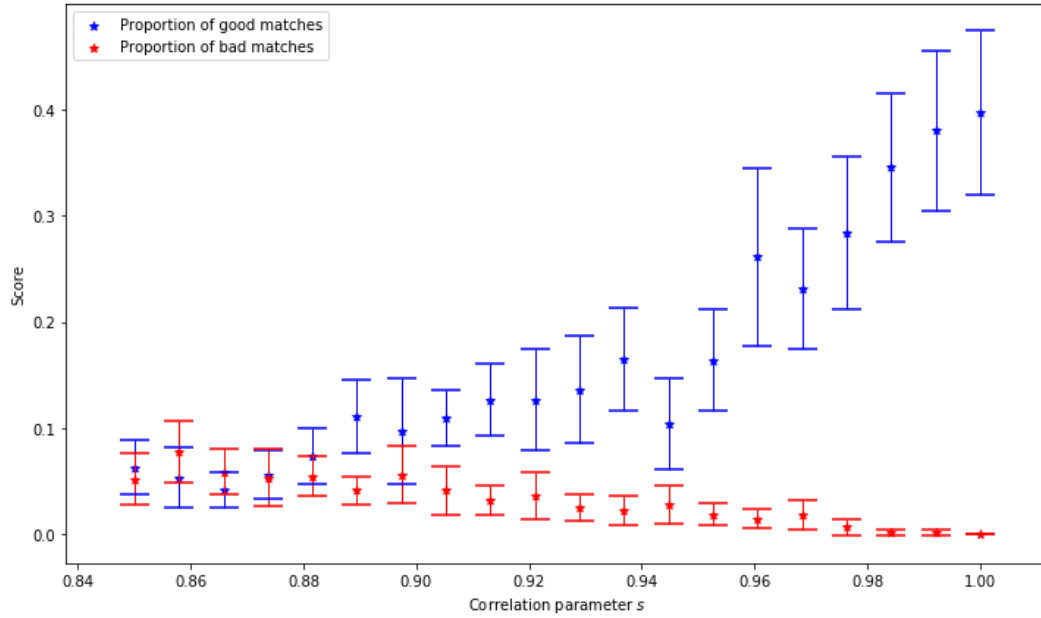
(*a*) $s = 0.95$.

(*b*) Isomorphism case , $s = 1.0$.

Figure 7: Mean score of NTMA–2 for $\lambda = 3.1$, $d = 4$ (25 iterations per value of n).

(a) $n = 150, \lambda = 1.4, d = 5$.

(b) $n = 50, \lambda = 2.2, d = 3$.

Figure 8: Mean score of NTMA–2 with different values of s (25 iterations per value of n).

and converges monotonically to the smallest root in $[0, 1]$ of $x = e^{-\lambda(1-x)}$. This root, that we denote p_e, is the probability of ultimate extinction. For small enough $\varepsilon = \lambda - 1$, it holds that

$$p_e = 1 - 2\varepsilon + O(\varepsilon^2),$$

as can be seen by analysis of the fixed point equation satisfied by p_e. Let then d_0 be such that for all $d \geq d_0$, $p_d = 1 - 2\varepsilon + O(\varepsilon^2)$. Clearly, on the event $\mathcal{E}_d \cup \mathcal{E}_d'$, the set of matchings $\mathcal{M}_d(\mathcal{T}, \mathcal{T}')$ is empty, so that $\mathcal{W}_d(\mathcal{T}, \mathcal{T}') = 0$. Recall that we define $\mathcal{T}_d$ the random variable $r_d(\mathcal{T})$ where $\mathcal{T}$ is conditioned to survive up to depth d.

Now fix $r \in (0, 1)$. We shall prove that for sufficiently small $\varepsilon > 0$, letting $\gamma = 1 + r\varepsilon$, there exists some constants $c, m, d_0 > 0$ such that for all $x > 0$, all $d \geq d_0$, one has

$$\mathbb{P}\left(\mathcal{W}_d(\mathcal{T}_d, \mathcal{T}_d') \geq \gamma^{d-d_0} mx\right) \leq e^{-(x-c)_+}. \tag{17}$$

We proceed by induction over $d - d_0$. To initialize the induction, notice that one obviously has $\mathcal{W}_{d_0}(\mathcal{T}_{d_0}, \mathcal{T}_{d_0}') \leq |\mathcal{L}_{d_0}(\mathcal{T}_{d_0})| =: X$. By Corollary C.1, for all $m, x, \theta > 0$, one has:

$$\mathbb{P}\left(\mathcal{W}_{d_0}(\mathcal{T}_{d_0}, \mathcal{T}_{d_0}') > mx\right) \leq \mathbb{P}(X > mx) \leq \mathbb{E}e^{\theta X}e^{-\theta mx}.$$

Let now $\theta = 1/m$. By taking m sufficiently large, from dominated convergence we can make $\mathbb{E}e^{(1/m)X}$ as close to 1 as we like. Choose for instance m such that $\mathbb{E}e^{(1/m)X} \leq 2$. Then

$$\mathbb{P}(\mathcal{W}_{d_0}(\mathcal{T}_{d_0}, \mathcal{T}_{d_0}') > mx) \leq 2e^{-x} \leq e^{-x+c}.$$

for any $c \geq \ln(2)$. Hence, for sufficiently large m, we can initialize the induction at $d = d_0$ with any $c \geq \ln(2)$.

Recall we set $\gamma = 1 + r\varepsilon$. Define the random variables

$$X_d := \gamma^{-(d-d_0)}m^{-1}\mathcal{W}_d\left(\mathcal{T}_d, \mathcal{T}_d'\right).$$

Let D (resp. D') denote the number of children of the root in $\mathcal{T}_d$ (resp. D' in $\mathcal{T}_d'$). Given D and D', noting $\mathcal{T}_d = (\mathcal{T}_{d-1,1}, \ldots, \mathcal{T}_{d-1,D})$ and $\mathcal{T}_d' = \left(\mathcal{T}_{d-1,1}', \ldots, \mathcal{T}_{d-1,D'}'\right)$, we have that

$$\mathcal{W}_d(\mathcal{T}_d, \mathcal{T}_d') = \sup_{\mathfrak{m} \in \mathcal{M}([D],[D'])} \sum_{(i,u)\in\mathfrak{m}} \mathcal{W}_{d-1}(\mathcal{T}_{d-1,i}, \mathcal{T}_{d-1,u}'),$$

where $\mathcal{M}([D], [D'])$ denotes the set of all $(D \vee D')^{\underline{D \wedge D'}}$ maximal injective mappings between $\mathcal{E}_0 \subseteq [D]$ and $[D']$. Let

$$X_{d-1,i,u} := \gamma^{-(d-1-d_0)}m^{-1}\mathcal{W}_{d-1}(\mathcal{T}_{d-1,i}, \mathcal{T}_{d-1,u}').$$

