ABSTRACT

Title of dissertation: EXTENDING THEORIES OF THE GRAPH MATCHING PROBLEM AND ITS VARIANTS

Zhirui Li, Doctor of Philosophy, 2025

Dissertation directed by: Professor Vince Lyzinski
University of Maryland, College Park

Graphs serve as powerful tools for modeling complex real-world relationships, making reliable statistical inference on graphs a crucial task. A fundamental problem in this domain is the graph matching problem, which seeks to align node labels across graphs while minimizing structural and feature discrepancies. In this thesis, I investigate algorithms for the graph matching problem and one of its key variants, the subgraph detection problem.

I develop theoretical frameworks that leverage signals from a clustered, vertex-aligned collection of graphs to accurately recover node labels in a newly shuffled network and classify this new network into one of the clusters. Additionally, I propose a novel approach for detecting multiple instances of a noisily embedded template graph within a large background graph. Furthermore, I explore the relationship between the anonymization time and the mixing time of a specific class of Markovian noise applied to graph edges. Beyond these contributions, I address several related challenges in graph matching and its related optimization algorithms, offering new insights into their theoretical and practical aspects.

To validate our methodologies, I provide rigorous theoretical justifications and conduct extensive experiments using both simulated and real-world network data. These findings demonstrate the effectiveness of the proposed approaches, bridging the gap between theo-

retical advancements and practical applications in graph inference.

EXTENDING THEORIES OF THE GRAPH MATCHING PROBLEM AND ITS VARIANTS

by

Zhirui Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2025

Advisory Committee:
Professor Vince Lyzinski, Chair/Advisor
Professor Radu Balan
Professor Wojciech Czaja
Professor Bruce Golden
Professor Eric Slud

Acknowledgments

I am deeply grateful to my advisor, Dr. Vince Lyzinski, for his unwavering guidance, insightful mentorship, and steadfast support throughout my doctoral studies. His expertise and encouragement have been instrumental in shaping my research and academic growth.

I extend my sincere appreciation to my dissertation committee members: Dr. Wojciech Czaja, Dr. Radu Balan, Dr. Bruce Golden, and Dr. Eric Slud. Their valuable feedback, diverse perspectives, and constructive critiques have significantly enriched my work and broadened my academic horizons.

I am fortunate to have collaborated with exceptional researchers beyond the University of Maryland. Notably among these collaborators—though certainly not exhaustively—I would like to acknowledge Dr. Carey Priebe (Johns Hopkins University), Dr. Keith Levin (University of Wisconsin–Madison), Dr. Daniell Sussman (Boston University), and Dr. Jesús Arroyo (Texas A&M University). Their collaboration, guidance, and insights have been both enlightening and inspiring, profoundly influencing my research trajectory and career development.

My academic foundation was laid at the University of Iowa, where I was mentored by Dr. Yangbo Ye. His supervision and mentorship sparked my passion for applied mathematics and statistics. I also wish to express my gratitude to Dr. Boxiang Wang, Dr. Xueyu Zhu, and Dr. Osnat Stramer for their impactful teaching, dedicated supervision, and continuous support during my undergraduate studies.

To my fellow research group members: your camaraderie, intellectual discussions, and shared experiences have made this journey both productive and enjoyable. The collaborative environment we've fostered has been a constant source of motivation and intellectual stimulation.

I am also thankful to my friends and peers here at UMD, whose encouragement, thoughtful discussions, and mutual exploration of emerging research trends have significantly en-

riched my academic experience. Their support and understanding have provided balance, inspiration, and joy throughout my academic journey.

Finally, I owe my deepest gratitude to my family—my parents and grandparents—for their unconditional love, patience, and unwavering belief in my abilities. Their support has been the bedrock of my achievements.

Table of Contents

Chapter 1: Introduction

This dissertation focuses on the Graph Matching Problem (GMP) and its various extensions. A graph $G = (V, E)$ consists of two components: a vertex set V, typically representing entities, and an edge set E, representing connections between these entities (see Fig. 1 below). We denote the set of graphs containing n nodes as $\mathscr{G}_n$. Note that in the network science literature, the terms networks, nodes, and links are often used in place of graphs, vertices, and edges, respectively [12]. Throughout this dissertation, we use these terms interchangeably.

The statistical analysis of networks commonly starts with specifying a random network model to account for network-valued data [50, 59]. To facilitate statistical inference and enhance interpretability, researchers commonly use low-rank graph models. These models

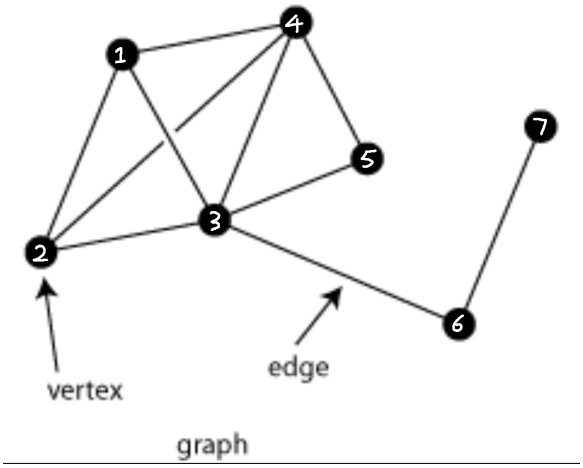

Figure 1.1: Illustration of a graph with 7 nodes

require significantly fewer parameters than the $O(n^2)$ edge probabilities necessary in a fully general network model, making inference more tractable. Common low-rank network models include:

Definition 1.1. *Let n and $p \in [0,1]$ be fixed. The Erdős–Rényi Model (ER) with parameters n, p, denoted as* $\mathrm{ER}(n,p)$*, is a probabilistic graph model where for any pair of vertices $i, j \in V$ with $i \neq j$, we have that an edge exists in G between i and j independently with probability p.*

Various thresholds of the graph properties for this model have been established in [21, 47] for the homogeneous Erdős-Rényi case and in [22] for the inhomogeneous (edges exist independently but with different probability) case.

An extension of this model is the stochastic block model (SBM), which introduces community structures within the graph [54].

Definition 1.2. *Let n be fixed, and let K be a positive integer. Consider a symmetric real matrix $\Lambda \in [0,1]^{K \times K}$. The Stochastic Block Model (SBM) with parameters $K, \Lambda, \tau, \vec{n}$, denoted as* $\mathrm{SBM}(K, \Lambda, \tau, \vec{n})$*, is a probabilistic graph model where*

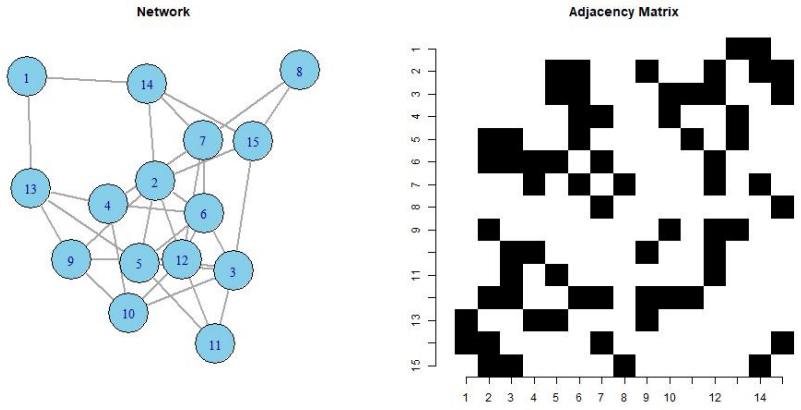

Figure 1.2: Network and Adjacency Matrix plots for a network sampled from the Erdős–Rényi Model with 15 nodes and $p = 0.3$.

i. K represents the number of node clusters (blocks) in the graph. These clusters/communities partition V; denote the i-th cluster via C_i for $i \in [K]$;

ii. The function $\tau : \{1,\dots,n\} \to \{1,\dots,K\}$ is a community assignment function that maps each node to its respective block/cluster. The vector $\vec{n}$ is a K-dimensional vector where $\vec{n}_i$ denotes the number of nodes belonging to community i, satisfying $\sum_{i=1}^{K} \vec{n}_i = n$

iii. The matrix Λ is a symmetric probability matrix such that for any pair of nodes assigned to communities i and j, an edge between them exists independently with probability Λ_{ij}.

In essence, the SBM model relaxes the assumption of having a uniform edge existing probability, allowing different values across different communities, as well as some (potentially) smaller probability values between nodes across communities. For theories and applications of such a model, we refer the reader to, for example, [1, 18, 56, 63, 77, 101, 113].

The aforementioned inhomogeneous Erdős–Rényi model allows each edge to have its

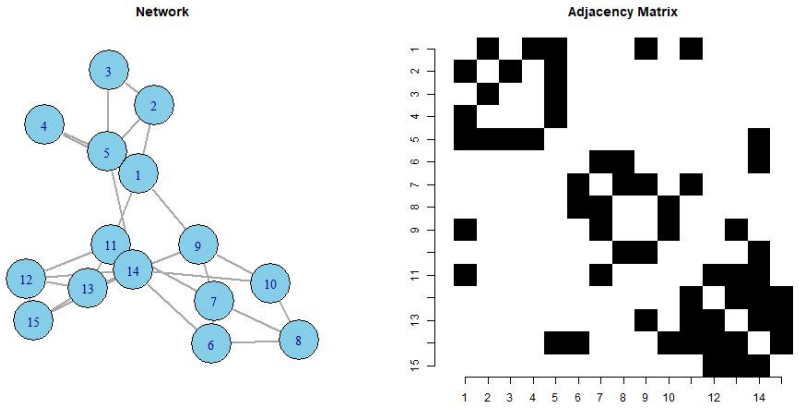

Figure 1.3: Network and adjacency matrix plots for a network sampled from a stochastic block model (SBM) with three communities of size 5 each. The within-community edge probabilities are set to 0.8 for all communities, and the across-community edge probabilities are uniformly set to 0.1.

own parameter, thus deviating from low-rank modeling. However, the random dot product graph (RDPG) model [135] offers greater flexibility in edge probabilities while requiring only parameters proportional to the number of nodes, see [7] for a detailed discussion of the RDPG model. The RDPG model is a special case of the more general Latent Space Model (LSM) [51].

Definition 1.3. *The Latent Space Model (LSM) [51], where vertices are embedded in a latent space, and edge probabilities are determined by a kernel function of their pairwise latent positions. Specifically, let n be a positive integer representing the number of nodes, and let d be a positive integer denoting the dimension of the latent space. The Latent Space Model with parameters* $n, d, f, \mathbf{Z}$*, denoted as* $\mathrm{LSM}(n, d, f, \mathbf{Z})$*, is a probabilistic graph model where each node* $i \in \{1, \ldots, n\}$ *is associated with a latent position* $\mathbf{Z}_i \in \mathbb{R}^d$*. The function* $f : \mathbb{R}^d \times \mathbb{R}^d \to [0,1]$ *is a kernel function that determines the probability of an edge between two nodes based on their latent positions, such that for any pair of nodes i and j, the edge* e_{ij} *exists independently with probability* $f(\mathbf{Z}_i, \mathbf{Z}_j)$*.*

These models (and their many variants) exhibit conditional independence of edges,

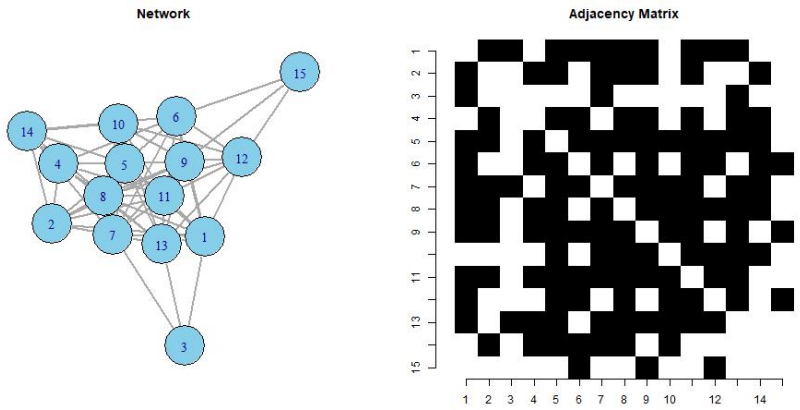

Figure 1.4: Network and adjacency matrix plots for a network sampled from a random dot product graph (RDPG) model with 15 nodes. The latent dimension d is set to 2, and the latent positions of the nodes are independently sampled uniformly from the unit square.

meaning that edges are independent given node memberships in the SBM or latent positions in the LSM. This property makes them mathematically tractable and facilitates establishing key statistical results such as consistent estimation [7, 17, 19], asymptotic normality [8, 116], and efficiency [118]. Although these models may not fully capture the complexity of real-world networks [105], there is a growing body of research demonstrating that they can nonetheless capture meaningful and significant structures in even complex real-world networks [26, 95, 123, 130].

Statistical network inference typically begins with observed graphs that are assumed to be noisy copies of an underlying latent background graph [6, 67, 95, 128]. A fundamental task in network inference is to estimate the parameters governing this latent structure, a problem referred to as latent position estimation. A widely used approach is Adjacency Spectral Embedding (ASE) [7, 112], which applies matrix decomposition techniques, such as principal component analysis (PCA), to the adjacency matrices of observed graphs. ASE and related spectral embedding methods have enabled various advancements in network analysis, including classification, hypothesis testing, and other inference tasks [32, 35, 100, 114, 124].

The reliability of the latent position estimation depends critically on the number of observed graphs, where we implicitly assume that the vertices of these observed networks have been aligned. However, inference tasks such as latent position estimation or network hypothesis testing deteriorate significantly if the nodes are not *a priori* aligned. To address this challenge, we introduce the graph matching problem—the task of recovering node correspondence between networks—which is the central focus of this thesis. We now formally introduce the graph matching problem.

1.1 The Graph Matching Problem

The formulation of the graph matching problem we consider herein can be defined as follows.

Definition 1.4. *Consider two undirected, loop-free graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with corresponding adjacency matrices A and B, respectively, where $|V_1| = |V_2| = n$. The NP-hard Graph Matching Problem (GMP) seeks*

$$\min_{P \in \Pi_n} \|A - PBP^T\|_F$$

where Π_n denotes the set of permutation matrices of order n and $\|\cdot\|_F$ represents the Matrix Frobenius Norm defined as $\|M\|_F = \left(\sum_{j=1}^{m} \sum_{k=1}^{n} M_{jk}^2\right)^{1/2}$ for any $m \times n$ real matrix M.

The goal is to find the best possible alignments between nodes of the two graphs to minimize the edgewise structural differences. If the graphs are allowed being weighted, directed and loopy the above formulation is equivalent to the NP-hard quadratic assignment problem. That said, there are efficient graph matching algorithms for special classes of graphs (e.g., planar graphs). Note that when discussing the GMP, we will refer to a graph and its adjacency matrix interchangeably in this context due to the one-to-one correspondence between weighted graphs and square matrices. Specifically, there is a bijection between the set of unweighted, undirected, loop-free graphs and the set of symmetric hollow matrices (hollow here means the diagonal entries of the matrix are all 0) with binary entries.

The GMP has a long and rich history. Initially, research primarily focused on determining graph isomorphisms (such that the objective function necessarily equals zero); see [10] for quasipolynomial complexity of the isomorphism problem. For these purposes, tree-based methodologies emerged, leveraging hierarchical structures to efficiently detect graph isomorphisms, thus enabling faster and more scalable solutions. However, the stringent

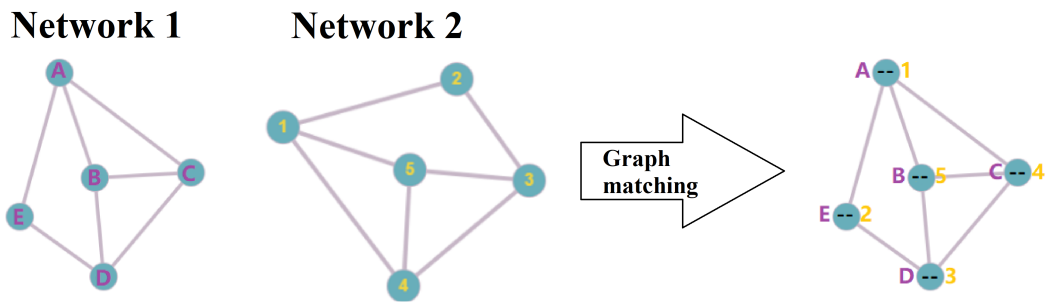

Figure 1.5: Simple Illustration of Graph Matching of 2 Graphs

nature of exact matching limited its applicability in real-world scenarios involving noise and distortion. Consequently, recent developments aim to approximate matches within noisy data, employing optimization techniques and probabilistic frameworks to more robustly handle cases where exact graph correspondences are unattainable. See [28] for more background.