Note that conditional on D and D', for each matching $\mathfrak{m} \in \mathcal{M}([D], [D'])$, the variables $(X_{d-1,i,u})_{(i,u)\in\mathfrak{m}}$ are i.i.d. with the same distribution as X_{d-1}. The induction hypothesis states that each $X_{d-1,i,u}$ is less, for the strong stochastic ordering of comparison of cumulative distribution functions, than c plus an exponential random variable with parameter 1. With an easy union bound, we can derive the following bounds:

$$\mathbb{P}\left(X_d > x\right) \leq \sum_{1 \leq k \leq \ell < \infty} \mathbb{P}\left(D \wedge D' = k, D \vee D' = \ell\right) \min\left(1, \ell^{\underline{k}}\, \mathbb{P}\left(\mathcal{E}_1 + \ldots + \mathcal{E}_k > \gamma x - kc\right)\right),$$

(18)

where $\mathcal{E}_1, \ldots, \mathcal{E}_k$ are independent exponential random variables of parameter 1. Lemma 1.1 states that

$$\mathbb{P}(D = k) = e^{-\lambda(1 - p_{d-1})} \frac{\lambda^k (1 - p_{d-1})^k}{k!\, (1 - p_d)} =: q_{d,k}.$$

We can increase d_0 such that for some constant $\kappa > 0$, for all $d \geq d_0$:

$$q_{d,1} \leq 1 - \varepsilon + \kappa\varepsilon^2, \quad q_{d,k} \leq \frac{(3\varepsilon)^{k-1}}{k!}, \; k \geq 2.$$

Note that for $x \leq c$, there is nothing to prove in (17), since a probability is always upper-bounded by 1. We thus only need to consider the case $x > c$. We conclude the proof of this Theorem by appealing to the following

Lemma C.1 *Let $\kappa, C > 0$ and $r \in (0, 1)$ be given constants. Then there exists $c > 0$ large enough and $\varepsilon_0 > 0$ such that, for all $\varepsilon \in (0, \varepsilon_0)$, letting $\gamma = 1 + r\varepsilon$, $q_1 = 1 - \varepsilon + \kappa\varepsilon^2$, $q_k = (C\varepsilon)^{k-1}/k!$ for $k \geq 2$, one has*

$$\forall x > c, \quad \sum_{k,\ell \geq 1} q_k q_l \min\left(1, (k \vee \ell)^{\underline{k \wedge \ell}}\, \mathbb{P}\left(\mathcal{E}_1 + \ldots + \mathcal{E}_{k \wedge \ell} > \gamma x - (k \wedge \ell)c\right)\right) \leq e^{-(x-c)}, \quad (19)$$

where the $\mathcal{E}_i$ are independent exponential random variables of parameter 1.

Its assumptions are indeed verified here with $C = 3$, so (17) can be propagated by using this Lemma in (18), and the conclusion of Theorem 1.1 follows.

∎

C.2. Proof of Theorem 1.2

Proof [Proof of Theorem 1.2] We assume that $\lambda = 1 + \varepsilon$. We fix $r \in (0, 1)$, and we let $\gamma = 1 + r\varepsilon$ for some fixed $r \in (0, 1)$. . We work with trees such that $(\mathcal{T}, \mathcal{T}') \sim GW(\lambda, s, \delta)$. If we assume that the path from ρ to ρ' does not survive down to depth d in $\mathcal{T}$, then this path is no more present in $\mathcal{T}_d$, and the two trees $\mathcal{T}_d$ and $\mathcal{T}_d'$ can be coupled with two trees $\widetilde{\mathcal{T}}_d$ and $\widetilde{\mathcal{T}}_d'$ where $(\widetilde{\mathcal{T}}, \widetilde{\mathcal{T}}') \sim GW(\lambda)$, and we are in the case of Theorem 1.1.

In the following proof, we will thus condition to the event $S_{\rho,d}$ that the path from ρ to ρ' survives down to depth d in $\mathcal{T}$. Recall that the tree $\mathcal{T}_d$ (resp. $\mathcal{T}_d$) is obtained, conditionally on the fact that $\mathcal{T}$ (resp. in $\mathcal{T}'$) survives down to depth d, by suppressing nodes at depth greater than d in $\mathcal{T}$ (resp. in $\mathcal{T}'$), and then pruning alternatively leaves of depth strictly less than d. As in the proof of Theorem 1.1, we shall establish that for sufficiently small $\varepsilon > 0$, there exist constants $c, m, d_0 > 0$ such that for all $x > 0$, all $d \geq d_0$, one has

$$\mathbb{P}\left(\mathcal{W}_d(\mathcal{T}_d, \mathcal{T}_d') \geq \gamma^{d-d_0} mx \big| S_{\rho,d}\right) \leq e^{-(x-c)^+}.$$

(20)

Define the random variables

$$X_d' := \gamma^{-(d-d_0)} m^{-1} \mathcal{W}_d \left(\mathcal{T}_{d+\delta}, \mathcal{T}_d' \right),$$

conditional on $S_{\rho,d}$. The proof will again be by induction on d, the initial step being established with the same argument as in the proof of Theorem 1.1. Note that this argument does not depend on δ.

Denote by D the number of children of ρ in $\mathcal{T}_d$, D' the number of children of ρ' in $\mathcal{T}_d'$ that are in the intersection tree $\mathcal{T}_d \cap \mathcal{T}_d'$, and D'' the number of children of ρ' in $\mathcal{T}_d' \setminus \mathcal{T}_d$. By branching property, note that these three variables are independent.