We define here the term *matchability* of graphs as the ability to recover the node alignments with respect to a given ground truth. With this definition, it becomes clear that matchability is primarily determined by the edge correlations in the ground-truth edge pairs, and [30, 31, 73, 76, 132] identified a sharp threshold on edge correlation for graph matchability. In short, in moderately dense graphs, a correlation of order $\log n/n$ is required for matchability with high probability. Note that, when applicable, this sharp threshold could be used to quickly obtain rough, though potentially overly restrictive, conditions for graph matchability across various models and under different types of noise.

Before proceeding to the next section, we briefly mention that once the networks have been matched, we can measure the similarity between them using the *graph edit distance* (GED), defined as $\mathrm{GED}(G_1, G_2) = \min_{\text{edit set } S \text{ to make } G_1 = G_2} \sum_{s \in S} c(s)$, where G_1 and G_2 are two vertex-aligned networks and $c(s)$ is the cost of the edits. The allowed edits are: (i) add/delete nodes; (ii) substitute labels/features of nodes; (iii) add/delete edges; (iv)

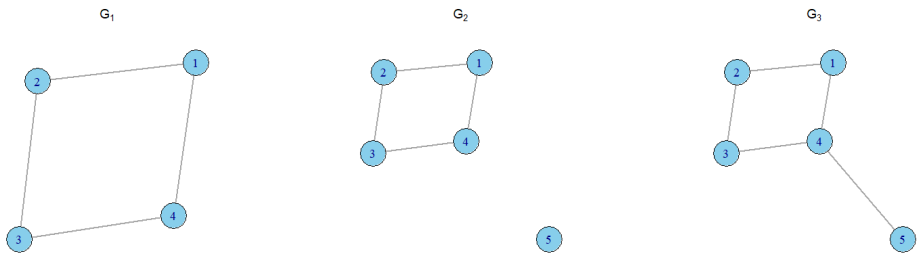

Figure 1.6: GED Network Plots

substitute labels/features of edges. Consider the pairwise graph edit distances among the networks plotted in Fig. 1.6. While it can be easily seen that GED(G_1, G_3) is the largest, the relationship between GED(G_1, G_2) and GED(G_2, G_3) depends on the costs of edge insertion and node deletion.

1.2 Overview and Structure

This dissertation extends the theoretical and methodological foundations of the Graph Matching Problem (GMP), introducing novel techniques to handle noisy and dynamically evolving networks.

The remainder of this dissertation is structured as follows:

- Section 1.3 provides a summary of the notation used throughout the thesis.
- Chapter 2 introduces a new algorithm that simultaneously performs node matching across networks and classification of a new graph using a set of clustered, node-aligned reference networks. The paper has been published in IEEE Transactions on Network Science and Engineering, Volume: 10 Issue: 6 [70].
  - Sections 2.1–2.4 present and compare different approaches for achieving this task, highlighting their respective advantages.

  - Sections 2.5 and 2.8 present primary and supplementary experimental evaluations to demonstrate the effectiveness of the proposed methods.
  - Section 2.6 discusses the findings and outlines future research directions.
  - Section 2.7 contains proofs of theoretical results.

- Chapter 3 proposes a new approach for identifying multiple noisy embeddings of a network template within a larger network. The paper has been published in IEEE Transactions on Signal and Information Processing over Networks, Volume: 10 [71].
  - Section 3.1 discusses background information and introduces the subgraph detection problem, a variant of GMP.
  - Section 3.2 presents a novel algorithm along with theoretical guarantees for its performance.
  - Sections 3.3 and 3.6 provide experimental results using both simulated and real data.
  - Section 3.4 offers discussions and potential future research directions.
  - Section 3.5 contains the proof of the main theorem.

- Chapter 4 investigates the theoretical relationship between graph anonymization time and mixing time under Markovian noise, a type of edge- and time-dependent stochastic perturbation.
  - Section 4.1 highlights the necessity of considering edge-dependent noise in network inference.
  - Section 4.2.1 introduces a Markovian noise model for Erdős–Rényi graphs and establishes the equivalence between anonymization time and mixing time under this noise model.

- Section 4.2.2 extends this framework to the stochastic block model (SBM) and demonstrates cases where (partial) anonymization occurs before full mixing, provided that latent cluster structures exist.
- Sections 4.3 and 4.6 present experimental validation using simulated and real data.
- Section 4.4 discusses theoretical implications and outlines avenues for future research.
- Section 4.5 contains formal proofs of key results.

1.3 Notations

We will use the following asymptotic notations: for functions $f, g : \mathbb{Z}^+ \to \mathbb{R}^+$

- $f(n) = o(g(n))$, written as $f \ll g$, if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$;
- $f(n) = \omega(g(n))$, written as $f \gg g$, if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$;
- $f(n) = O(g(n))$, written as $f \lesssim g$, if $\exists C > 0$ and n_0 such that $\forall n \geq n_0$, $f(n) \leq Cg(n)$;
- $f(n) = \Omega(g(n))$, written as $f \gtrsim g$, if $g(n) = O(f(n))$;
- $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $g(n) = O(f(n))$.

When the context is clear, for $j \in \mathbb{Z}^+$, we use $\binom{n}{j}$ to denote the binomial coefficient if n is an integer, and $\binom{S}{j}$ to denote the collection of all subsets with j elements if S is a set. For a given matrix M, the decomposition

$$M = \begin{array}{c} \\ l \\ m \end{array} \begin{array}{c} \begin{array}{cc} j & k \end{array} \\ \begin{pmatrix} M^{11} & M^{12} \\ M^{21} & M^{22} \end{pmatrix} \end{array}$$

divides M into four blocks where the numbers on the borders denote the corresponding dimensions. For example, M^{11} is the upper-left $l \times j$ block of the matrix M. For a real number $\mathbf{r}$ and positive integers m and n, $\mathbf{r}_m$ denotes the m-dimensional all $\mathbf{r}$ vector and $\mathbf{r}_{m\times n}$ denotes the $m \times n$ matrix with all entries equal to $\mathbf{r}$. For a square matrix M, $\mathrm{tr}(M)$ is the trace of M defined as the sum of the diagonal entries of M. $\lfloor \cdot \rfloor$ denotes the floor function, defined as $\lfloor x \rfloor = n \in \mathbb{Z}$ such that $n \leq x$ and $n+1 > x$.

Since graphs and their adjacency matrices convey equivalent information, throughout this sequel we occasionally abuse notation by referring to a graph and its adjacency matrix interchangeably using the same notation.

Chapter 2: Clustered Graph Matching for Label Recovery and Graph Classification

2.1 Introduction

The graph matching literature is recently divided into (at least) two distinct branches: algorithmic development and theoretic graph de-anonymization (with notable cross-over work tackling provable algorithmic de-anonymization; see for example [13, 40]). In the graph de-anonymization literature, a latent alignment across vertex sets is posited and the question of whether an oracle graph matching algorithm can recover this alignment under various noise models is tackled. Recent work in this area has focused on establishing phase transitions for graph de-anonymization in terms of the error level in correlated Erdős-Rényi models [30, 31, 57, 76, 132], in the correlated SBM model [73, 91, 97], and in more general correlated edge-independent graph models [75]. In these models, it is often assumed that edges within each network are (conditionally) independent, and that edges across the network pair are independent except that for each $\{i, j\} \in \binom{V}{2}$, A_{ij} and B_{ij} are positively correlated, where A, B denotes the adjacency matrices of the networks, and $\binom{V}{2}$ denotes the set of all unordered 2-tuples of distinct elements of V.

Inspired by the error model in [6] (introduced first in the context of correlated Erdős-Rényi models in [57]), we will work in the following network error model.

Definition 2.1. *Let $Q \in [0,1]^{n \times n}$ be a symmetric matrix. Given $B \in \mathscr{G}_n$, we say that S is a*

Q-errorful observation from B (written $S \sim \mathrm{BF}(B,Q)$ *for S a "bit-flipped" perturbed B) if for each* $\{i,j\} \in \binom{V}{2}$*, we have that*

$$S_{ij} = B_{ij}(1-X_{ij}) + (1-B_{ij})X_{ij},$$

where $X_{ij} = X_{ji} \overset{ind.}{\sim} \mathrm{Bernoulli}(Q_{ij})$*. Note that we do not allow for self-loops in B or S so the diagonal elements of Q are not used in this construction. When Q is the constant matrix with entries identically equal to q, we write* $S \sim \mathrm{BF}(B,q)$ *in lieu of* $S \sim \mathrm{BF}(B,Q)$*.*

This model makes no a priori assumptions on the underlying distribution of A, which allows for de-anonymization criteria to be established in dependent-edge network settings (i.e., in settings where edges within a network are not (conditionally) independent); see [6] for detail.

Remark 2.1. *Note that in the sequel, we will be considering "bit-flipped" perturbed graphs* $S \sim \mathrm{BF}(B,Q)$ *where B is a Erdős-Rényi random graph with parameter p (abbreviated* $B \sim \mathrm{ER}(n,p)$*); i.e., if each edge is present in B with probability p independent of the presence or absence of all other edges. For further connection of our "bit-flipped" model to the graph de-anonymization phase transition work of [31, 132] in Erdős-Rényi graphs, see Section 2.7.2.*

The inference task we consider herein is a hybridization of graph matching and graph classification. Classification tasks on networks consist of two main sub-categories: node classification and graph classification. Node or vertex classification considers labels at the level of vertices in the network, and seeks to use the information from a priori labeled vertices in the network to classify vertices whose label is initially unknown; note that label classification can occur within a single network or across vertices of a collection of networks. Graph classification considers a class label at the graph level, and seeks to use

the information from an a priori labeled collection of networks to classify networks whose label is initially unknown; note that graph classification must occur in the setting of multiple observed networks. One popular method for graph-level classification is to use graph kernels to measure the similarity of graphs and then define a classifier on the similarity matrices, see [23], [89], [106]. Traditional classifiers on vectorized graphs are also equipped with regularizations that enforce some network structure [100, 125, 127]. Deep learning based classifiers are also popular, especially with the growing interest in neural networks; for example [36, 88]. Another common approach is to find a proper embedding of the graph (e.g., spectral embedding) and then build a classifier for the graphs in the embedding space; e.g., perform a hierarchical clustering via a proper metric [103].

2.1.1 Shuffled Graph Classification

The authors in [124] consider the shuffled graph classification problem, which is the task of classifying graphs at the graph-level. They note that when the vertex correspondences are fully observed across each pair in a collection of networks, then classical classification methods can be used to classify graphs with unknown class types (e.g., a straight-forward classification algorithm can be implemented by choosing a suitable metric across labeled graphs and considering either the Bayes plug-in classifier or the k-nearest neighbor classifier). However, the paper points out that usually the assumption of fully labeled vertices is unrealistic. Rather, sets of shuffled graphs—which are labeled graphs with unknown labeling functions, or unlabeled graphs—should be considered instead. Under this setting, one approach is to apply a graph matching algorithm to reconcile the vertex label uncertainties, after which classical classification algorithms can be employed.

Before defining our task further, we first remark that a pair of graphs A and B in $\mathscr{G}_n$ are said to be *vertex-aligned* if the identity permutation is a priori known to be the true

alignment across the vertex sets of the two graphs. We note here that the notions of vertex-aligned and graph matched are subtly different. Vertex-aligned graphs have a true, known correspondence across their vertex sets. This alignment is often dictated by known vertex-labels or features in the network, or is provided by a subject matter expert in real data scenarios. This true alignment is *not* necessarily the optimal alignment for the graph matching problem. This is often the case in real data networks, where the behavior of vertices across networks is not correlated as strongly as in our models (see, e.g., [75]).

Inspired by the work above, we then consider the following shuffled graph classification problem setup. Consider a collection of m vertex-aligned graphs of k different classes/types (heretofore called the "in-sample" networks), where we model the vertex-alignment across each pair as being known a priori. Note that we will consider these in-sample networks as being graphs on a common vertex set. While this could be relaxed to allow for partial alignment, the main results are analogous, and for the sake of readability, we do not pursue this further herein. Note that if we assume that the graph class labels are initially unknown, we can estimate the class memberships of the in-sample networks via graph-level clustering. We can then use these estimated class labels in our classification procedure. Given an additional ("out-of-sample") graph with both unknown type (assumed to be one of the k represented in the initial collection of m graphs) and unknown vertex correspondence to the collection of m networks, how would we best (i) recover the vertex correspondences between the collection of in-sample networks and the out-of-sample network and (ii) classify its graph type? Note that while we assume all graphs have the same vertex count (denoted n here), this can be relaxed easily in our graph matching framework via strategic network padding; see [44].

This is an important problem in the area of data fusion, in which two samples might come from different data sources. Ideally we would want to utilize all of the existing data/information (including the vertex and graph labels) in subsequent inference, and al-

gorithms that require known vertex correspondences would require the label correspondences to be resolved across samples (e.g., tensor factorization [61], joint graph embedding [5, 67, 87], network regression [138], paired graph testing [115], etc.). While often we can anticipate data coming from the same source to be already matched (i.e., vertex-aligned), such assumption often would not carry over different sources.

The main contributions of this chapter are as follows: We provide a novel exploration of the problem of matching a label-shuffled graph to a collection of vertex-aligned networks (Section 2.2). We provide approaches for matching the shuffled graph to the matched collection at three levels of granularity: matching to a coarse average (Section 2.3), to a clustered average (Section 2.4), and to each graph individually. Throughout, we provide both theory and illustrative experiments showing the benefits/costs of matching at each level of granularity based on the latent structure of the a priori matched collection, with an emphasis on the benefit of clustered matching if the clusters in the matched collection are sufficiently different.

2.2 Clustered graph matching for classification

Before formally defining our graph matching setup, we first note that all graphs/parameters considered herein are implicitly indexed by n; so that the background graphs $B^{(i)}$ are graph sequence $\{B_n^{(i)}\}_n$, permutations P are permutation sequence $\{P_n\}_n$, with model parameters $m = m_n$, $k = k_n$, $\xi = \xi_n$, $p = p_n$, etc., all varying in n. In the sequel, we suppress the n-index moving forward to ease notation.

Formally, the problem we consider is defined as follows. Suppose $B^{(1)}, B^{(2)}, \ldots, B^{(k)} \in \mathscr{G}_n$ denote k vertex-aligned, unobserved graphs; each of them represents a distinct graph type/class (in the classification framework). We will consider both settings in which these background graphs are assumed to be latent and fixed, or in which they are assumed to be

latent graph-valued random variables. In the latter case, we will condition on the $B^{(j)}$'s before generating the subsequent $S_i^{(j)}$'s below. For each $j \in [k] := \{1,2,3,\cdots,k\}$ let $m_j \in \mathbb{N}$ be such that $\sum_j m_j = m$, and consider $S_i^{(j)} \sim \mathrm{BF}(B^{(j)}, p_j)$ for $0 < p_j < 1/2$, $i = 1,2,\ldots,m_j$, and further assume that the collection of graphs $\{\{S_i^{(j)}\}_{i=1}^{m_j}\}_{j=1}^{k}$ are conditionally independent given $B^{(1)}, B^{(2)}, \ldots, B^{(k)}$. The assumption that $p_j < 1/2$ is justified as follows. If $p_j = 1/2$ for some j, then $S_i^{(j)} \overset{i.i.d.}{\sim} \mathrm{ER}(n, 1/2)$, and they carry no information on $B^{(j)}$. If $p_j > 1/2$, then $\mathrm{BF}(B^{(j)}, p_j) \overset{\mathscr{L}}{=} \mathrm{BF}(\bar{B}^{(j)}, 1-p_j)$, where $\bar{B}^{(j)}$ is the complement graph of $B^{(j)}$. Thus, by replacing B_j with its complement we can reduce to the case where $p_j < 1/2$.

For each $j \in [k]$, the graphs in $\{S_i^{(j)}\}_{i=1}^{m_j}$ represent the observed in-sample networks of type j, which can be thought of as edge-noisy, vertex-aligned versions of the background graph $B^{(j)}$. Consider further a fixed $r \in [k]$ and further simulate $A \sim \mathrm{BF}(B^{(r)}, p_r)$ independent (conditionally given the $B^{(1)}, B^{(2)}, \ldots, B^{(k)}$) of all $\{S_i^{(j)}\}$ where $0 < p_r < 1/2$ are fixed; letting P^* be a fixed but unknown permutation in Π_n, we observe $R = (P^*)^T A P^*$, which here represents the out-of-sample, label-obfuscated graph. Note that we assume the class memberships j are known for the in-sample networks.

Our task then is as follows: given the collection of vertex-aligned networks $\{S_i^{(j)}\}$, we seek to recover both the vertex alignment (here P^*) and the graph label (here h) of R. Matching R to $\{S_i^{(j)}\}$ to recover the correct vertex alignment of R can here proceed at (at least) three levels of granularity:

i. (Coarse matching) Define the global average matrix C by $C = \frac{1}{m}\sum_{i,j} S_i^{(j)}$; note each entry of C is in the interval $[0,1]$. We can match R to C to recover the labels of R.

ii. (Clustered matching) Compute the class-level graph means: Let $\ell \in [k]$, and let $\mathscr{C}_\ell$ be the set of graphs in class ℓ, define

$$C_\ell := \frac{1}{|\mathscr{C}_\ell|} \sum_{S_i^{(j)} \in \mathscr{C}_\ell} S_i^{(j)}.$$

Match R to each C_ℓ, computing $\Delta_\ell = \min_{P\in\Pi_n} \|C_\ell - PRP^T\|_F$. Letting $\ell^* \in \text{argmin}_\ell \Delta_\ell$, classify R as type ℓ^* and label R via $P_{\ell^*} \in \text{argmin}_{P\in\Pi_n} \|C_{\ell^*} - PRP^T\|_F$. Note that if the class labels are initially unobserved for the in-sample graphs, we can obtain estimated labels via clustering the $S_i^{(j)}$'s into k clusters, and then use these cluster assignments as class labels for the above procedure.

iii. (Fine matching) Match R to each $S_i^{(j)}$, computing

$$\Delta_{ij} = \min_{P\in\Pi_n} \|S_i^{(j)} - PRP^T\|_F.$$

Letting $\{i^* j^*\} \in \text{argmin}_{ij} \Delta_{ij}$, label R via $P_{\{i^* j^*\}} \in \text{argmin}_{P\in\Pi_n} \|S_{i^*}^{(j^*)} - PRP^T\|_F$.

While we suspect (and empirically it is often the case; see Section 2.5.3) that the clustered matching strategy would yield the highest fidelity recovery of P^* (i.e., of the permutation that unshuffles R), this is not always the case. Indeed, the data smoothing obtained via cluster/class averaging can yield worse matchings if there is sufficient variability/bias across the elements being averaged, in which case the fine matching may yield higher fidelity results. While this is an important issue to untangle, we do not pursue this further here as in our simulations and experiments, clustered averaging yields the best (or close to the best) results.

There is a further computational advantage to clustered matching, as it only requires computing k matchings. In settings where m and n are large, computing all pairwise matchings can be prohibitively expensive. At the other extreme, while coarse matching is computationally less expensive, if there is significant structural differences across $B^{(1)}, B^{(2)}, \ldots, B^{(k)}$, then it is natural to expect the signal of the true cluster to be whitened out in C, and matching R to C will not recover P^*. We shall demonstrate below that the clustered matching balances computational feasibility and within-class signal fidelity to

produce an accurate, more scalable estimate of P^*. Moreover, the clustered matching alone is able to solve both aspects of our inference task simultaneously, both matching and classifying R in one step.

2.3 The good and the bad of coarse matching

In this section, we explore both the potential benefits and potential problems associated with the coarse matching strategy. We consider first the case where $k = 2$; i.e., where we have two distinct asymmetric (i.e., $PB^{(i)}P^T \neq B^{(i)}$ for all $P \neq I_n$) background graphs $B^{(1)}$ and $B^{(2)}$. Suppose further

$$\{I_n\} \notin \text{argmin}_{P \in \Pi_n} \|B^{(1)} - PB^{(2)}P^T\|_F. \tag{2.1}$$

Note that if Eq. 2.1 did not hold, then (under modest assumptions) coarse matching would be successful in unshuffling R with high probability. Eq. 2.1 is necessary for us to explore the break-down point when coarse matching may fail and clustered matching succeed. We note here that in this section $B^{(1)}$ and $B^{(2)}$ are still modeled as vertex-aligned in that the true underlying permutation between graphs is still the identity matrix. The setting in this section captures the often-true reality that the true underlying permutation (according to the assigned data labels) is not Graph Matching optimal.

Without loss of generality, let $A \sim \text{BF}(B^{(1)}, p_1)$ for $p_1 \in (0, 1/2)$, so that $R = (P^*)^T AP^*$ is our out-of-sample network and P^* is the correct permutation that unshuffles R. Here, matching R to C amounts to trying to find P^* by solving the following quadratic assignment problem (where, to ease notation, $f(P) := \sum_{ij} \text{tr}(S_i^{(j)} PRP^T)$):

$$\min_P \|C - PRP^T\|_F \Leftrightarrow \max_P \sum_{ij} \text{tr}(S_i^{(j)} PRP^T) \Leftrightarrow \max_P f(P)$$

Letting $\mathbb{E}_B(\cdot) = \mathbb{E}(\cdot|B^{(1)}, B^{(2)})$, if $\bar{B}^{(1)} \in \mathscr{G}_n$ (resp., $\bar{B}^{(2)}$) denotes the complement graph of $B^{(1)}$ (resp., $B^{(2)}$) we have, where $P^\star := P(P^*)^T$ to ease notation,

$$\begin{aligned}
\mathbb{E}_B(\mathrm{tr}(S_i^{(j)} PRP^T)) &= (1-p_j)(1-p_1)\mathrm{tr}(B^{(j)} P^\star B^{(1)} (P^\star)^T) \\
&\quad + p_j(1-p_1)\mathrm{tr}(\bar{B}^{(j)} P^\star B^{(1)} (P^\star)^T) \\
&\quad + (1-p_j)p_1\mathrm{tr}(B^{(j)} P^\star \bar{B}^{(1)} (P^\star)^T) \\
&\quad + p_j p_1 \mathrm{tr}(\bar{B}^{(j)} P^\star \bar{B}^{(1)} (P^\star)^T) \qquad (2.2) \\
&= (1-2p_j)(1-2p_1)\mathrm{tr}(B^{(j)} P^\star B^{(1)} (P^\star)^T) + T(B^{(1)}, B^{(j)}, n).
\end{aligned}$$

where $T(B^{(1)}, B^{(j)}, n)$ is independent of P and P^* (see Section 2.7.6).

2.3.1 The benefits of averaging

Assume for the moment that $P^\star := P(P^*)^T$ shuffles exactly ξ labels, and that $\mathbb{E}_B(f(P) - f(P^*)) < 0$, which implies that P^* is better than P for matching R to C, on average. This condition ensures that there are enough "good" matches to A (i.e., those from the same background) in the in-sample set to mitigate the effect of averaging the entire collection of m networks, as those from $B^{(2)}$ will, with high probability, not match correctly to A.

Proposition 2.1. *With notation as above, if*

$$-\mathbb{E}_B(f(P) - f(P^*)) = \omega(m\xi\sqrt{n\log n}), \qquad (2.3)$$

holds for all $\xi \in \{2, 3, \cdots, n\}$ and all P such that $P(P^*)^T \in \Pi_{n,\xi}$ (where $\Pi_{n,\xi}$ is the set of permutations shuffling exactly ξ labels), then

$$\mathbb{P}(\{P^*\} \notin \mathit{argmin}_P \|C - PRP^T\|_F) = e^{-\omega(\log n)}$$

The proof of this proposition combines McDiarmid's inequality with a standard union over such P and ξ; see Section 2.7.6 for the derivation). Note that this union bound combined with McDiarmid or similar concentration bounds is a standard argument in the literature, appearing in multiple other graph matching works (see, for example, [74, 75, 111] among others). Lastly, as an example of the feasibility of Eq. 2.3, note that if the background graphs $B^{(i)} \sim \text{ER}(n, q_i)$, then under mild assumptions $-\mathbb{E}_B(f(P) - f(P^*)) = \Theta(m\xi n) \in \omega(m\xi\sqrt{n\log n})$, and Eq. 2.3 holds.

2.3.2 The cost of averaging

The case where coarse averaging is detrimental to matchability is a bit more nuanced. Assume that there exists a P such that $P(P^*)^T \in \Pi_{n,\xi}$, and $\mathbb{E}_B(f(P)-f(P^*)) > 0$. This is tantamount to the noise contributed by the class 2 graphs obfuscating the alignment signal present in the in-sample class 1 graphs. Indeed, the optimal graph matching permutation between a class 1 and class 2 graph will, with high probability, not be the true latent (in the case of the out-of-sample graph) or observed (in the case of in-sample graphs) alignment.

To see the effect of averaging with this noise, we first define for each $x \in \{0,1\}^4$, the following quantity, which captures the edge/non-edge patterns in the graphs before and after shuffling,

$$N_x := \left| \left\{ \{h,\ell\} \in \binom{V}{2} \text{ s.t. } \left(B^{(1)}[\sigma(h),\sigma(l)], B^{(1)}[h,l], B^{(2)}[\sigma(h),\sigma(l)], B^{(2)}[h,l] \right) = x \right\} \right|.$$

We then have the following theorem (see Section 2.7.3 for the proof using Stein's method).

Theorem 2.1. *Under the setup as above, let $p_1 = p_2 = p$ for fixed $p \in (0,1/2)$. If any of the following conditions hold*

i. $|m_1 - m_2| = o(m)$ *and* $N_{1110} + N_{0001} = \omega((n\xi)^{2/3})$;

ii. $m_1, m_2 = \Theta(m)$, $|m_1 - m_2| = \Theta(m)$ *and* $N_{1110} + N_{0001} + N_{1001} + N_{0110} = \omega((n\xi)^{2/3})$;

iii. $m_2/m_1 = \omega(1)$ *and* $N_{1110} + N_{0001} + N_{1001} + N_{0110} = \omega((n\xi)^{2/3})$;

iv. $\frac{n\xi}{m^3} = \omega(1)$,

then we have that

$$\frac{f(P) - f(P^*) - \mathbb{E}_B(f(P) - f(P^*))}{\sqrt{Var_B(f(P) - f(P^*))}}$$

converges in law to a standard normal random variable with

$$Var_B(f(P) - f(P^*)) = O(n\xi m^2).$$

The conditions in Theorem 2.1 ensure that $B^{(1)}$ and $B^{(2)}$ have sufficiently many edgewise structural differences post-shuffling to provide an adequate sample size for Stein's normal approximation method to provide approximate normality of $f(P) - f(P^*)$, as well as sufficient variance growth for $f(P) - f(P^*)$ which will be used later to provide sharp concentration of this difference. We suspect these precise conditions are not necessary, and can be relaxed with more careful analysis of the mismatch between $B^{(1)}$ and $B^{(2)}$, though we do not pursue this further here.

As an immediate consequence of Theorem 2.1, we have the following corollary, which shows that the incorrect permutation P is a better solution of the quadratic assignment problem.

Corollary 2.1. *Given the conditions of Theorem 2.1, if* $\mathbb{E}_B(f(P) - f(P^*)) > 0$ *we have the following:*

i. With no further assumptions on $\mathbb{E}_B(f(P) - f(P^*))$*, we have that* $\mathbb{P}(f(P) > f(P^*)) \geq 1/2(1 - o(1))$.

ii. If we assume that $\mathbb{E}_B(f(P) - f(P^*)) = \omega(m\sqrt{n\xi \log n})$, *we have that* $\mathbb{P}(f(P) > f(P^*)) \geq 1 - o(1)$.

Note that in the case where $\mathbb{E}_B(f(P) - f(P^*)) < 0$ for every $P \neq P^*$, if we do not provide an associated growth rate, then the same proof as in Theorem 2.1 yields $\mathbb{P}(f(P) > f(P^*)) \leq 1/2(1 - o(1))$. The growth rate assumption in Eq. 2.3 is made to uniformly bound these probabilities close to 0.

Remark 2.2. *Sections 2.3.1 and 2.3.2 imply that the key for correctly recovering the latent vertex alignment for the out-of-sample graph is*

$$\frac{tr(B^{(2)}B^{(1)}) - tr(B^{(2)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T)}{tr(B^{(1)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T) - tr(B^{(1)}B^{(1)})} < \frac{m_1(1-2p_1)}{m_2(1-2p_2)}$$

This is akin to a signal–to–noise ratio bound, so that coarse matching is successful if the noise contributed by the class 2 graphs is comparatively small. Note that there is a gap in the growth rates of Eq. 2.3 and Corollary 2.1 used to ensure coarse matching will/will not fail with high probability. While we suspect a sharp phase transition is present, we do not pursue this further herein.

2.3.3 Matching when k greater than 2

We next consider cases where $k > 2$; i.e., where we have multiple distinct backgrounds $B^{(1)}, B^{(2)}, \ldots, B^{(k)}$. Further suppose that for all $j = 2, 3, \ldots, k$,

$$\{I_n\} \notin \text{argmin}_{P \in \Pi_n} \|B^{(1)} - PB^{(j)}P^T\|_F.$$

Without loss of generality, let $A \sim \text{BF}(B^{(1)}, p_1)$, so that we observe $R = (P^*)^T A P^*$. In the $k = 2$ case, we saw that the noise contributed by the graphs not from the background class of A (i.e., the one satisfying Eq. 2.1) could overwhelm the signal provided by the graphs

from the same background class as A. When $k > 2$, the effect of this noise can be more nuanced.

In one direction, note that the same McDiarmid's inequality argument as in the $k = 2$ case yields that if $-\mathbb{E}_B(f(P) - f(P^*))$ is sufficiently big for all $P \neq P^*$, then, with high probability, matching R to C will yield the correct alignment.

In the other direction, if there exists a P such that $P(P^*)^T$ shuffles ξ vertex labels and $\mathbb{E}_B(f(P) - f(P^*)) > 0$, then we have the following result (which is an immediate corollary of the analogue of Theorem 2.1 in the present setting).

Corollary 2.2. *Under the setup as above with $p_i = p$ for all $i \in [k]$, if $n\xi/m^3 = \omega(1)$ and $\mathbb{E}_B(f(P) - f(P^*)) > 0$, then*

i. with no further assumptions on $\mathbb{E}_B(f(P) - f(P^*))$, we have that $\mathbb{P}(f(P) > f(P^*)) \geq 1/2(1 - o(1))$.

ii. if we assume that

$$\mathbb{E}_B(f(P) - f(P^*)) = \omega(m\sqrt{n\xi \log n}),$$

we have that $\mathbb{P}(f(P) > f(P^*)) \geq 1 - o(1)$.

As in the $k = 2$ case, the behavior hinges on $\mathbb{E}_B(f(P) - f(P^*))$, which can be more nuanced in the $k > 2$ setting, as the following example illuminates.