Recall that p_d denotes the probability that a Galton-Watson tree with offspring $\mathrm{Poi}(\lambda)$ becomes extinct before d generations. Then, conditionally on $S_{\rho,d}$, the random variables D, D' and D'' have the following distributions:

$$
\begin{aligned}
D &\sim 1 + \mathrm{Poi}\left(\lambda\left(1 - p_{d-1}\right)\right), \\
D' &\sim \mathrm{Poi}\left(\lambda s \left(1 - p_{d-1}\right)\right) \\
D'' &\sim \mathrm{Poi}\left(\lambda(1 - s)\left(1 - p_{d-1}\right)\right), \text{ conditional on } D' + D'' > 0.
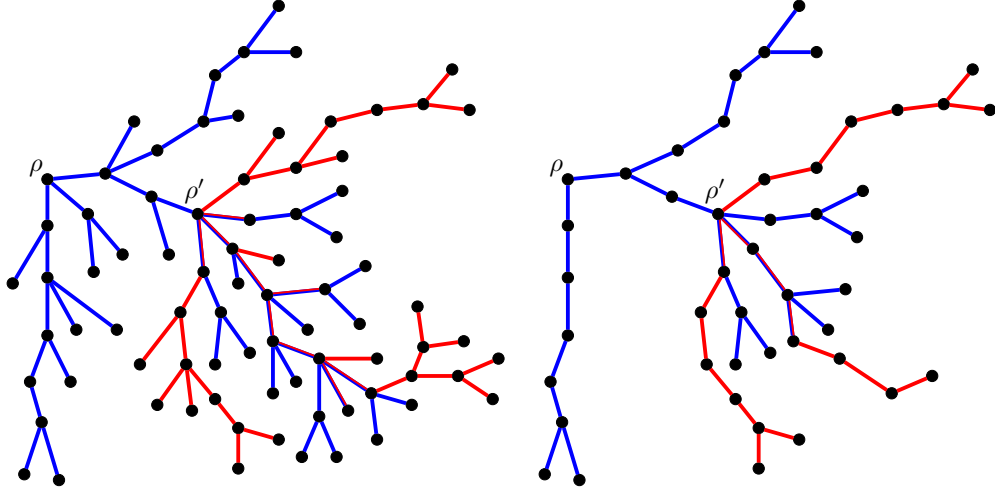\end{aligned}
$$

We show an illustration on Figure 9.

Figure 9: Random trees $\mathcal{T}$ (blue) and $\mathcal{T}'$ (red) from Figure 3 (left), and the results $\mathcal{T}_d$ and $\mathcal{T}_d'$ after applying r_d (right). In this example, $\delta = 3$ and $d = 6$, $D = 2$, $D' = 2$ and $D'' = 1$.

We condition on the values ℓ, k', k'' taken by D, D', D''. The number of maximal one-to-one mappings between the children of ρ in $\mathcal{T}_d$ and those of ρ' in $\mathcal{T}_d'$ is given by $[(k' + k'') \vee \ell]^{\underline{(k'+k'')\wedge\ell}}$, and each of them is of size $\ell \wedge (k' + k'')$. Note here again that for a fixed matching between the children of ρ and ρ', the weights of the matched sub-trees are independent. We distinguish between several cases (to help understand these cases, the reader could keep figure 9 in mind):

- For a child i of ρ that is not on the path to ρ', the corresponding sub-trees are independent so that the corresponding weight is distributed as $\mathcal{W}_{d-1}(\widetilde{\mathcal{T}}_{d-1}, \widetilde{\mathcal{T}}'_{d-1})$ in the independent-tree model $GW(\lambda)$.

- If the child of ρ on the path to ρ' is matched with a child of ρ' that is not in the intersection tree, again the corresponding weight is similarly distributed.

- Finally, if the child i of ρ leading to ρ' is matched to a child u of ρ' in the intersection tree, setting the new root at $\widetilde{\rho} := i$ in $\mathcal{T}$ and at $\widetilde{\rho}' := u$ in $\mathcal{T}'$, the corresponding weight has the same distribution as $\mathcal{W}_{d-1}(\mathcal{T}_{d-1}, \mathcal{T}'_{d-1})$ in the model $GW(\lambda, s, \delta)$, still conditioned to $S_{\widetilde{\rho}, d-1}$. Indeed, there is a path from $\widetilde{\rho}$ to $\widetilde{\rho}'$, and the corresponding Poisson distributions are conserved.

The induction hypothesis for case 3, together with Theorem 1.1 for cases 1 and 2, therefore give us:

$$\mathbb{P}\left(X'_d \geq x\right) \leq \sum_{k,\ell} \mathbb{P}\left(D' + D'' = k, D = \ell\right) \min\left(1, (k \vee \ell)^{k \wedge \ell} \mathbb{P}\left(\mathcal{E}_1 + \ldots + \mathcal{E}_{k \wedge \ell} > \gamma x - (k \wedge \ell)c\right)\right),$$

where the $\mathcal{E}_i$ are independent exponential random variables of parameter 1. Assume, as in the proof of Theorem 1.1, that d_0 is chosen such that for all $d \geq d_0$,

$$p_d^\lambda = 1 - 2\varepsilon + O(\varepsilon^2).$$

With simple computations, we can then ensure that for some $\kappa > 0$, noting $q_{d,\cdot}$ the distribution of D, one has

$$q_{d,1} \leq 1 - \varepsilon + \kappa \varepsilon^2, \quad q_{d,k} \leq \frac{(3\varepsilon)^{k-1}}{(k-1)!} \leq \frac{(6\varepsilon)^{k-1}}{k!}, \ k \geq 2,$$

where we used $k \leq 2^{k-1}$ in the last step. By independence of D' and D'', $D' + D''$ follows a $\mathrm{Poi}(\lambda(1 - p_{d-1}))$ distribution, conditional on being positive. Noting $q'_{d,\cdot}$ this distribution, we have, as in the previous proof,

$$q'_{d,1} \leq 1 - \varepsilon + \kappa \varepsilon^2, \quad q'_{d,k} \leq \frac{(3\varepsilon)^{k-1}}{k!}, \ k \geq 2.$$

We can then invoke Lemma C.1 to conclude. Note that every control in the proof is made uniformly on $\delta \geq 1$. $\blacksquare$

C.3. Proof of lemma C.1

Proof [Proof of Lemma C.1]. Let

$$S_1 := e^{x-c} q_1^2 e^{-(\gamma x - c)_+} + 4 q_1 q_2 e^{-(\gamma x - c)_+},$$

$$S_2 := 2 e^{x-c} q_1 \sum_{\ell \geq 3} q_\ell \min\left(1, \ell e^{-(\gamma x - c)_+}\right),$$

$$S_3 := 2 e^{x-c} \sum_{2 \leq k \leq \ell} q_k q_\ell \min\left(1, \ell^{\underline{k}} \mathbb{P}\left(\mathcal{E}_1 + \ldots + \mathcal{E}_k > \gamma x - kc\right)\right).$$