For each $i = 1, 2, 3$, let $B^{(i)} \overset{ind.}{\sim} \text{SBM}(3n, [n, n, n], \Lambda^{(i)})$, so that for each i, the $3n$ vertices in $B^{(i)}$ are divided into three communities, each of size n. Next, sample independent $S_h^{(i)} \sim \text{BF}(B^{(i)}, p_i)$ and let $A \sim \text{BF}(B^{(1)}, p)$. Then,

$$\mathbb{E}\text{tr}(P^T APC) = \frac{m_1}{m}\mathbb{E}\text{tr}(P^T APS_1^{(1)}) + \frac{m_2}{m}\mathbb{E}\text{tr}(P^T APS_1^{(2)}) + \frac{m_3}{m}\mathbb{E}\text{tr}(P^T APS_1^{(3)}).$$

Let $b_i : V \mapsto \{1, 2, 3\}$ denote the community membership function (so that $b_i(v) = j$ if

vertex v is in community j), and assume that $b_1 = b_2 = b_3$ with

$$b_i(v) = 1 + \mathbb{1}\{n+1 \leq v \leq 2n\} + 2 * \mathbb{1}\{2n+1 \leq v \leq 3n\}.$$

For each i, $\Lambda^{(i)} \in [0,1]^{3\times 3}$ is a symmetric 3×3 matrix such that for each $\{u,v\} \in \binom{V}{2}$, (where E_i is the set of edges of $B^{(i)}$), $\mathbb{1}\{\{u,v\} \in E_i\} \overset{ind.}{\sim} \text{Bernoulli}(\Lambda^{(i)}[b_i(u), b_i(v)])$.

Let $p_1 = p$, $p_2 = p_3 = q$, and let $a > r > 0$, and $\varepsilon > 0$. Define $\Lambda^{(i)}$ as follows. Each $\Lambda^{(i)}$ has all entries identically equal to r except that $\Lambda^{(1)}[1,1] = a$, $\Lambda^{(2)}[2,2] = a + \varepsilon$, and $\Lambda^{(3)}[3,4] = a + \varepsilon$. To demonstrate the complications of averaging multiple backgrounds, consider for example (among other similar choices) $a = 0.3$, $\varepsilon = 0.5$; $r = 0.1$, $p = 0.4$, $q = 0.1$. Let P be any fixed permutation that flips all the vertices between blocks 1 and 2. When $m_2 = 2m_1$ and $m_3 = 0$,

$$\mathbb{E}\text{tr}(AC) = c_1 n^2 (1 - o(1)); \; \mathbb{E}\text{tr}(P^T APC) = c_2 n^2 (1 - o(1));$$

and when $m_2 = m_1 = m_3$,

$$\mathbb{E}\text{tr}(AC) = c_1 n^2 (1 - o(1)); \; \mathbb{E}\text{tr}(P^T APC) = c_3 n^2 (1 - o(1)).$$

where $1 < c_3 < c_1 < c_2 < 2$ are constants that can be obtained from direct mathematical computation. As $\text{tr}(AC)$ and $\text{tr}(P^T APC)$ concentrate tightly about their means, we see that for sufficiently large n, flipping blocks 1 and 2 via P (an optimal alignment of $\Lambda^{(1)}$ and $\Lambda^{(2)}$) when $m_2 = 2m_1$ and $m_3 = 0$ will, with high probability, result in a better match for the average than the true identity alignment. This is unsurprising, as $\Lambda^{(2)}$ is designed for this end; i.e., to attract the dense block in $\Lambda^{(1)}$ to block 2 in $\Lambda^{(2)}$. If, however, the wrong-class in-sample graphs are evenly split between classes 2 and 3 with m_1 from each of the three classes, then the alignment provided by P is no longer better than the identity alignment

(again with high probability). Noting the same analysis holds for flipping blocks 1 and 3 (an optimal alignment of $\Lambda^{(1)}$ and $\Lambda^{(3)}$), we see here that the noise from the in-sample, wrong-class networks effectively cancels across classes as the wrong-classes pushing the optimal permutation in different, counteracting directions.

It is clear that if all the $P^{(i)}$'s that optimally align $B^{(1)}$ and $B^{(i)}$ are equal (or overlap significantly), then the noise cancellation demonstrated in the example above will not occur. In the SBM setting, this can be achieved by ensuring that the optimal alignment of $\Lambda^{(i)}$ to $\Lambda^{(j)}$ is the identity mapping for $i, j \neq 1$. We next seek to generalize this idea to other network models. To this end, we consider the following multiple random dot product graph model from [5].

Definition 2.2. *Let U be an $n \times d$ matrix with orthonormal columns, and for $j \in [n]$, let U_j denote the j-th row of U. Let $R^{(1)}, \ldots, R^{(k)}$ be $d \times d$ symmetric matrices such that $0 \leq U_j R^{(i)} U_h^T \leq 1$ for all $j, h \in [n]$, $i \in [k]$. We say that the random adjacency matrices $B^{(1)}, \ldots, B^{(k)}$ are jointly distributed according to the common subspace independent-edge graph (COSIE) model with rank d and parameters U and $R^{(1)}, \ldots, R^{(k)}$ if given U and $\{R^{(i)}\}_{i=1}^k$, the collection of networks $\{B^{(i)}\}_{i=1}^k$ is independent, and for each $i \in [k]$, the upper-triangular entries of $B^{(i)}$ are independent and distributed according to*

$$\mathbb{P}(B^{(i)} \mid U, R^{(i)}) = \prod_{j<h} \left(U_j R^{(i)} U_h^T \right)^{B^{(i)}[j,h]} \left(1 - U_j R^{(i)} U_h^T \right)^{1-B^{(i)}[j,h]}$$

The COSIE model of Definition 2.2 provides a flexible framework for modeling a collection of networks on a common vertex set, and it encompasses many important network models including the multilayer stochastic blockmodel of [53]. The score matrices $R^{(i)}$ in the COSIE model allow us a similar opportunity as in the SBM setting to ensure that the wrong-class, in-sample graphs are all misaligned in synchrony. We shall now demonstrate this in the following example.

Assume that $B^{(1)},\ldots,B^{(k)}$ are jointly distributed according to the COSIE model with rank d and parameters U, $R^{(1)},\ldots,R^{(k)}$, and assume further that the $R^{(j)}$'s are diagonal matrices for all j (this is similar to the model considered in [34, 128]). Suppose further that the diagonal of $R^{(1)}$ are ordered to be non-decreasing, and that there exists a common $Q \in \Pi_d \setminus \{I_d\}$ such that for all $j \in [k] \setminus \{1\}$,

$$Q \in \operatorname{argmin}_{P\in\Pi_d} \|R^{(1)} - PR^{(j)}P^T\|_F,$$
$$I_d \notin \operatorname{argmin}_{P\in\Pi_d} \|R^{(1)} - PR^{(j)}P^T\|_F.$$

The following lemma, proven in Section 2.7.5, will codify sufficient conditions under which wrong-class in-sample graphs are all misaligned in synchrony.

Lemma 2.1. *With setup as above, if there exists a permutation $P \in \Pi_n$ such that for all $j \neq 1$,*

$$1 - 2\|U^TPU - Q\|_F > \frac{\operatorname{tr}(R^{(1)}I_dR^{(j)}I_d^T)}{\operatorname{tr}(R^{(1)}QR^{(j)}Q^T)}, \tag{2.4}$$

then $\forall\, j \neq 1$, $\operatorname{tr}(P^T\mathbb{E}(B^{(1)})P\mathbb{E}(B^{(j)}) > \operatorname{tr}(\mathbb{E}(B^{(1)})\mathbb{E}(B^{(j)}))$.

The technical Lemma condition (Eq. 2.4) is used to ensure that the action of shuffling $\mathbb{E}(B)$ by P (and yielding $PUR^{(i)}U^TP^T$) is sufficiently close to the action of shuffling $R^{(i)}$ by Q (and yielding $UQR^{(i)}Q^TU^T$); this is used to then lift the shuffling of the unknown $R^{(i)}$'s to a shuffling of the observed $B^{(i)}$'s.

Next, define

$$\tilde{f}_j(P) := \operatorname{tr}(P^T\mathbb{E}(B^{(1)})P\mathbb{E}(B^{(j)}) - \operatorname{tr}(\mathbb{E}(B^{(1)})\mathbb{E}(B^{(j)})).$$

If P satisfies the conditions in Lemma 2.1 and $\sum_{i\neq 1} m_i$ is sufficiently large relative to m_1,

we have

$$\sum_{i\neq 1} m_i \tilde{f}_i(P) > -m_1 \tilde{f}_1(P). \tag{2.5}$$

Consider now the setting where $p_1 = \cdots = p_k = p$, and let $A \sim \mathrm{BF}(B^{(1)}, p)$ and let $\{S_j^{(i)}\}$ as before. We seek then to match the observed network $R = (P^*)^T A P^*$ with $C = \frac{1}{m}\sum_{i,j} S_j^{(i)}$. Eq. (2.5) ensures that

$$\begin{aligned}
\mathbb{E}\left(\mathrm{tr}\left(P^T A P C\right)\right) &= \mathbb{E}\left(\mathbb{E}_B\left(\mathrm{tr}(P^T A P C)\right)\right) \\
&= \sum_i \frac{m_i(1-2p)^2}{m} \mathrm{tr}(P^T \mathbb{E}(B^{(1)}) P \mathbb{E}(B^{(i)})) \\
&> \sum_i \frac{m_i(1-2p)^2}{m} \mathrm{tr}(\mathbb{E}(B^{(1)})\mathbb{E}(B^{(i)})) \\
&= \mathbb{E}\left(\mathbb{E}_B\left(\mathrm{tr}(AC)\right)\right) = \mathbb{E}\left(\mathrm{tr}\left(AC\right)\right).
\end{aligned}$$

A similar application of Stein's method as in Theorem 2.4 will yield that $\mathrm{tr}(A(PCP^T - C))$ suitably scaled and centered will converge to a standard normal random variable. This will yield the following theorem.

Theorem 2.2. *With assumptions as in Lemma 2.1, assume that $p_i = p$ for some fixed $0 < p < 1/2$ for all $i \in [k]$. Letting P satisfy the conditions of Lemma 2.1, and assume that $\{m_i\}$ is such that Eq. 2.5 holds. If $P(P^*)^T$ shuffles ℓ vertex labels, then $n\ell/m^3 = \omega(1)$ implies that*

i. with no further assumptions on $\mathbb{E}\,\mathrm{tr}(A(PCP^T - C))$, we have that $\mathbb{P}(f(P) > f(P^*)) \geq 1/2(1 - o(1))$.

ii. if we assume that $\mathbb{E}\,\mathrm{tr}(A(PCP^T - C)) = \omega(\sqrt{n\ell \log n})$, we have that $\mathbb{P}(f(P) > f(P^*)) \geq 1 - o(1)$.

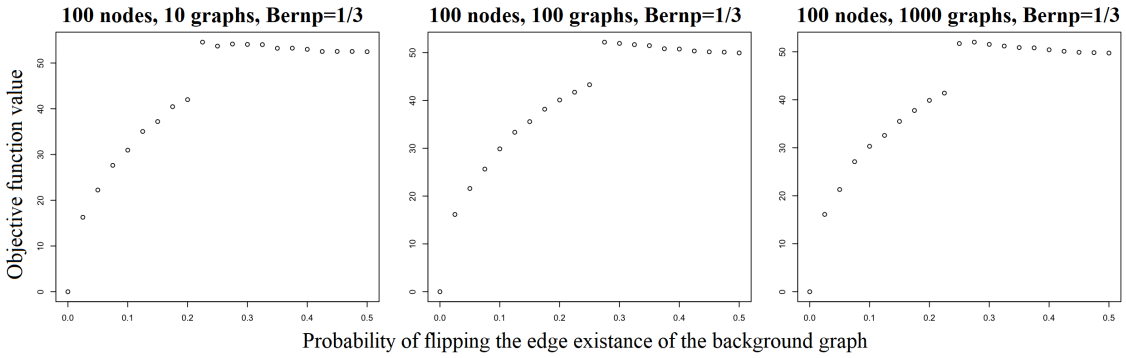

Figure 2.1: With a single background $B \sim \text{ER}(n = 100, 1/3)$, we consider $A, S_i^{(1)} \overset{i.i.d.}{\sim} \text{BF}(B, q)$, and we match A (i.e., $P^* = I_n$) to C using `SGM` with 5 seeds. Varying the number of in-sample graphs ($m = 10$ in the left panels, $m = 100$ in the middle panels, and $m = 1000$ in the right panels), we plot the `SGM` objective function value $f = \|A - PCP^T\|_F$ versus the value of the edge perturbation parameter q, averaged over 10 Monte Carlo iterates.

2.4 Clustered matching

Consider next the case of clustered matching, where for simplicity we will assume the class labels are observed or the clustering perfectly recovers the class labels amongst the in-sample networks $S^{(i)}$. The case in which the clusters are noisily recovered is of great interest, and will be the subject of subsequent work. For each $i \in [k]$, let $C^{(i)}$ be the cluster average of the graphs from class i, so that $C^{(i)} = \frac{1}{m_i} \sum_{j=1}^{m_i} S_j^{(i)}$. With $A \sim \text{BF}(B^{(1)}, p_1)$ as before (recall that we observe the shuffled A, i.e., $R = (P^*)^T A P^*$), recalling the form of $\mathbb{E}_B(\text{tr}(C^{(i)} P R P^T))$ from Eq. (2.2), we have that for $P \neq P^*$ and $p = p_1 = \cdots = p_k$ (note that this equation is shown in Section 2.7.6),

$$\begin{aligned}
&\mathbb{E}_B(\text{tr}(C^{(1)} P^* R (P^*)^T)) - \mathbb{E}_B(\text{tr}(C^{(i)} P R P^T)) \\
&\quad = (1-p)(1-2p)\|B^{(1)}\|_F^2 - p(1-2p)\|B^{(i)}\|_F^2 - (1-2p)^2 \text{tr}(B^{(i)} P (P^*)^T B^{(1)} P^* P^T).
\end{aligned} \tag{2.6}$$

Theorem 2.3. *With notation as above, denote* $X_{i,P} = tr(C^{(1)}P^*R(P^*)^T) - tr(C^{(i)}PRP^T)$. *If for all integer* $2 \leq \xi \leq n$, *and for all* $i \neq 1$, *and* P *s.t.* $P(P^*)^T \in \Pi_{n,\xi}$, *we have Eq. 2.6 is of order* $\omega(n\sqrt{\xi \log(n)})$, *then*

$$\mathbb{P}_B(\exists i \in [k] \setminus \{1\}, P \in \Pi_n \text{ s.t. } X_{i,P} \leq 0) = e^{-\omega(\log(n))}, \tag{2.7}$$

The proof of Theorem 2.3 is a straightforward application of Hoeffding's inequality; see Section 2.7.6 for detail.

Theorem 2.3 implies that with high probability the correct matching of R to $C^{(1)}$ will yield a better objective function value than any other matching of R to any other class mean. Hence, clustered matching can be used to both unshuffle *and* classify R by assigning it to the cluster/class it matches best to (best as in lowest objective function value). As an example, consider the SBM setup of Section 2.3.3, with $\Lambda^{(1)}$ and $\Lambda^{(2)}$ defined as before, and $\Lambda^{(3)}$ set to be $\Lambda^{(2)}$. If $m_1 = m_2 = m_3$, then the results of Section 2.3.3 imply that coarse matching would not recover the true permutation, while Theorem 2.3 implies clustered matching would recover the right permutation with high probability.

n	50	50	50	100	100	100
m	10	100	1000	10	100	1000
$q = 0.200$	1	1	1	1	1	1
$q = 0.225$	0.14	1	1	0.10	1	1
$q = 0.250$	0.40	0.50	0.42	0.12	1	0.19
$q = 0.275$	0.20	0.12	0.36	0.12	0.12	0.12
$q = 0.300$	0.20	0.30	0.12	0.07	0.09	0.07
$q = 0.325$	0.22	0.14	0.20	0.06	0.07	0.11
$q = 0.350$	0.14	0.24	0.14	0.08	0.08	0.08
$q = 0.375$	0.20	0.18	0.10	0.05	0.06	0.06
$q = 0.400$	0.10	0.14	0.10	0.05	0.07	0.10

Table 2.1: Table of matching accuracy in the single Erdős-Rényi background setting with $p = 1/3$, averaged over 10 Monte Carlo iterates; similar results are obtained in the $p = 0.5$ setting; see Section 2.8 for detail.

2.5 Simulations and Real Data Experiments

We will now explore the impact of the three different strategies for matching R to C outlined in Section 2.2, namely coarse matching, clustered matching, and fine matching. Note that in the experiments below, as computing the exact solution of the graph matching problem is often computationally intractable, we rely on the approximate graph matching algorithm, `SGM`, of [44]. This algorithm will use seeded vertices across R and C (those whose alignments via P^* are a priori provided), as this will help us to hone in on when $f(P^*)$ is sub-optimal, which is our chief computational question.

2.5.1 Matching in the ER model

We first consider the effectiveness of the coarse matching strategy in the $k = 1$ setting in a simple Erdős–Rényi model with n nodes and edge probability denoted by p. In the $k = 1$ setting, all in-sample networks are equally informative and averaging them into a

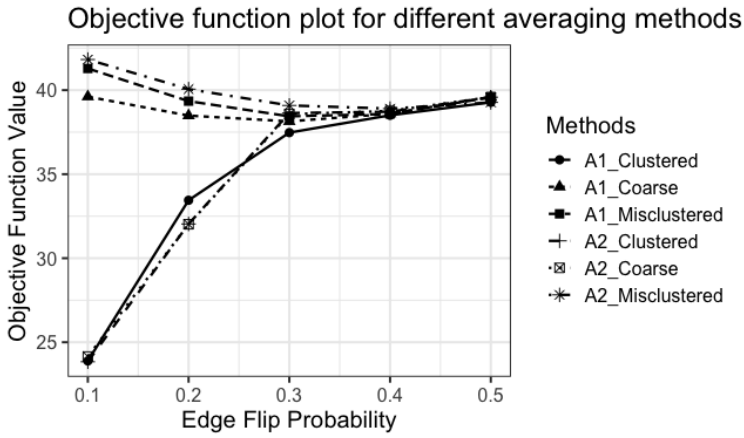

Figure 2.2: Objective function plot for different averaging methods for the two Erdős-Rényi background setting considered in Section 2.5.1. For each of the two out-of-sample networks we perform coarse matching, clustered matching with its own cluster, and cluster matching with the incorrect cluster. We plot the objective function of the match versus q for each matching strategy/out-of-sample graph pair, averaged over 50 Monte Carlo iterates. Note that the A_2-clustered and A_2-coarse point values and subsequent lines are nearly identical, and are hard to distinguish; see Table 2.2.

background C is sensible and recommended as long as the edge flipping probability is not too large. When the Q matrix in Definition 2.1 is close to $1/2$, the in-sample and out-of-sample graphs become closer to independent, though this can be overcome to an extent by considering a large value of m. Formalizing this, we consider $B \sim \text{ER}(n,p)$, and $A, S_i^{(1)} \overset{i.i.d.}{\sim} \text{BF}(B,q)$, and we match A (i.e., $P^* = I_n$) to C using `SGM` with 5 randomly chosen seed vertices. In Figure 2.1 we consider $p = 1/3$ (similar results are obtained with $p = 0.5$, see Section 2.8 and Figure 2.6 for detail), and we consider the effect of varying the number of nodes n ($n = 100$ in the figure, see Section 2.8 for $n = 50$ plots), and the number of in-sample graphs ($m = 10$ in the left panel, $m = 100$ in the middle panel, and $m = 1000$ in the right panel). In each panel, we plot the `SGM` objective function value $f = \|A - PCP^T\|_F$ versus the value of the edge perturbation parameter q. When combined with the information in Table 2.1 (see also Section 2.8 for a full table), we see that for sufficiently small q (here less than 0.2), we will always recover the exact match, and the objective function is steadily increasing. The jump in the objective function scores correspond to the point at which the `SGM` algorithm no longer recovers the true alignment, which is evidence for the true alignment no longer being optimal. While subtle, we do see that this transition point occurs at a larger value of q when n and m generally increase as expected. The nature of the jump, and the relatively flat objective function value post-jump, across all the figures when `SGM` fails is indicative of the presence of phantom alignment strength after this critical threshold; see [43] for further detail.

We next consider the case of two backgrounds $B^{(1)} \sim \text{ER}(n=80, p=0.2)$ and $B^{(2)} \sim \text{ER}(n=80, p=0.4)$. We let $S_1^{(1)}, \ldots, S_{m_1}^{(1)}$ i.i.d. sampled from $\text{BF}(B^{(1)}, q)$ and $S_1^{(2)}, \ldots, S_{m_2}^{(2)}$ i.i.d. sampled from $\text{BF}(B^{(2)}, q)$ where $m_1 = 200$, $m_2 = 2000$ and $0 < q < 0.5$ is the edge flipping probability. We draw two out-of-sample networks $A_i \sim \text{BF}(B^{(i)}, q)$ for $i = 1, 2$ and match them with the full average of all the S's, the average of just the $S^{(1)}$'s and the average of just $S^{(2)}$'s. We plot the objective function of the match versus q in Figure 2.2, and provide

the corresponding matching error rates (i.e., the proportion of labels incorrectly recovered) in Table 2.2; both are averaged over 50 Monte Carlo (MC) iterates.

We see that matching either graph A_i to the coarse average, or the wrong cluster (i.e., matching A_i to the average of $S^{(j)}$'s for $i \neq j$) yields poor matching accuracy and nearly uniformly high objective function value. The exception is matching A_2 to the coarse mean when q is small, due to the large proportion of type-2 graphs in the in-sample data, still enables a high fidelity matching. As expected, matching to the correct in-sample cluster yields both better matching accuracy and better objective function value (compared to the wrong cluster matching), at least for modest values of q. This points to the utility of using the class labels to locally average (or clustering) before matching, as the objective function value of matching to the class means can be used to identify the right class to match to which will then yield higher matching accuracy.

Method	A_i class	q=0.1	q=0.2	q=0.3	q=0.4	q=0.5
Coarse	1	0.080	0.076	0.076	0.076	0.077
Clustered	1	1.000	0.737	0.129	0.084	0.076
Misclustered	1	0.074	0.073	0.076	0.075	0.074
Coarse	2	1.000	1.000	0.170	0.093	0.075
Clustered	2	1.000	0.987	0.199	0.089	0.074
Misclustered	2	0.074	0.075	0.075	0.076	0.074

Table 2.2: Table of matching accuracy in the two Erdős-Rényi background setting, averaged over 50 Monte Carlo iterates. Values are rounded to three decimal places.

2.5.2 Clustered matching in the COSIE model

Our theoretical results in the COSIE model show that when the score matrices are disordered in a similar direction, averaging across samples drawn from multiple backgrounds can produce inferior label recovery in the downstream out-of-sample matching task. If the score matrices are disordered in different enough directions, we expect that the noise in the score matrices could cancel (as in the SBM case of Section 2.3.3), which would result in

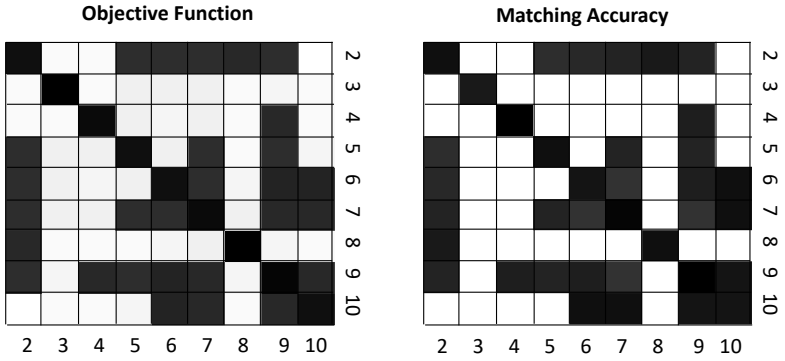

Figure 2.3: In the COSIE model considered in Section 2.5.2, we consider matching A to $C_{a,b}$ for $a \neq b$ ranging over $\{2,\ldots,10\}$. In the left (resp., right) heatmap, we plot the objective function value (resp., matching error rate) obtained from `SGM` with 5 seeds. In both heatmaps, lighter shade denotes smaller values/better matches while darker shade denotes larger values/worse matches; note that the diagonal blocks are not included as we assume $a \neq b$.

strong label recovery in the downstream out-of-sample matching task even when averaging a large number of wrong-cluster in-sample networks.

We further explore this phenomenon in the following simple, yet illustrative experiment. We generate $k = 10$ COSIE background graphs as follows: We consider $k = 10$ independent $G_i \sim \mathrm{ER}(100, 0.5)$ graphs (i.e., uniformly random graphs), and use the procedure in [5] to project these graphs into a common COSIE framework (i.e., finding a common U and $R^{(i)}$'s such that $G_i \approx UR^{(i)}U^T$ and where each $R^{(i)} \in \mathbb{R}^{10\times 10}$). We then sample $B^{(i)} \sim \mathrm{COSIE}(U, R^{(i)})$, and for each $i \in [10]$, we sample m_i i.i.d. networks $S_1^{(i)}, \cdots, S_{m_i}^{(i)}$ from $\mathrm{BF}(B^{(i)}, 0.1)$. We consider $A \sim \mathrm{BF}(B^{(1)}, 0.1)$, and $m_1 = 10$, $m_i = 5$ for $i \neq 1$.

We then consider matching A to $C_{a,b}$ where $C_{a,b}$ is formed via $C_{a,b} = \frac{1}{20}\sum_{i\in\{1,a,b\}}\sum_{j=1}^{m_i} S_j^{(i)}$, and where $a \neq b$ range over $\{2,\ldots,10\}$. We plot a pair of heatmaps in Figure 2.3 with indices representing values of a, b chosen. In the left (resp., right) heatmap, we plot the objective function value (resp., matching error rate) obtained from `SGM` with 5 seeds. In both heatmaps, lighter shade denotes smaller values/better matches while darker shade denotes larger values/worse matches; note that the diagonal blocks are not included as we assume $a \neq b$. From the figure, we see a strong positive correlation between matching error rate

and objective function score, and that which combination of background graphs are being averaged into $C_{a,b}$ is consequential and nuanced. In Section 2.3, we saw that the nature of the backgrounds was crucial for determining whether a coarse matching would produce good results. In this example, similar to the SBM example considered in Section 2.3.3, we consider $k = 3$ and consider coarse matching of $(P^*)^T A P^*$ to C, with the aim of better understanding when the coarse class averaging is beneficial/harmful for label recovery of the shuffled A. To this end, we set $m_1 = m_2 + m_3$, and we consider different combinations of background graphs $B^{(a)}$ and $B^{(b)}$ for representing classes 2 and 3 ($B^{(1)}$ will always represent class 1).

As demonstrated in Theorem 2.2, the wrong combination of in-sample backgrounds can lead to poor performance via coarse matching; this figure suggests that this phenomenon is neither uncommon nor straightforward. Indeed, while some background graph class pairs (e.g., (5,7)) have their order relative to $B^{(1)}$ combine to provide poor matching accuracy and large matching objective function, those same graphs paired differently (e.g., (5,6) and (7,8)) are relatively innocuous when averaged with the $S^{(1)}$'s, as the true alignment is still well-recovered even with coarse matching.

2.5.3 Matching human connectomes

We next consider a real data set of human connectomes from the HNU1 data repository [139]. In the dataset, for each of 30 subjects there are 10 test/retest DTMRI brain scans. The raw scans were processed via NeuroData's MRI Graphs (m2g) pipeline of [58] and registered to the Desikan atlas [33], yielding a 70 vertex weighted graph for each scan. The graphs are a priori vertex-aligned both within and across subjects, with vertices in each graph representing regions of interest in the brain atlas, and with edges measuring the strength of the neuronal connections between regions. The post-processed brain graphs are

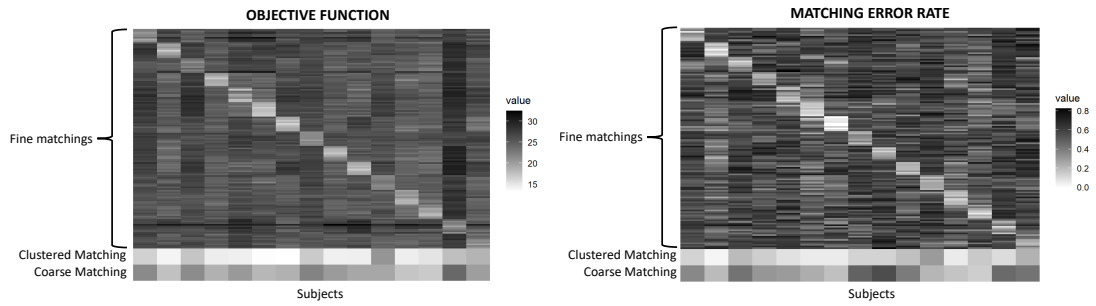

Figure 2.4: For each of the 15 out-of-sample brain networks, we match with: (i) the average of all existing 135 graphs (coarse averaging); (ii) the average of the 9 in-sample graphs from the same subject (clustered averaging); (iii) each of the existing simulated graph (fine averaging). In the top heatmap, we plot the objective function value obtained from `SGM` with 5 seeds, and in the bottom heatmap we plot the matching error rate. In both heatmaps, lighter shade denotes smaller values/better matches while darker shade denotes larger values/worse matches. In each heatmap, the columns correspond to the 15 out-of-sample networks, with the rows corresponding to: top 135 (thinner) rows the fine matching with each in-sample network separately; the second-to-the bottom (thicker) row the clustered matching and the bottom (thicker) row the coarse matching result.

available from `neurodata.io`.

For our experiment, we randomly select 15 different subjects and their corresponding $15 \times 10 = 150$ scans. We perform the experiment as follows: for each individual, we randomly take 9 brain graphs as the existing matched graphs (i.e., in-sample), with 1 brain graph assumed to be the out-of-sample network. These 15 out-of-sample graphs will have both their class labels and vertex alignments (to the 135 in-sample graphs) treated as unknown/hidden in this experiment, with the goal then to recover the hidden class label (i.e., subject label) and vertex alignments for these out-of-sample graphs. To recover the vertex alignments, for each of these 15 out-of-sample networks, we match them with: (i) the average of all 135 in-sample graphs (coarse averaging); (ii) the average of the 9 in-sample graphs from the same subject (clustered averaging); and (iii) each of the in-sample graphs separately (fine averaging). Note that while we used the true class/subject labels in our clustered averaging, these can be readily obtained via a simple k-means procedure applied to an embedded inter-graph distance matrix; see Section 2.8.3 for detail.

We plot heatmaps of the matching objective function and matching error in Figure 2.4. In the top heatmap, we plot the objective function value obtained from `SGM` with 5 seeds, and in the bottom heatmap we plot the matching error rate. In both heatmaps, lighter shade denotes smaller values/better matches while darker shade denotes larger values/worse matches. In each heatmap, the columns correspond to the 15 out-of-sample networks, with the rows corresponding to: the fine matching (top 135 *thinner* rows) with each in-sample network separately; the clustered matching (the second-to-the bottom *thicker* row) and the coarse matching (the bottom row) results. From the figure, we see that for the majority of subjects, the clustered matching yields smaller objective function error and better matching accuracy than coarse matching (the subject in column 11 being the notable exception). Moreover, we see that in some cases the best of the fine matchings yields better matching accuracy than even the clustered matching, though this is not always the case. For example, considering the matching accuracy at differing levels of granularity for a pair of subjects displayed in Table 2.3, we see that for some patients the best fine matching yields the best matching accuracy while for others the clustered matching is best.