Our goal is to show that for a suitable choice of c, for all $x > c$, $S_1 + S_2 + S_3 \leq 1$. One has

$$S_1 \leq e^{-r\varepsilon x}\left((1 - \varepsilon + \kappa\varepsilon^2)^2 + 2C\varepsilon\right) \leq e^{-r\varepsilon x}(1 + 2C\varepsilon), \tag{21}$$

and

$$S_2 \leq 2e^{-r\varepsilon x}(1 - \varepsilon + \kappa\varepsilon^2) \sum_{\ell \geq 3} \frac{(C\varepsilon)^{\ell-1}}{(\ell-1)!} \leq 2e^{-r\varepsilon x}\left(e^{C\varepsilon} - 1 - (C\varepsilon)\right) \leq 2e^{-r\varepsilon x}C^2\varepsilon^2. \tag{22}$$

We let k_0 be such that $\gamma x \in [k_0 c, (k_0 + 1)c)$. We then upper-bound S_3 by $A + B$ where

$$A = 2e^{x-c} \sum_{k=2}^{k_0} q_k \sum_{\ell \geq k} q_\ell \frac{\ell!}{(\ell-k)!} \mathbb{P}\left(\mathcal{E}_1 + \ldots + \mathcal{E}_k > \gamma x - kc\right), \tag{23}$$

$$B = 2e^{x-c} \sum_{k \geq (k_0+1)\vee 2} \sum_{\ell \geq k} q_k q_\ell. \tag{24}$$

One readily has

$$B \leq 2e^{-r\varepsilon x}e^{\gamma x - c} \sum_{k \geq (k_0+1)\vee 2} \frac{(C\varepsilon)^{k-1}}{k!} \sum_{\ell \geq k} \frac{(C\varepsilon)^{\ell-1}}{\ell!} \tag{25}$$

$$\leq 2e^{-r\varepsilon x}e^{\gamma x - c} \sum_{k \geq (k_0+1)\vee 2} \frac{(C\varepsilon)^{2(k-1)}}{k!} \tag{26}$$

$$\leq 2e^{-r\varepsilon x}e^{k_0 c}(C\varepsilon)^{2((k_0+1)\vee 2 - 1)} \tag{27}$$

$$\leq 2e^{-r\varepsilon x}\left(C\varepsilon e^c\right)^2, \tag{28}$$

where in the last steps we assumed that $C\varepsilon e^c < 1$, so that

$$e^{k_0 c}(C\varepsilon)^{2((k_0+1)\vee 2 - 1)} \leq (C\varepsilon e^c)^{2((k_0+1)\vee 2 - 1)} \leq (C\varepsilon e^c)^2.$$

Note that for $y \geq 0$, $\mathbb{P}\left(\mathcal{E}_1 + \ldots + \mathcal{E}_k > y\right) = \mathbb{P}(\mathrm{Poi}(y) < k) = e^{-y}\sum_{j=0}^{k-1} y^j/j!$. Write then

$$\begin{aligned} A &\leq 2e^{x-c}\sum_{k=2}^{k_0}\frac{(C\varepsilon)^{2(k-1)}}{k!}\sum_{j=0}^{k-1}e^{-\gamma x + kc}\frac{(\gamma x)^j}{j!} \\ &\leq 2e^{-r\varepsilon x}\sum_{k=2}^{k_0}\frac{(C^2 e^c)^k}{k!}\sum_{j=0}^{k-1}\frac{(\gamma x\varepsilon^2)^j}{j!}\varepsilon^{2(k-1-j)} \\ &\leq 2e^{-r\varepsilon x}\sum_{k=2}^{k_0}\frac{(C^2 e^c)^k}{k!}\left[\varepsilon^{2(k-1)} + e^{\gamma x\varepsilon^2} - 1\right] \\ &\leq 2e^{-r\varepsilon x}\left[\varepsilon^{-2}\left(e^{\varepsilon^2 C^2 e^c} - 1 - \varepsilon^2 C^2 e^c\right) + e^{C^2 e^c}\left(e^{\gamma x\varepsilon^2} - 1\right)\right] \end{aligned} \tag{29}$$

Summing the upper bounds (21)-(29), the desired property will then hold if for all $x > c$, one has:

$$\begin{aligned} &e^{-r\varepsilon x}\left[1 + 2C\varepsilon + 2C^2\varepsilon^2 + 2[C\varepsilon e^c]^2 + 2\varepsilon^{-2}\left(e^{\varepsilon^2 C^2 e^c} - 1 - \varepsilon^2 C^2 e^c\right)\right] \\ &+ 2e^{-r\varepsilon x}e^{C^2 e^c}\left(e^{\gamma x\varepsilon^2} - 1\right) \leq 1. \end{aligned} \tag{30}$$

The first term is, for any fixed c, and for sufficiently small ε, upper bounded $e^{-r\varepsilon x}\left(1 + (2C+1)\varepsilon\right)$. We now distinguish three cases for x.

- Case 1: $x \in [c, 1/\sqrt{\varepsilon}]$. The second term is then $O(\varepsilon\sqrt{\varepsilon})$. Provided $rc > 2C + 1$, since $e^{-r\varepsilon x} \leq e^{-r\varepsilon c} = 1 - r\varepsilon c + O(\varepsilon^2)$, the left-hand side of (30) is then upper-bounded by $1 - (rc - 2C - 1)\varepsilon + O(\varepsilon\sqrt{\varepsilon})$, and is thus less than 1.

- Case 2: $x \in [1/\sqrt{\varepsilon}, 1/\varepsilon]$. Since $e^{-r\varepsilon x} \leq e^{-r\sqrt{\varepsilon}} = 1 - \Omega(\sqrt{\varepsilon})$, and $e^{\gamma x \varepsilon^2} - 1 \leq e^{\gamma\varepsilon} - 1 = O(\varepsilon)$, the left-hand side of (30) is upper-bounded by $1 - \Omega(\sqrt{\varepsilon})$ and is thus less than 1.

- Case 3: $x \geq 1/\varepsilon$. The first term is then bounded by $e^{-r}(1 + (2C + 1)\varepsilon)$, which is less than $1 - \Omega(1)$ for ε small enough. Letting $y = \varepsilon x$, the second term reads

$$e^{-ry}[e^{\varepsilon\gamma y} - 1](2e^{C^2 e^c}).$$

For small ε, this function is maximized for $y = 1/r + O(\varepsilon)$, at which point it evaluates to $O(\varepsilon)$. Thus the left-hand side of (30) is upper-bounded by $1 - \Omega(1)$ in that range.