We next explore whether clustered averaging can be used to uncover the correct brain class labels as well. This would be a key step for identifying the correct cluster to average to in Figure 2.5. To explore this, for each of the 15 out-of-sample brain networks, we plot a heatmap of the objective function obtained by `SGM` with 5 seeds by matching with each of the 15 in-sample cluster averages. In each heatmap, the columns correspond to the 15 out-of-sample networks, and the rows correspond to the 15 in-sample network averages (the diagonal corresponds to the matched indices). Larger values in the heatmap are denoted by darker colors. We indeed see that across the board, the cluster matching

Subject	Coarse Matching	Clustered Matching	Fine Matching
0025435	0.8286	0.9429	0.8857
0025440	0.6000	0.8143	0.8571

Table 2.3: Matching accuracy for a pair of subjects across levels of granularity.

that obtains the best objective function is the one that matches the out-of-sample brains to the correct in-sample cluster average, pointing again to the validity of using this approach (with high fidelity clusters) for simultaneous classification and label alignment. While we do not suspect these brain graphs follow our posited bit-flipped model, the theory developed for our model nevertheless plays out in this real data setting: the differences among the background connectome classes cause coarse matching to be less effective than clustered matching. This is as predicted by the theory, and clustered matching here provides both a computationally more efficient alternative to fine matching (that can produce better matching results) and an empirically better match than coarse matching.

2.6 Conclusion and discussion

We investigate strategies for recovering the vertex labels of an out-of-sample graph by using the information in a collection of vertex-aligned in-sample graphs. In both theory and synthetic/real data simulations, we explore the effectiveness of recovering the out-of-

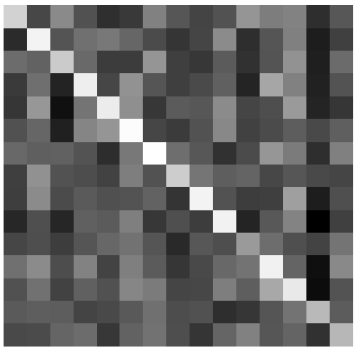

Figure 2.5: For each of the 15 out-of-sample brain networks, we plot a heatmap of the objective function obtained by `SGM` with 5 seeds by matching with each of the 15 in-sample cluster averages. In each heatmap, the columns correspond to the 15 out-of-sample networks, and the rows correspond to the 15 in-of-sample network averages (the diagonal corresponds to the matched indices). Larger values/worse matches in the heatmap are denoted by darker colors, with smaller values/better matches denoted by lighter colors.

sample graph vertex labels by matching it to the in-sample collection at three levels of granularity. While it can be the case that the best method is to match the out-of-sample graph to all individual in-sample graphs and take the labels according to the matching result with smallest loss function, often this is too computationally expensive and the data-smoothing inherent to clustered matching often yields better alignment than the fine-grain matching. At the other end of the granularity spectrum, in both theory and practice we demonstrate that labeling the out-of-sample graph by matching it to the full average of all in-sample graphs can yield poor label recovery, especially in settings where there are significant differences in the structures across the in-sample graphs.

Our proposed matching algorithm is a compromise between these two extremes. Our "clustered matching" involves matching the out-of-sample graph individually to each class's average and labeling it via the matching result with smallest loss function. A consequence of our theory is that given high enough fidelity classes, under mild model conditions the clustered matching will recover the right cluster and the right alignment with high probability. We also used both simulated as well as real world data to demonstrate the validity of the proposed algorithm as well as the advantage of clustered matching compared to the fine-grain and coarse-grain strategies outlined in Section 2.2.

We also proposed the following possible extension and questions. While we consider matching here to the usual sample average, there are a number of different notions of network means we could consider aligning to (e.g., Frechet means [48, 60] or smoothed means [119]). We next seek to relate this work to the phantom alignment strength conjecture proposed by Fishkind et. al. in [43]. In particular, our result in the Erdős-Rényi model simulations (Fig. 2.1) showed matching objective functions similar to the "hockey stick" matchability plots in [43]. Both our work and that in [43] deal with edge-wise correlations, and we are working to unify our results and use our results and computations to support the foundation of the phantom alignment strength conjecture, ideally finding explanation

or causation for the "hockey sticks" matchability plots. We would then be able to propose more precise conditions on when our three aforementioned matching strategies will behave similarly and when they will differ significantly.

Another important issue we want to explore is the edge-wise matchability of the out-of-sample graph. In particular, standing on a single edge level, it is hard to predict if the matching is exact for both clustered matching and fine matching. We want to find conditions or ways to verify if the edge-wise matching is indeed the exact one by looking at the edge mismatch level and finding computationally tractable remedies for misaligned structure. Also, as in [132], we want to explore the information theoretic recovery limitations of clustered versus coarse matching as well.

Finally, it is important to note that if class labels are not known a priori, our proposed clustered matching relies heavily on a good graph clustering algorithm. If a clustering algorithm is provided, then our matching approaches are essentially standard GMP's and can be solved using existing methods and packages.

2.7 Proofs

2.7.1 Notation used throughout the proofs

Throughout the section, we will observe that

$$f(P)-f(P^*)=\sum_{\substack{h,\ell,\ \text{s.t.}\\ \{\sigma(h),\sigma(\ell)\}\neq\{h,\ell\}}}\left(\sum_{i,j}S_i^{(j)}[h,\ell]\right)(A[\sigma(h),\sigma(\ell)]-A[h,\ell]), \tag{2.8}$$

where σ is the permutation associated with $P^{\bigstar}=P(P^*)^T$. This expansion is essential for applying the concentration inequalities appearing throughout the manuscript.

2.7.2 Connection between Definition 2.1 and Erdős-Rényi matchability

The model in Definition 2.1 can be used to study the phenomenon of graph matchability/graph de-anonymization. Loosely stated, if graphs A and B in $\mathscr{G}_n$ have true, but latent, alignment P^*, graph matchability is concerned with understanding the conditions (often in terms of the edge correlation across networks) under which

$$\{P^*\} = \text{argmin}_{P\in\Pi_n}\|A - PBP^T\|_F.$$

(determining whether there is enough signal to match the vertices of two random graphs). Considering $B \sim \text{ER}(n,p)$ in Definition 2.1 and $Q = sJ_n$ (where J_n is the hollow $n\times n$ matrix with all off-diagonal entries identically equal to 1), matchability in the classical correlated Erdős-Rényi model is obtained by considering alignments of S_1 and (a shuffled) S_2, where $S_1, S_2|A \overset{i.i.d.}{\sim} \text{BF}(B,Q)$. Indeed, (no longer conditioning on A) in this case S_1 and S_2 would both have $\text{ER}(n, p(1-s)+s(1-p))$ distributions and the edge-wise correlation is given by

$$\text{corr}(S_{1,ij}, S_{2,ij}) = \frac{p(1-p)(1-2s)^2}{(p+s-2sp)(1-p-s+2sp)}.$$

Here, sharp matchability thresholds are established in [31, 132] in terms of s and p (i.e., in terms of the correlation across networks).

2.7.3 Proof of Theorem 1:

Proof. We will use Stein's method to prove this result; principally Theorem 3.6 in [102]. We say that a collection of random variables $(X_1, \cdots, X_n)$ has *dependency neighborhoods* $N_i \subset [n]$ for $i \in [n]$ if for each i, X_i is independent of $\{X_j \text{ s.t. } j \notin N_i\}$.

Theorem 2.4 (Adapted from Theorem 3.6 in [102])**.** *Let d_K be the Kolmogorov metric, so*

that for random variables X and Y

$$d_K(X,Y) = \sup_{x\in\mathbb{R}} |F_X(x) - F_Y(x)|,$$

where F_X (resp., F_Y) is the distribution function of X (resp., Y). Let $X_1,\ldots,X_n$ be random variables such that for all $i \in [n]$, $\mathbb{E}(X_i^4) < \infty$, $\mathbb{E}(X_i) = 0$, $\sigma^2 = Var(\sum_i X_i)$, and define $W = \sum_i X_i/\sigma$. Let the collection $(X_1,\ldots,X_n)$ have dependency neighborhoods N_i, $i = 1,\ldots,n$, and also define $D := \max_{i\in[n]} |N_i|$. Then for Z a standard normal random variable

$$d_K(W,Z) \leq \sqrt{(2/\pi)^{1/2}\left(\frac{D^2}{\sigma^3}\sum_{i=1}^{n}\mathbb{E}|X_i|^3 + \frac{\sqrt{28}D^{3/2}}{\sqrt{\pi}\sigma^2}\sqrt{\sum_{i=1}^{n}\mathbb{E}(X_i^4)}\right)}$$

Recalling that σ is the permutation associated with $P(P^*)^T$, define

$$Y_{h\ell} := (\sum_{i,j} S_i^{(j)}[h,\ell])(A[\sigma(h),\sigma(\ell)] - A[h,\ell]).$$

Note that the maximum size of the dependency neighborhoods for each of the $(Y_{h\ell})$'s is at most 2 (i.e., D in Theorem 2.4 is 2). Let

$$\alpha_{hl} = \sum_{i,j} S_i^{(j)}[h,l]$$

and

$$\beta_{hl} = (A[\sigma(h),\sigma(l)] - A[h,l]),$$

so that $Y_{hl} = \alpha_{hl}\beta_{hl}$. It is immediate that conditioning on $B^{(i)}, i = 1,2$, we have $\{\alpha_{hl}\}_{h,l}$ is independent of $\{\beta_{hl}\}_{h,l}$. Below, we will implicitly condition on $B^{(i)}, i = 1,2$ in all expectations.

We define

$$X_{hl} := Y_{hl} - \mathbb{E}(Y_{hl}) = \alpha_{hl}\beta_{hl} - \mathbb{E}(\alpha_{hl})\mathbb{E}(\beta_{hl})$$

We first note:

$$\begin{aligned}
\mathbb{E}[X_{hl}^4] &= \mathbb{E}\left([\alpha_{hl}\beta_{hl} - \mathbb{E}(\alpha_{hl})\mathbb{E}(\beta_{hl})]^4\right) \\
&= \mathbb{E}\left(([\alpha_{hl} - \mathbb{E}(\alpha_{hl})]\,[\beta_{hl} + \mathbb{E}(\beta_{hl})] + [\beta_{hl}\mathbb{E}(\alpha_{hl}) - \alpha_{hl}\mathbb{E}(\beta_{hl})])^4\right) \\
&\leq 2^3\left(\mathbb{E}([\alpha_{hl} - \mathbb{E}(\alpha_{hl})]^4)\mathbb{E}([\beta_{hl} + \mathbb{E}(\beta_{hl})]^4) + \mathbb{E}([\beta_{hl}\mathbb{E}(\alpha_{hl}) - \alpha_{hl}\mathbb{E}(\beta_{hl})]^4)\right) \\
&= 2^3\left(\mathbb{E}([\alpha_{hl} - \mathbb{E}(\alpha_{hl})]^4)\mathbb{E}([\beta_{hl} + \mathbb{E}(\beta_{hl})]^4) + \mathbb{E}([\beta_{hl}(\mathbb{E}(\alpha_{hl}) - \alpha_{hl}) + \alpha_{hl}(\beta_{hl} - \mathbb{E}(\beta_{hl}))]^4)\right) \\
&\leq 2^3\Big(\mathbb{E}([\alpha_{hl} - \mathbb{E}(\alpha_{hl})]^4)\mathbb{E}([\beta_{hl} + \mathbb{E}(\beta_{hl})]^4) \\
&\qquad + 2^3\left[\mathbb{E}(\beta_{hl}^4)\mathbb{E}([\alpha_{hl} - \mathbb{E}(\alpha_{hl})]^4) + \mathbb{E}(\alpha_{hl}^4)\mathbb{E}([\beta_{hl} - \mathbb{E}(\beta_{hl})]^4)\right]\Big) \\
&= A_1\mathbb{E}([\alpha_{hl} - \mathbb{E}(\alpha_{hl})]^4) + B_2\mathbb{E}(\alpha_{hl}^4)
\end{aligned}$$

where $A_1 = 8\mathbb{E}([\beta_{hl} + \mathbb{E}(\beta_{hl})]^4) + 64\mathbb{E}(\beta_{hl}^4)$ and $B_2 = 64\mathbb{E}([\beta_{hl} + E(\beta_{hl})]^4)$.

Note that α_{hl} follows the Poisson-binomial distribution with m independent summands. The 4-th central moment of the Poisson-binomial distribution can be calculated via its Excess Kurtosis which has magnitude $O(1/m)$ and its variance which has magnitude of $\sigma^2 = O(m)$. The 4th central moment therefore has magnitude of $O(1/m)O(m^2) = O(m)$. The 4th non-central moment of the Poisson binomial distribution is of order $O(m^4)$. Turning our attention to β, there are three cases to consider:

1. If $B^{(1)}[h,l] = B^{(1)}[\sigma(h),\sigma(l)]$, then we know

$$\mathbb{P}(\beta_{hl} = a) = \begin{cases} p^2 + (1-p)^2, & a = 0 \\ p(1-p), & a = 1 \\ p(1-p), & a = -1 \end{cases}$$

and $\mathbb{E}(\beta_{hl}) = 0; \mathbb{V}(\beta_{hl}) = 2p(1-p)$

2. If $B^{(1)}[h,l] = 1, B^{(1)}[\sigma(h), \sigma(l)] = 0$, then we know

$$P(\beta_{hl} = a) = \begin{cases} 2p(1-p), & a = 0 \\ p^2, & a = 1 \\ (1-p)^2, & a = -1 \end{cases}$$

$\mathbb{E}(\beta_{hl}) = 2p - 1; \mathbb{V}(\beta_{hl}) = 2p(1-p)$

3. If $B^{(1)}[h,l] = 0, B^{(1)}[\sigma(h), \sigma(l)] = 1$, then we know

$$P(\beta_{hl} = a) = \begin{cases} 2p(1-p), & a = 0 \\ p^2, & a = -1 \\ (1-p)^2, & a = 1 \end{cases}$$

$\mathbb{E}(\beta_{hl}) = 1 - 2p; \mathbb{V}(\beta_{hl}) = 2p(1-p)$

Now, it is clear that $A_1 = 8E[\beta_{hl} + E(\beta_{hl})]^4 + 64E(\beta_{hl}^4)$ and $B_2 = 64E[\beta_{hl} + E(\beta_{hl})]^4$ are two constants that do not grow with m. This yields that $E[X_{hl}^4] = O(m^4)$. Similarly, we can show $E[|X_{hl}|^3] = O(m^3)$.