We have thus shown that for any $r > 0$, provided $c > (2C + 1)/r$, then for all sufficiently small ε, the desired property holds with $\gamma = 1 + r\varepsilon$. ∎

C.4. Proof of lemma 1.1

Proof [Proof of Lemma 1.1] For a tree t, we write $t = (t^1, \ldots, t^k)$ to represent the fact that its root has k children, whose offsprings are given by trees $t^1, \ldots, t^k$. Write, noting by D the number of children of $\rho(\mathcal{T})$, fixing $k \geq 1$, $t^1, \ldots, t^k \in \mathcal{A}_{d-1}$, and letting $S = i_1 < \cdots < i_k$ run over all k subsets of $[\ell]$:

$$\mathbb{P}(\mathcal{T}_d = (t^1, \ldots, t^k)) = \sum_{\ell \geq 0} \mathbb{P}(\mathcal{T}_d = (t^1, \ldots, t^k), D = \ell)$$

$$= \sum_{\ell \geq k} \sum_{S} \mathbb{P}\left(D = \ell, r_d(\mathcal{T}^{i_j}) = t^j, j \in [k], r_d(\mathcal{T}^v) = \varnothing, v \notin S \Big| \overline{\mathcal{E}_d}\right)$$

$$= \frac{1}{1 - p_d} \sum_{\ell \geq k} \binom{\ell}{k} e^{-\lambda} \frac{\lambda^\ell}{\ell!} p_{d-1}^{\ell-k} \prod_{j=1}^{k} \mathbb{P}(\mathcal{T}_{d-1} = t^j)(1 - p_{d-1})$$

$$= \frac{1}{1 - p_d} \frac{(\lambda(1 - p_{d-1}))^k}{k!} \prod_{j=1}^{k} \mathbb{P}(\mathcal{T}_{d-1} = t^j) \sum_{\ell \geq k} e^{-\lambda} \frac{(\lambda p_{d-1})^{\ell-k}}{(\ell - k)!}$$

$$= \frac{e^{-\lambda(1 - p_{d-1})}}{1 - p_d} \frac{(\lambda(1 - p_{d-1}))^k}{k!} \prod_{j=1}^{k} \mathbb{P}(\mathcal{T}_{d-1} = t^j)$$

The conclusion follows by noting that $1 - p_d = 1 - e^{-\lambda(1 - p_{d-1})}$. ∎

Appendix D. Detailed proofs for Section 2

The following proofs are adapted from the previous work of Massoulié (2013) and Bordenave et al. (2015).

D.1. Proof of Lemma 2.1

Proof [Proof of Lemma 2.1] Fix $K > 0$ to be specified later and $\gamma > 0$. Fix $i \in [n]$, and define

$$T := \inf \{t \leq d, |\mathcal{S}_G(i,t)| \geq K \log n\}.$$

If $T = \infty$, there is nothing to prove. Given $|\mathcal{S}_G(i, T-1)|$,

$$|\mathcal{S}_G(i,T)| \sim \mathrm{Bin}\left(n - |\mathcal{S}_G(i,0)| - \ldots - |\mathcal{S}_G(i,T-1)|, 1 - \left(1 - \frac{\lambda}{n}\right)^{|\mathcal{S}_G(i,T-1)|}\right).$$

Thus

$$|\mathcal{S}_G(i,T)| \overset{\mathrm{sto.}}{\leq} \mathrm{Bin}\left(n, \lambda K \frac{\log n}{n}\right).$$

Using Bennett's inequality, for $K' > \lambda K$:

$$\mathbb{P}\left(|\mathcal{S}_G(i,T)| \geq K' \log n\right) \leq e^{-\lambda K h\left(\frac{K'-\lambda K}{\lambda K}\right) \log n},$$

with $h(u) = (1+u)\log(1+u) - u$. This probability is $\leq n^{-2-\gamma}$ if K' is large enough to verify $\lambda K h\left(\frac{K'-\lambda K}{\lambda K}\right) > \gamma + 2$. With a simple use of the union bound, one gets that $|\mathcal{S}_G(i,T)| \in [K \log n, K' \log n]$ for all $i \in [n]$ with probability $1 - O(n^{-1-\gamma})$.

Take $\varepsilon > 0$ to be specified later. We then check by induction that with high probability, for all $T \leq t \leq d$,

$$|\mathcal{S}_G(i,t)| \in \left[K\left(\frac{\lambda}{2}\right)^{t-T}(\log n) \prod_{s=T}^{t}\left(1 - \varepsilon\left(\frac{\lambda}{2}\right)^{-\frac{s-T}{2}}\right), K'\lambda^{t-T}(\log n)\prod_{s=T}^{t}\left(1 + \varepsilon\lambda^{-\frac{s-T}{2}}\right)\right]. \tag{31}$$

The case $t = T$ is proved here above. We will next use the inequality

$$\lambda u/(2n) \leq \lambda u/n - \lambda^2 u^2/(2n^2) \leq 1 - (1 - \lambda/n)^u \leq \lambda u/n. \tag{32}$$

that holds as soon as $\lambda u/n < 1$.

Assuming (31) holds up to t, inequality (32) holds for $u = |\mathcal{S}_G(i,t)|$ for n large enough, since $|\mathcal{S}_G(i,t)| < n/\lambda$ for $c \log \lambda < 1$. Thus for n large enough $\mathbb{E}|\mathcal{S}_G(i,t+1)|$ lies in the interval

$$\left[\frac{K}{2}\lambda\left(\frac{\lambda}{2}\right)^{t-T}(\log n)\underbrace{\prod_{s=T}^{t}\left(1 - \varepsilon\left(\frac{\lambda}{2}\right)^{-\frac{s-T}{2}}\right)}_{=1-O(\varepsilon)}, \lambda K'\lambda^{t-T}(\log n)\prod_{s=T}^{t}\left(1 + \varepsilon\lambda^{-\frac{s-T}{2}}\right)\right]$$

With $\hat{\varepsilon} > 0$ to be specified later, Bennett's inequality writes

$$\mathbb{P}\left(\left||\mathcal{S}_G(i,t+1)| - \mathbb{E}|\mathcal{S}_G(i,t+1)|\right| \geq \hat{\varepsilon}\mathbb{E}|\mathcal{S}_G(i,t+1)|\right) \leq 2e^{-\frac{1}{2}\lambda\left(\frac{\lambda}{2}\right)^{t-T}\log n(1-O(\varepsilon))h(\hat{\varepsilon})},$$

which is $\leq n^{-2-\gamma}$ if $K\left(\frac{\lambda}{2}\right)^{t+1-T}h(\hat{\varepsilon}) > 2+\gamma$. Since for $u \to 0$, $h(u) = u^2/2 + o(u^2)$, it suffices to take $\hat{\varepsilon} = \varepsilon\left(\frac{\lambda}{2}\right)^{-\frac{t+1-T}{2}}$ with ε small enough and K large enough such that $K\varepsilon > 2+\gamma$. Thus (31) holds for $t+1$ with probability $1 - O(n^{-2-\gamma})$.