Next, we have (where $\mathbb{V}$ is shorthand for variance, and we are implicitly conditioning

on the $B^{(i)}$'s below)

$$\begin{aligned}\mathbb{V}\left(\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} X_{hl}\right) &= \mathbb{V}\left(\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} Y_{hl}\right)\\ &= \underbrace{\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \mathbb{V}(Y_{hl})}_{:=V1}\\ &\quad + \underbrace{\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \sum_{\substack{h_2,l_2 \text{ s.t. } \{h_2,l_2\}\neq\{h,l\} \text{ and}\\ \{\sigma(h_2),\sigma(l_2)\}\neq\{h_2,l_2\}}} \mathrm{Cov}(Y_{hl},Y_{h_2l_2})}_{:=C2}\end{aligned}$$

Now, as PP^* shuffles exactly k labels, the size of the set $\{h,l,\text{ s.t. }\{\sigma(h),\sigma(l)\}\neq\{h,l\}\}$ is $\Theta(nk)$. We then have

$$\begin{aligned}V1 &= \sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \mathbb{V}(\alpha_{hl}\beta_{hl})\\ &= \sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \mathbb{V}(\alpha_{hl})\mathbb{V}(\beta_{hl}) + [\mathbb{E}(\beta_{hl})]^2\mathbb{V}(\alpha_{hl}) + [\mathbb{E}(\alpha_{hl})]^2\mathbb{V}(\beta_{hl})\\ &= \sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \mathbb{E}(\beta_{hl}^2)\underbrace{\mathbb{V}(\alpha_{hl})}_{=\Theta(m)} + \mathbb{V}(\beta_{hl})[\mathbb{E}(\alpha_{hl})]^2\\ &= \Theta(nkm) + 2p(1-p)\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} [\mathbb{E}(\alpha_{hl})]^2\end{aligned}$$

Next, we have

$$
\begin{aligned}
C2 &= \sum_{\substack{h,l,\ \text{s.t.} \\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \left\{ \mathrm{Cov}\left[\alpha_{hl}\beta_{hl}, \alpha_{\sigma^{-1}(h)\sigma^{-1}(l)}\beta_{\sigma^{-1}(h)\sigma^{-1}(l)}\right] + \mathrm{Cov}\left[\alpha_{hl}\beta_{hl}, \alpha_{\sigma(h)\sigma(l)}\beta_{\sigma(h)\sigma(l)}\right] \right\} \\
&= - \sum_{\substack{h,l,\ \text{s.t.} \\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \{\mathbb{E}(\alpha_{hl})\mathbb{E}(\alpha_{\sigma^{-1}(h)\sigma^{-1}(l)})\mathbb{V}(A[h,l]) + \mathbb{E}(\alpha_{hl})\mathbb{E}(\alpha_{\sigma(h)\sigma(l)}) \cdot \mathbb{V}(A[\sigma(h),\sigma(l)])\} \\
&= -2p(1-p) \sum_{\substack{h,l,\ \text{s.t.} \\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \mathbb{E}(\alpha_{hl})\mathbb{E}(\alpha_{\sigma(h)\sigma(l)})
\end{aligned}
$$

Combining, we then see

$$
\mathbb{V}\left(\sum_{\substack{h,l,\ \text{s.t.} \\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} X_{hl} \right) = \Theta(nkm) + p(1-p) \sum_{\substack{h,l,\ \text{s.t.} \\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \left(\mathbb{E}(\alpha_{hl}) - \mathbb{E}(\alpha_{\sigma(h)\sigma(l)})\right)^2
$$

Now, α_{hl} follows the Poisson-Binomial distribution, and

$$
\mathbb{E}(\alpha_{hl}) = m_1 \left((1-2p)B^{(1)}[h,l] + p \right) + m_2 \left((1-2p)B^{(2)}[h,l] + p \right),
$$

and so

$$
\begin{aligned}
&\mathbb{E}(\alpha_{\sigma(h)\sigma(l)}) - \mathbb{E}(\alpha_{hl}) \\
&\quad = (1-2p) \left(m_1 \left(B^{(1)}[\sigma(h),\sigma(l)] - B^{(1)}[h,l] \right) + m_2 \left(B^{(2)}[\sigma(h),\sigma(l)] - B^{(2)}[h,l] \right) \right).
\end{aligned}
$$

Without loss of generality, we will consider $P^* = I_n$ below (this is done to simply ease

notation). By assumption, we have that

$$\frac{h(B^{(2)},B^{(1)},P)}{-h(B^{(1)},B^{(1)},P)} > \frac{m_1(1-2p)}{m_2(1-2p)} = \frac{m_1}{m_2}, \tag{2.9}$$

where we recall

$$h(B^{(i)},B^{(j)},P) = \mathrm{tr}(B^{(i)}PB^{(j)}P^T) - \mathrm{tr}(B^{(i)}B^{(j)}).$$

Note that the possible values of $\left(B^{(1)}[\sigma(h),\sigma(l)] - B^{(1)}[h,l]\right)$ and $\left(B^{(2)}[\sigma(h),\sigma(l)] - B^{(2)}[h,l]\right)$ are $-1, 0$ or 1. A key term in the variance computation above is

$$\begin{aligned}
&p(1-p) \sum_{\substack{h,l, \text{ s.t.} \\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \left(\mathbb{E}(\alpha_{hl}) - \mathbb{E}(\alpha_{\sigma(h)\sigma(l)})\right)^2 \\
&= p(1-p)(1-2p)^2 \cdot \\
&\quad \sum_{\substack{h,l, \text{ s.t.} \\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \left(m_1\left(B^{(1)}[\sigma(h),\sigma(l)] - B^{(1)}[h,l]\right) + m_2\left(B^{(2)}[\sigma(h),\sigma(l)] - B^{(2)}[h,l]\right)\right)^2
\end{aligned} \tag{2.10}$$

We desire (for the application of Stein's method in Theorem 2.4) that this term is $\omega(m^2(nk)^{2/3})$. When will this be the case?

For each $x \in \{0,1\}^4$, let

$$N_x := \left|\left\{ \{h,\ell\} \in \binom{V}{2} \text{ s.t. } \left(B^{(1)}[\sigma(h),\sigma(l)], B^{(1)}[h,l], B^{(2)}[\sigma(h),\sigma(l)], B^{(2)}[h,l]\right) = x \right\}\right|$$

Note that, by parity, we have

$$N_{0110} + N_{0111} + N_{0100} + N_{0101} = N_{1010} + N_{1011} + N_{1000} + N_{1001}$$

$$N_{0001} + N_{1101} + N_{1001} + N_{0101} = N_{0010} + N_{1110} + N_{1010} + N_{0110}$$

Equation 2.9 is then equivalent to

$$m_2(N_{0110}+N_{1110}-N_{0101}-N_{1101}) > m_1(N_{0110}+N_{0111}+N_{0100}+N_{0101}).$$

This then implies

$$\begin{aligned}
&\frac{m_2}{2}(N_{0110}+N_{1110}-N_{0101}-N_{1101})+\frac{m_2}{2}(N_{1001}+N_{0001}-N_{1010}-N_{0010}) \\
&\qquad > \frac{m_1}{2}(N_{0110}+N_{0111}+N_{0100}+N_{0101})+\frac{m_1}{2}(N_{1010}+N_{1011}+N_{1000}+N_{1001}) \\
&\Leftrightarrow \frac{m_2}{2}(N_{0110}+N_{1110}+N_{1001}+N_{0001}) > \frac{m_1}{2}(N_{0110}+N_{0111}+N_{0100}+N_{0101}) \\
&\qquad\qquad + \frac{m_1}{2}(N_{1010}+N_{1011}+N_{1000}+N_{1001}) \\
&\qquad\qquad + \frac{m_2}{2}(N_{0101}+N_{1101}+N_{1010}+N_{0010}) \qquad (2.11)
\end{aligned}$$

Note that in Eq. 2.10, each

$$\begin{aligned}
&N_{1010} \text{ term contributes } (m_1+m_2)^2; \quad N_{0101} \text{ term contributes } (m_1+m_2)^2; \\
&N_{1001} \text{ term contributes } (m_1-m_2)^2; \quad N_{0110} \text{ term contributes } (m_1-m_2)^2; \\
&N_{0010} \text{ term contributes } m_2^2; \quad N_{1110} \text{ term contributes } m_2^2; \\
&N_{0001} \text{ term contributes } m_2^2; \quad N_{1101} \text{ term contributes } m_2^2; \\
&N_{1011} \text{ term contributes } m_1^2; \quad N_{1000} \text{ term contributes } m_1^2; \\
&N_{0111} \text{ term contributes } m_1^2; \quad N_{0100} \text{ term contributes } m_1^2.
\end{aligned}$$

We consider the following cases:

i. $|\mathbf{m_1}-\mathbf{m_2}|=\mathbf{o}(\mathbf{m})$: In this case the N_{1001} and N_{0110} terms contribute minimally (i.e., of order $o(m^2)$ and not of order m^2) to Eq. 2.10. In order for Eq. 2.10 to be of order

$\omega(m^2(nk)^{2/3})$ it is necessary and sufficient for at least one of

$$N_{1010}, N_{0101}, N_{0010}, N_{1110}, N_{0001}, N_{1101}, N_{1011}, N_{1000}, N_{0111}, N_{0100}$$

to be $\omega((nk)^{2/3})$, which, by Eq. 2.11, is equivalent to

$$N_{1110} + N_{0001} = \omega((nk)^{2/3}).$$

ii. $\mathbf{m_1}, \mathbf{m_2} = \blacksquare(\mathbf{m})$, $|\mathbf{m_1} - \mathbf{m_2}| = \blacksquare(\mathbf{m})$: In this case, all terms contribute meaningfully (i.e., order m^2) to Eq. 2.10. If $m = \omega(1)$, then in order for Eq. 2.10 to be of order $\omega(m^2(nk)^{2/3})$ it is necessary and sufficient for at least one of

$$N_{1010}, N_{0101}, N_{1001}, N_{0110}, N_{0010}, N_{1110}, N_{0001}, N_{1101}, N_{1011}, N_{1000}, N_{0111}, N_{0100},$$

to be $\omega((nk)^{2/3})$, which, by Eq. 2.11, is equivalent to

$$N_{1110} + N_{0001} + N_{1001} + N_{0110} = \omega((nk)^{2/3}).$$

iii. $\mathbf{m_2}/\mathbf{m_1} = \mathbf{!}(\mathbf{1})$: In this case the N_{1011}, N_{1000}, N_{0111}, and N_{0100} terms contribute minimally (i.e., of order $m_1^2 \ll m^2$) to Eq. 2.10. If $m = \omega(1)$, then in order for Eq. 2.10 to be of order $\omega(m^2(nk)^{2/3})$ it is necessary and sufficient for at least one of

$$N_{1010}, N_{0101}, N_{1001}, N_{0110}, N_{0010}, N_{1110}, N_{0001}, N_{1101},$$

to be $\omega((nk)^{2/3})$, which, by Eq. 2.11, is equivalent to

$$N_{1110} + N_{0001} + N_{1001} + N_{0110} = \omega((nk)^{2/3}).$$

If the conditions above hold, we have

$$\mathbb{V}\left(\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} X_{hl}\right) = \Theta(nkm) + p(1-p)\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}} \left(\mathbb{E}(\alpha_{hl}) - \mathbb{E}(\alpha_{\sigma(h)\sigma(l)})\right)^2$$
$$= \Theta(nkm) + \omega(m^2(nk)^{2/3})$$

In this case, the bound in Stein's method becomes(where $\sum_*$ is shorthand for $\sum_{\substack{h,l,\text{ s.t.}\\ \{\sigma(h),\sigma(l)\}\neq\{h,l\}}}$) and

$$W = \frac{\sum_* X_{hl}}{\sqrt{\mathbb{V}_B(\sum_* X_{hl})}},$$

$$d_K(W,Z) \leq \sqrt{\frac{O(nkm^3)}{\Theta((nkm)^{3/2}) + \omega(m^3nk)} + \frac{O((nk)^{1/2}m^2)}{\Theta(nkm) + \omega(m^2(nk)^{2/3})}} = o(1)$$

as desired. In the event that none of the growth conditions outlined above for the N_{ijkl}'s hold, then

$$\mathbb{V}_B(\sum_* X_{hl}) = \Omega(nkm)$$

and we can bound $d_K(W,Z)$ via

$$d_K(W,Z) \leq \sqrt{\frac{O(nkm^3)}{\Omega((nkm)^{3/2})} + \frac{O((nk)^{1/2}m^2)}{\Omega(nkm)}} = \sqrt{O\left(\frac{m^{3/2}}{(nk)^{1/2}}\right) + O\left(\frac{m}{(nk)^{1/2}}\right)}$$

and this bound is $o(1)$ when $nk \gg m^3$ as desired. □

2.7.4 Proof of Corollary 2.1

By the normal convergence in Theorem 2.1, we have that (where $Z \sim$N(0,1))

$$\begin{aligned}
\mathbb{P}(f(P) - f(P^*) > 0) &= \mathbb{P}\left(\frac{f(P) - f(P^*) - \mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} > \frac{-\mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} \right) \\
&\geq \mathbb{P}\left(\frac{f(P) - f(P^*) - \mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} > 0 \right) \\
&\to \mathbb{P}(Z > 0) = 1/2.
\end{aligned}$$

For part ii., let $\varepsilon > 0$ fixed. We have that for any constant $C > 0$,

$$\begin{aligned}
\mathbb{P}(f(P) - f(P^*) > 0) &= \mathbb{P}\left(\frac{f(P) - f(P^*) - \mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} > \frac{-\mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} \right) \\
&\geq \mathbb{P}\left(\frac{f(P) - f(P^*) - \mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} > \frac{-Cm\sqrt{n\xi \log n}}{\sqrt{n\xi m^2}} \right) \\
&\geq \mathbb{P}\left(\frac{f(P) - f(P^*) - \mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} > -C\sqrt{\log n} \right)
\end{aligned}$$

For n sufficiently large, this last term is bounded below by (where $\Phi(x) = \mathbb{P}(Z \leq x)$)

$$\begin{aligned}
&\mathbb{P}\left(\frac{f(P) - f(P^*) - \mathbb{E}_B(f(P) - f(P^*))}{\sqrt{\mathrm{Var}_B(f(P) - f(P^*))}} > -\Phi^{-1}(1-\varepsilon) \right) \\
&\to \Phi(\Phi^{-1}(1-\varepsilon)) = 1 - \varepsilon.
\end{aligned}$$

As ε was arbitrary, letting it go to 0 finishes the proof.

2.7.5 Proof of Lemma 2.1

Suppose such P matrix exists, for any graph $B^{(j)}$, where $j = 2,3,\ldots,m$, we consider the matching objective function

$$\|P^T\mathbb{E}(B^{(1)})P - \mathbb{E}(B^{(j)})\|_F^2 = \|P^\top UR^{(1)}U^TP - UR^{(j)}U^T\|_F^2.$$

We can lift U and R's to $\tilde{U}$ and $\tilde{R}^{(j)}$ such that $\tilde{U}$ is an orthogonal matrix, $\tilde{R}^{(j)}$'s are still diagonal matrices, and $\tilde{U}\tilde{R}^{(j)}\tilde{U}^T = \mathbb{E}(B^{(j)})$ for all j. Therefore we know

$$\begin{aligned}
-\|P^TUR^{(1)}U^TP - UR^{(j)}U^T\|_F^2 &= -\|P^T\tilde{U}\tilde{R}^{(1)}\tilde{U}^TP - \tilde{U}\tilde{R}^{(j)}\tilde{U}^T\|_F^2 \\
&= -\|P^T\tilde{U}\tilde{R}^{(1)}\tilde{U}P^T\|_F^2 - \|\tilde{U}\tilde{R}^{(j)}\tilde{U}^T\|_F^2 + 2\operatorname{tr}(P^T\tilde{U}\tilde{R}^{(1)}\tilde{U}^TP\tilde{U}\tilde{R}^{(j)}\tilde{U}^T) \\
&= 2\operatorname{tr}(\tilde{R}^{(1)}X\tilde{R}^{(j)}X^T) - K
\end{aligned}$$

where $X = \tilde{U}^TP\tilde{U}$, and $K = -\|\tilde{R}^{(1)}\|_F^2 - \|\tilde{R}^{(j)}\|_F^2 \in \mathbb{R}$ is independent of P.