All this ensures that the desired inequality (12) holds for all $i \in [n]$, $t \in [d]$ with probability $1 - O(n^{-\gamma})$. ∎

D.2. Proof of Lemma 2.2

Proof [Proof of Lemma 2.2] Fix $i \in [n]$. Define

$$k^* := \inf\{t \leq d, \ \mathcal{B}_G(i,t) \text{ contains a cycle}\}.$$

Note that $k^* \geq 2$, and that if $k^* = \infty$ then $\mathcal{B}_G(i,d)$ does not contain any cycle. Now assume that $k^* < \infty$. For any $k \geq 2$, $k^* = k$ if and only if there are two vertices of $\mathcal{S}_G(i, k-1)$ that are connected, or if there is a vertex of $\mathcal{S}_G(i,k)$ connected to two vertices of $\mathcal{S}_G(i, k-1)$. On the event

$$\mathcal{A} := \bigcap_{t \leq d} \left\{ |\mathcal{S}_G(i,t)| < C(\log n)\lambda^t \right\},$$

this happens with probability at most

$$|\mathcal{S}_G(i,k-1)|^2 \times \frac{\lambda}{n} + |\mathcal{S}_G(i,k)| \times |\mathcal{S}_G(i,k-1)|^2 \times \frac{\lambda^2}{n^2} \leq C^2 \frac{(\log n)^2 \lambda^{2k}}{n} + C^3 \frac{(\log n)^3 \lambda^{3k}}{n^2}.$$

Taking $\varepsilon > 0$ such that $c\log\lambda \leq 1/2 - \varepsilon$, choosing C such that $\mathbb{P}(\mathcal{A}) = 1 - O\left(n^{-2\varepsilon}\right)$ with Lemma 2.1, the probability that $\mathcal{B}_G(i,d)$ contains a cycle is less than

$$\mathbb{P}(k^* < \infty) \leq \mathbb{P}(\bar{\mathcal{A}}) + \sum_{k=2}^{d} \mathbb{P}(k^* = k \mid \mathcal{A})$$

$$\leq O\left(n^{-2\varepsilon}\right) + O\left(\frac{(\log n)^2 \lambda^{2d}}{n}\right) + O\left(\frac{(\log n)^3 \lambda^{3d}}{n^2}\right)$$

$$\leq O\left(n^{-2\varepsilon}\right) + O\left((\log n)^2 n^{-2\varepsilon}\right) + O\left((\log n)^3 n^{-3\varepsilon}\right) \leq O(n^{-\varepsilon}).$$

∎

D.3. Proof of Lemma 2.3

Proof [Proof of lemma 2.3] For fixed $i \neq j \in [n]$, let $\left(\tilde{\mathcal{S}}(i,t)\right)_{t \leq d}$ and $\left(\tilde{\mathcal{S}}(j,t)\right)_{t \leq d}$ denote two independent realizations of the neighborhoods (i.e. with independent underlying Bernoulli variables). We then construct recursively a coupling $(\mathcal{S}(i,t), \mathcal{S}(j,t))_{t \leq k}$:

- For $k = 1$, take $\mathcal{S}(i,t)$ to be a set of vertices uniformly chosen among sets of $[n]$ of size $\left|\tilde{\mathcal{S}}(i,0)\right|$. Independently, take $\mathcal{S}(j,t)$ to be a set of vertices uniformly chosen among sets of $[n]$ of size $\left|\tilde{\mathcal{S}}(j,0)\right|$.

- Now if $k > 1$, construct $\mathcal{S}(i, k)$ as follows: select a subset of $[n] \setminus \left(\bigcup_{s \leq k-1} \mathcal{S}(i, s) \right)$ of size $\left| \tilde{\mathcal{S}}(i, k) \right|$ uniformly at random. Then we construct independently $\mathcal{S}(j, k)$ taking a uniform subset of $[n] \setminus \left(\bigcup_{s \leq k-1} \mathcal{S}(j, s) \right)$ of size $\left| \tilde{\mathcal{S}}(j, k) \right|$.

This coupling is well defined, and coincides with the independent setting up to step k as long as the sets $\bigcup_{s \leq k} \mathcal{S}(i, s)$ and $\bigcup_{s \leq k} \mathcal{S}(j, s)$ do not intersect. On the event

$$\mathcal{A} := \bigcap_{t \leq d} \left\{ |\mathcal{S}(i, t)|, |\mathcal{S}(j, t)| < C(\log n) \lambda^t \right\},$$

one has

$$\mathbb{E} \left[\left| \bigcup_{k \leq d} \mathcal{S}(i, s) \cap \bigcup_{k \leq d} \mathcal{S}(j, s) \right| \right] \leq \mathbb{E} \left[\sum_{k=1}^{d} \text{Bin} \left(C(\log n) \lambda^k, \frac{\sum_{t=1}^{k} C(\log n) \lambda^t}{n - \sum_{t=1}^{k} C(\log n) \lambda^t} \right) \right]$$

$$\leq C^2 (\log n)^2 \left(\frac{\lambda}{\lambda - 1} \right) \sum_{k=1}^{d} \frac{\lambda^{2k}}{n - \frac{\lambda}{\lambda-1} C(\log n) \lambda^k}$$

$$\leq O \left((\log n)^2 \lambda^{2d} / n \right)$$

if $(\log n) \lambda^d = o(n)$, which is the case if $c \log \lambda < 1$. The expectation is upper-bounded by $O \left((\log n)^2 \lambda^{2d} / n \right) = O \left((\log n)^2 n^{-2\varepsilon} \right)$ if $c \log \lambda \leq 1/2 - \varepsilon$.

With Lemma 2.1, choosing C such that $\mathbb{P}(\mathcal{A}) = 1 - O \left(n^{-2\varepsilon} \right)$, we get

$$d_{\text{TV}} \left(\mathcal{L} \left((\mathcal{S}_G(i, t), \mathcal{S}_G(j, t))_{t \leq d} \right), \mathcal{L} \left((\mathcal{S}_G(i, t))_{t \leq d} \right) \otimes \mathcal{L} \left((\mathcal{S}_G(j, t))_{t \leq d} \right) \right) \leq O((\log n)^2 n^{-2\varepsilon}) + \mathbb{P}(\bar{\mathcal{A}})$$
$$\leq O(n^{-\varepsilon}).$$

∎

D.4. Proof of Lemma 2.4

We work here conditionally on

$$\mathcal{A} := \bigcap_{t \leq d} \left\{ |\mathcal{S}_G(i, t)| < C(\log n) \lambda^t \right\}.$$

Let's define a Galton-Watson process as follows: set $Z_0 = 1$, and for $t > 0$, $\mathcal{L}(Z_t | \mathcal{G}_{t-1}) = \text{Poi}(\lambda Z_{t-1})$, where $\mathcal{G}_t = \sigma(Z_s, s \leq t)$. Fix $t > 0$. Conditionally on $\mathcal{F}_{t-1} := \sigma(|\mathcal{S}_G(i, s)|, s \leq t - 1)$, define a random variable W_t with distribution $\text{Poi}(\lambda |\mathcal{S}_G(i, t - 1)|)$. Note that

$$\mathcal{L} \left(|\mathcal{S}_G(i, t)| \, \middle| \, \mathcal{F}_{t-1} \right) = \text{Bin} \left(n - |\mathcal{S}_G(i, 0)| - \ldots - |\mathcal{S}_G(i, t - 1)|, \ 1 - \left(1 - \frac{\lambda}{n} \right)^{|\mathcal{S}_G(i, t-1)|} \right).$$

The Stein-Chen method (see e.g. Barbour and Chen (2005)) enables to bound $\mathrm{d}_{\mathrm{TV}}\left(\mathrm{Bin}(n, \lambda/n), \mathrm{Poi}(\lambda)\right)$ by $\min(1, \lambda^{-1})\lambda^2/n \leq \lambda/n$. We also use the classical bound $\mathrm{d}_{\mathrm{TV}}\left(\mathrm{Poi}(\lambda), \mathrm{Poi}(\lambda')\right) \leq |\lambda - \lambda'|$ together with inequality (32) (which holds for n large enough since $c\log\lambda < 1$) to obtain that conditionally on $\mathcal{F}_{t-1}$:

$$\mathrm{d}_{\mathrm{TV}}\left(|\mathcal{S}_G(i,t)|, W_t\right) \leq n^{-1}\left(n - |\mathcal{S}_G(i,0)| - \ldots - |\mathcal{S}_G(i,t-1)|\right)\frac{\lambda\,|\mathcal{S}_G(i,t-1)|}{n}$$

$$+ \left|\left(n - |\mathcal{S}_G(i,0)| - \ldots - |\mathcal{S}_G(i,t-1)|\right)\left(1 - \left(1 - \frac{\lambda}{n}\right)^{|\mathcal{S}_G(i,t-1)|}\right) - \lambda\,|\mathcal{S}_G(i,t-1)|\right|$$

$$\leq \frac{\lambda\,|\mathcal{S}_G(i,t-1)|}{n} + \lambda\,|\mathcal{S}_G(i,t-1)|$$

$$- \left(n - |\mathcal{S}_G(i,0)| - \ldots - |\mathcal{S}_G(i,t-1)|\right)\frac{\lambda\,|\mathcal{S}_G(i,t-1)|}{n} + \frac{\lambda^2\,|\mathcal{S}_G(i,t-1)|^2}{2n}.$$

Now, for $\varepsilon > 0$ such that $c\log\lambda \leq 1/2 - \varepsilon$, on the event $\mathcal{A}$, all variables $|\mathcal{S}_G(i,s)|$ are bounded by $C(\log n)n^{1/2-\varepsilon}$. This leads to

$$\mathrm{d}_{\mathrm{TV}}\left(|\mathcal{S}_G(i,t)|, W_t\right) \leq O\left((\log n)n^{-1/2-\varepsilon}\right) + O\left((\log n)^3 n^{-2\varepsilon}\right) + O\left((\log n)^2 n^{-2\varepsilon}\right)$$

$$= O\left((\log n)^3 n^{-2\varepsilon}\right).$$

This proves by induction that the total variation distance between $(|\mathcal{S}_G(i,t)|)_{t\leq d}$ and $(Z_t)_{t\leq d}$ is bounded by $O\left((\log n)^4 n^{-2\varepsilon}\right) = O\left(n^{-\varepsilon}\right)$, taking C large enough in Lemma 2.1 so that $\mathbb{P}(\mathcal{A}) \geq 1 - O\left(n^{-2\varepsilon}\right)$.

D.5. Proof of Theorem 2.2

Proof [Proof of Theorem 2.2] Define

$$d_{\max} := \max\left(\max_i \deg_{G_1}(i), \max_u \deg_{G_2}(u)\right).$$

We use the same notations as in the former proof: $G_{\cup} = G_1 \cup G_2$ and $G_{\cap} = G_1 \cap G_2$. Fix $i \in [n]$. In the rest of the proof we work conditionally to the event $C_{\cup,i,2d}$ that $\mathcal{B}_{G_{\cup}}(i, 2d)$ has no cycle. Since $c\log\lambda < 1/4$, $\mathbb{P}(C_{\cup,i,2d}) = 1 - o(1)$ by Lemma 2.2.

Fix another vertex $u \neq i$. The $d-$neighborhoods $\mathcal{B}_{G_1}(i,d)$ and $\mathcal{B}_{G_2}(u,d)$ have offspring distribution stochastically dominated by $\mathrm{Bin}(n, \lambda/n)$, which is also dominated by $\mathrm{Poi}(\lambda')$ as soon as $\lambda' = \lambda + O(1/n)$ (see e.g. Klenke and Mattner (2009)). We can choose λ' such that $\gamma > \gamma(\lambda', 0)$ still holds: indeed, by a standard coupling argument, one can see that $\gamma : \lambda \mapsto \gamma(\lambda)$ is increasing. We now build two dominating (in the usual edge presence sense) tree-like $d-$neighborhoods of i and u with the following construction.

- First, if the two neighborhoods don't intersect, just sample two independent trees from model $GW(\lambda')$ rooted in i and in u.

- If the two neighborhoods intersect, condition to the event that α is the contact point in the path $\mathfrak{p}_{\cup}$ (unique by conditioning on $C_{\cup,i,2d}$) from i to u in the joint graph. Then there is a path of edges of G_1 (say, blue) from i to α, then a path of edges of G_2 (say, red) from α

to u. Next, complete this construction: along $\mathfrak{p}_\cup$, propagate the blue path from α towards u with probability s on each edge, stopping at the first time when one red edge is not selected. Do the symmetrical construction to propagate the red path from α towards i. Finally, to each double-colored vertex, attach independent realizations of model $GW(\lambda', s)$, and to each single-colored vertex, attach independent realizations of model $GW(\lambda')$.

Note that these constructions lead to at most one path $\mathfrak{p}_\cup$ between i and u in $\mathcal{B}_{G_1}(i, d) \cup \mathcal{B}_{G_2}(u, d)$, so a fortiori in $\mathcal{B}_{G_1}(i, d) \cap \mathcal{B}_{G_2}(u, d)$. Denote by $\mathfrak{p}_\cap$ this hypothetical path (cf. figure 10). We then distinguish between several cases.
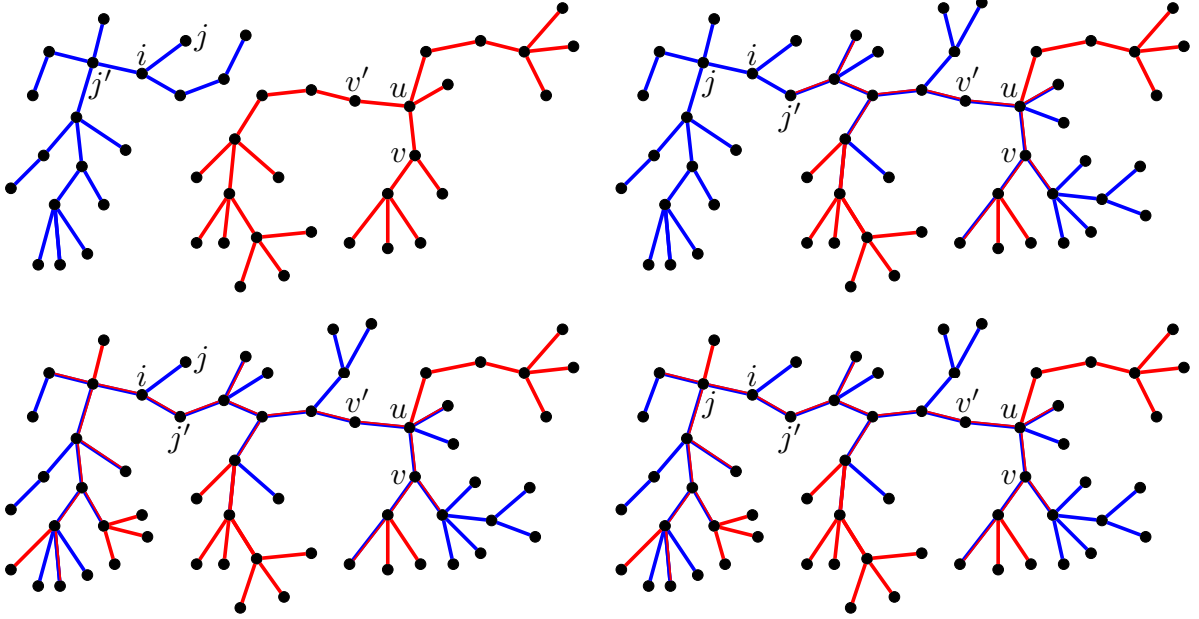

Figure 10: Possible realizations of $\mathcal{B}_{G_1}(i, d)$ (blue) and $\mathcal{B}_{G_2}(u, d)$ (red), with distinct cases (i) (top left), (ii) (top right), $(iii.a)$ (bottom left) and $(iii.b)$ (bottom right).

Case (i): $\delta_{G_\cup}(i, u) > 2d$ (figure 10, top left), i.e. $\mathcal{B}_{G_1}(i, d) \cap \mathcal{B}_{G_2}(u, d) = \varnothing$. The construction gives a coupling with two independent trees from model $GW(\lambda)$. By assumption $\gamma(\lambda) < \lambda s$, the probability that there exist j in $\mathcal{N}_{G_1}(i)$ and v in $\mathcal{N}_{G_2}(u)$ such that $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u) > \gamma^{d-1}$ is upper bounded by $O\left(d_{\max}^2 \exp\left(-n^\varepsilon\right)\right)$, following Remark 1.3. Hence i is matched to u with at most this probability.

Case (ii): $\delta_{G_\cup}(i, u) \leq 2d$ but $\mathfrak{p}_\cap$ does not exist (see figure 10, top right). Take $v \neq v'$ two neighbors of u and $j \neq j'$ two neighbors of i. Then (at least) one of these vertices is not on $\mathfrak{p}_\cup$ (e.g. vertex j on figure 10): the downstream tree from this vertex is independent from every other neighborhood in the other graph. They can be coupled with model $GW(\lambda)$, and the same bound as in case (i) holds.

Now assume that $\mathfrak{p}_\cap$ exists, and let $v \neq v'$ two neighbors of u and $j \neq j'$ two neighbors of i. Case $(iii.a)$: At least one of the edges $(i,j),(i,j'),(u,v),(u,v')$ is not in $G_\cap$ (e.g. edge (i,j) on figure 10, bottom left): again, the same argument applies. Case $(iii.b)$: Edges $(i,j),(i,j'),(u,v),(u,v')$ are all in $G_\cap$ (see figure 10, bottom right). Then one pair of vertices (say (j',v') as on figure 10) can be on $\mathfrak{p}_\cap$ and bring a high $\mathcal{W}_{d-1}(j' \leftarrow i, v' \leftarrow u) > \gamma^{d-1}$ matching weight, if their descendants spread over a great part of the intersection. In that case, since j and v can't be on $\mathfrak{p}_\cap$, the associated downstream trees are independent, and again $\mathcal{W}_{d-1}(j \leftarrow i, v \leftarrow u) < \gamma^{d-1}$ with high probability. The remaining case to be considered is that of matches (j,v') and (j',v), with j', v' on $\mathfrak{p}_\cap$. All trees involved are then correlated. However, the coupling construction induces a coupling of the two pairs of $(d-1)-$neighborhoods (from (j,v') and from (j',v), see figure 10) with two pairs of trees from model $GW(\lambda', s, \delta)$ where $\delta = |\mathfrak{p}_\cap|$. The Theorem assumes $\gamma(\lambda, s, \delta) < \lambda s$ so that, by Theorem 1.2, the probability that $\mathcal{W}_{d-1}(j \leftarrow i, v' \leftarrow u) > \gamma^{d-1}$ and $\mathcal{W}_{d-1}(j' \leftarrow i, v \leftarrow u) > \gamma^{d-1}$ is upper bounded by $O\left(\exp\left(-n^\varepsilon\right)\right)$.

Thus, for i fixed, one has

$$\mathbb{P}\left(\exists u \neq i,\ (i,u) \in \mathcal{S}\right) \leq 1 - \mathbb{P}\left(C_{\cup,i,2d}\right) + n \times \mathbb{P}\left(C_{\cup,i,2d}\right) \times d_{\max}^2 \times O\left(\exp\left(-n^\varepsilon\right)\right) = o(1).$$

The Theorem then follows by appealing to Markov's inequality. ∎