For general $X \in \mathbb{R}^{d\times d}$, define the matrix functional $f_2(X) = \operatorname{tr}(\tilde{R}^{(1)}X\tilde{R}^{(j)}X^T)$. Letting $\tilde{Q} = Q \oplus \mathbf{0}_{n-d}$ (where $\mathbf{0}_{n-d}$ is the $(n-d)\times(n-d)$ matrix of all 0's), we we have that $f_2(\tilde{Q}) > f_2(I) = f_2(\tilde{U}I\tilde{U}^T)$ by assumption. Further define the functional

$$g_2(X) = \sqrt{\operatorname{tr}\left(\left(\tilde{R}^{(1)}\right)^2X\left(\tilde{R}^{(j)}\right)^2X^T\right)}.$$

The diagonal elements of each $R^{(j)}$ are nonnegative, and by the $\ell 1 - \ell 2$ norm inequality, we have that $f_2(\tilde{Q}) \geq g_2(\tilde{Q})$. Let $W = U^TPU \oplus \mathbf{0}_{n-d}$, and define

$$\varepsilon = \|U^TPU - Q\|_F = \|W - \tilde{Q}\|_F.$$

Recall our assumption that

$$Q \in \operatorname{argmin}_{V \in \Pi_d} \|R^{(1)} - VR^{(j)}V^T\|_F$$
$$I_d \notin \operatorname{argmin}_{V \in \Pi_d} \|R^{(1)} - VR^{(j)}V^T\|_F,$$

Now, we know that

$$\begin{aligned} f_2(W) &= \operatorname{tr}(\tilde{R}^{(1)} W \tilde{R}^{(j)} W^T) \\ &= \operatorname{tr}(\underbrace{\tilde{U}\tilde{R}^{(1)}\tilde{U}^T}_{\mathbb{E}(B^{(1)})} \underbrace{P\tilde{U}\tilde{R}^{(j)}\tilde{U}^T P^T}_{P\mathbb{E}(B^{(j)})P^T}) \end{aligned}$$

As both $\mathbb{E}(B^{(1)})$ and $P\mathbb{E}(B^{(j)})P^T$ are Hermition with respective eigenvalues the diagonal entries of $R^{(1)}$ and $R^{(j)}$, we have that

$$\operatorname{tr}(\mathbb{E}(B^{(1)})P\mathbb{E}(B^{(j)})P^T) \leq f_2(Q)$$

as Q sorts the eigenvalues of $\mathbb{E}(B^{(1)})$ and $P\mathbb{E}(B^{(j)})P^T$ to both be in non-decreasing order; see Theorem 1 in [52]. Similarly $g_2(W) \leq g_2(\tilde{Q})$. Now we consider the mean value theorem (MVT) applied to the function f_2: By the multivariate MVT, we know there is a point $c\tilde{Q} + (1-c)W$ where $c \in (0,1)$ such that

$$f_2(\tilde{Q}) - f_2(W) = \left(\operatorname{vec}(\nabla f_2(c\tilde{Q} + (1-c)W))\right)^T \operatorname{vec}(\tilde{Q} - W)$$

Plugging in $\nabla f_2(X) = 2\tilde{R}^{(1)}X\tilde{R}^{(j)}$, we get

$$\begin{aligned}
f_2(\tilde{Q}) - f_2(W) &= 2\left(\mathrm{vec}(\tilde{R}^{(1)}(c\tilde{Q}+(1-c)W)\tilde{R}^{(j)})\right)^T \mathrm{vec}(\tilde{Q}-W) \\
&= 2\left[(\tilde{R}^{(1)}\otimes\tilde{R}^{(j)})\,\mathrm{vec}(c\tilde{Q}+(1-c)W)\right]^T \mathrm{vec}(\tilde{Q}-W) \\
&\leq 2\left\|\mathrm{vec}(c\tilde{Q}+(1-c)W)^T(\tilde{R}^{(1)}\otimes\tilde{R}^{(j)})\right\|_2 \|\mathrm{vec}(\tilde{Q}-W)\|_2 \\
&= 2\|\tilde{R}^{(1)}(c\tilde{Q}+(1-c)W)\tilde{R}^{(j)}\|_F \|\tilde{Q}-W\|_F \\
&\leq 2\varepsilon\left(c\|\tilde{R}^{(1)}\tilde{Q}\tilde{R}^{(j)}\|_F + (1-c)\|\tilde{R}^{(1)}W\tilde{R}^{(j)}\|_F\right) \\
&\leq 2\varepsilon\|\tilde{R}^{(1)}\tilde{Q}\tilde{R}^{(j)}\|_F \\
&= 2\varepsilon\sqrt{\mathrm{tr}\left(\left(\tilde{R}^{(1)}\right)^2\tilde{Q}\left(\tilde{R}^{(j)}\right)^2\tilde{Q}^T\right)} \\
&= 2\varepsilon g_2(\tilde{Q}) \\
&\leq 2\varepsilon f_2(\tilde{Q})
\end{aligned}$$

Thus we conclude $f_2(W) \geq (1-2\varepsilon)f_2(\tilde{Q})$, which implies (by the assumption on P)

$$\begin{aligned}
\mathrm{tr}(P^T\mathbb{E}(B^{(1)})P\mathbb{E}(B^{(j)})) &= \mathrm{tr}(P^T\tilde{U}\tilde{R}^{(1)}\tilde{U}^TP\tilde{U}\tilde{R}^{(j)}\tilde{U}^T) \\
&= \mathrm{tr}(W^T\tilde{R}^{(1)}W\tilde{R}^{(j)}) \\
&\geq (1-2\varepsilon)\,\mathrm{tr}(\tilde{Q}^T\tilde{R}^{(1)}\tilde{Q}\tilde{R}^{(j)}) \\
&> \mathrm{tr}(\tilde{R}^{(1)}\tilde{R}^{(j)}) \\
&= \mathrm{tr}(\tilde{U}\tilde{R}^{(1)}\tilde{U}^T\tilde{U}\tilde{R}^{(j)}\tilde{U}^T) \\
&= \mathrm{tr}(\mathbb{E}(B^{(1)})\mathbb{E}(B^{(j)}))
\end{aligned}$$

as desired.

2.7.6 Additional computational details

Proof of Proposition 2.1 McDiarmid Concentration

Fix ξ. From Eq. 2.8 we can see that $f(P) - f(P^*)$ is a function of $\Theta(n\xi)$ independent random variables

$$\left\{ A[h,\ell], X[h,\ell] := \left(\sum_{i,j} S_i^{(j)}[h,\ell] \right) \right\}_{\{\sigma(h),\sigma(\ell)\} \neq \{h,\ell\}}.$$

A single change in one of these variables can change the value of $f(P) - f(P^*)$ by at most $O(m)$. Suppose that $\mathbb{E}_B(f(P) - f(P^*)) < 0$ holds, then by McDiarmid Inequality, we know for any fixed $\xi < n$ and P such that $P(P^*)^T \in \Pi_{n,\xi}$, we have

$$\begin{aligned}
\mathbb{P}([f(P) - f(P^*)] \geq 0) \leq & \mathbb{P}(|\,[f(P) - f(P^*)] - \mathbb{E}_B(f(P) - f(P^*))| \geq \mathbb{E}_B(f(P) - f(P^*))) \\
\leq & 2\exp\left(- \frac{[\mathbb{E}_B(f(P) - f(P^*))]^2}{O(n\xi m^2)} \right) \\
= & 2\exp\left(-\omega(\xi \log n)\right)
\end{aligned}$$

A union over such P and ξ, we get:

$$\begin{aligned}
\mathbb{P}(\{P^*\} \notin \operatorname{argmin}_P \|C - PRP^T\|_F) &= \mathbb{P}\left\{ \exists \xi < n, P \in \Pi_n \text{ s.t. } P(P^*)^T \in \Pi_{n,\xi} : [f(P) - f(P^*)] \geq 0 \right\} \\
&\leq \sum_{\xi} e^{O(\xi \log(n))} \cdot 2e^{-\omega(\xi \log n)} \\
&= 2e^{-\omega(\log n)} = o(1)
\end{aligned}$$

as desired.

Derivation of Eq. 2.6

For any graph $B \in \mathscr{G}_n$, let $\bar{B}$ denote the complement network in $\mathscr{G}_n$. By linearity of the expectation and the trace, combined with Eq. 2.2 and the assumptions that all p_j identically equal p, we have (where $P^{\bigstar} := P(P^*)^T$, J is the hollow $n \times n$ matrix with all off-diagonal entries identically equal to 1 so that $\bar{B} = J - B$)

$$
\begin{aligned}
\mathbb{E}_B(\mathrm{tr}(C^{(j)}PRP^T)) =& \mathbb{E}_B(\mathrm{tr}(S_i^{(j)}PRP^T)) \\
=& (1-p)^2\mathrm{tr}(B^{(j)}P(P^*)^TB^{(1)}P^*P^T) + p(1-p)\,\mathrm{tr}(\bar{B}^{(j)}P(P^*)^TB^{(1)}P^*P^T) \\
&+ (1-p)p\,\mathrm{tr}(B^{(j)}P(P^*)^T\bar{B}^{(1)}P^*P^T) + p^2\mathrm{tr}(\bar{B}^{(j)}P(P^*)^T\bar{B}^{(1)}P^*P^T) \\
=& \mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) - 2p\,\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) \\
&+ p^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) + p\,\mathrm{Tr}\left(JP^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) \\
&- p\,\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) - p^2\mathrm{Tr}\left(JP^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) \\
&+ p^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) + p\,\mathrm{Tr}\left(B^{(j)}P^{\bigstar}J(P^{\bigstar})^T\right) \\
&- p\,\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) - p^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}J(P^{\bigstar})^T\right) \\
&+ p^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) + p^2\mathrm{Tr}(JP^{\bigstar}J(P^{\bigstar})^T) \\
&- p^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}J(P^{\bigstar})^T\right) - p^2\mathrm{Tr}\left(JP^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) \\
&+ p^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) \\
=& (1-2p)^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) + (p-2p^2)\left(\|B^{(1)}\|_F^2 + \|B^{(j)}\|_F^2\right) + p^2 2\binom{n}{2}
\end{aligned}
$$

We note that the identity $\mathrm{Tr}\left(JP^{\bigstar}B^{(j)}(P^{\bigstar})^T\right) = \mathrm{Tr}(JB^{(j)}) = \mathrm{Tr}\left((B^{(j)})^2\right) = \|B^{(j)}\|_F^2$ was used above. We then get

$$\begin{aligned}
&\mathbb{E}_B(\mathrm{tr}(C^{(1)}P^*R(P^*)^T)) - \mathbb{E}_B(\mathrm{tr}(C^{(i)}PRP^T)) \\
&\quad = (1-2p)^2\mathrm{Tr}\left(B^{(1)}B^{(1)}\right) + (p-2p^2)\|B^{(1)}\|_F^2 + (p-2p^2)\|B^{(1)}\|_F^2 \\
&\qquad - (1-2p)^2\mathrm{Tr}\left(B^{(j)}P^{\bigstar}B^{(1)}(P^{\bigstar})^T\right) - (p-2p^2)\|B^{(1)}\|_F^2 - (p-2p^2)\|B^{(j)}\|_F^2 \\
&\quad = [(1-2p)^2 + p - 2p^2]\|B^{(1)}\|_F^2 - (p-2p^2)\|B^{(j)}\|_F^2 - (1-2p)^2\mathrm{tr}(B^{(i)}P(P^*)^TB^{(1)}P^*P^T) \\
&\quad = (1-p)(1-2p)\|B^{(1)}\|_F^2 - p(1-2p)\|B^{(i)}\|_F^2 - (1-2p)^2\mathrm{tr}(B^{(i)}P(P^*)^TB^{(1)}P^*P^T)
\end{aligned}$$

as desired.

Proof of Theorem 2.3

Write

$$X_{i,P} = \sum_{h\ell}\left(C^{(1)}[h,\ell] - ((P^{\bigstar})^TC^{(i)}P^{\bigstar})[h,\ell]\right)A[\ell,h]$$

is a sum of $O(n^2)$ independent random variables—the $(C^{(1)}[h,\ell] - ((P^{\bigstar})^TC^{(i)}P^{\bigstar})[h,\ell])A[\ell,h]$'s—which are all bounded in $[-2,2]$. Hoeffding's inequality then yields

$$\begin{aligned}
\mathbb{P}(X_{i,P} \le 0) &\le \mathbb{P}(|X_{i,P} - \mathbb{E}X_{i,P}| \ge \mathbb{E}X_{i,P}) \\
&\le 2\exp\left\{-\frac{2(\mathbb{E}X_{i,P})^2}{16n^2}\right\} \\
&\le 2\exp\{-\omega(\xi\log(n))\}
\end{aligned}$$

Then

$$\begin{aligned}\mathbb{P}(\exists i \in [k] \setminus \{1\}, P \in \Pi_n \text{ s.t.} X_{i,P} \leq 0) &\leq \sum_{i=2}^{k} \sum_{P \text{ s.t. } P^{\bigstar} \in \Pi_{n,\xi}} 2\exp\{-\omega(\xi \log(n))\} \\ &\leq 2k \sum_{\xi} \exp\{O(\xi \log(n))\} \exp\{-\omega(\xi \log(n))\} \\ &= \exp\{-\omega(\log(n))\}\end{aligned}$$

as desired.

2.8 Additional experiments and figures

2.8.1 ER p=0.5

In this section, we include the results and output of additional experiments. We first display Table 2.4 and Figure 2.6 displaying matching accuracy and matching objective function for the ER($n, p = 0.5$) single background setting.

2.8.2 ER p=0.3

We display Table 2.5 and Figure 2.7 displaying matching accuracy for the ER($n, p = 0.3$) single background setting.

2.8.3 Clustering the brain graphs

To demonstrate how we can obtain the brain graph clusters, we consider the following simple example. Using 135 in-sample brain graphs considered from the HNU1, we compute the matrix of inter-graph distances $D_{ij} = \|A_i - A_j\|_F$ (displayed in Figure 2.8). Embedding this distance matrix into $\mathbb{R}^{14}$ using canonical multidimensional scaling (14 chosen

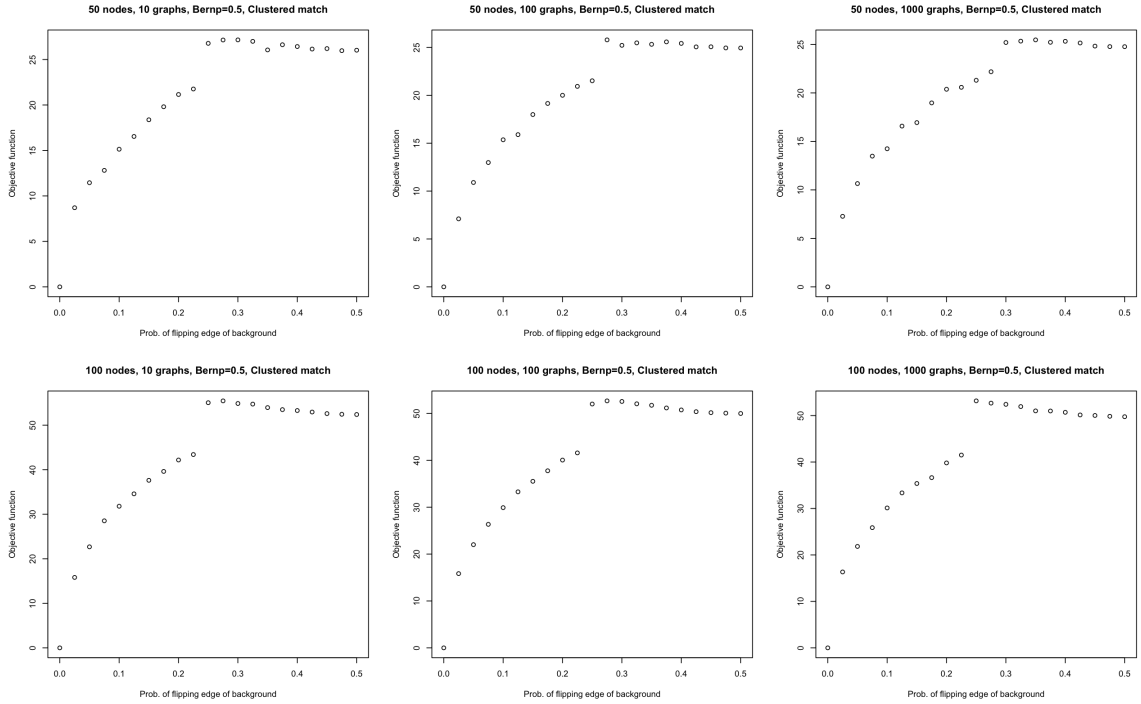

Figure 2.6: With a single background $B \sim \mathrm{ER}(n, 0.5)$, we consider $A, S_i^{(1)} \overset{i.i.d.}{\sim} \mathrm{BF}(B, q)$, and we match A (i.e., $P^* = I_n$) to C using `SGM` with 5 seeds. Varying the number of nodes n ($n = 50$ in the top panels, and $n = 100$ in the bottom panels), and the number of in-sample graphs ($m = 10$ in the left panels, $m = 100$ in the middle panels, and $m = 1000$ in the right panels), we plot the `SGM` objective function value $f = \|A - PCP^T\|_F$ versus the value of the edge perturbation parameter q, averaged over 10 Monte Carlo iterates.

by an elbow analysis of the scree plot of singular values of D) and clustering the embedded graphs via K-means clustering (with $K = 15$, with 25 random restarts) yields an Adjusted Rand Index [99] of 1 (i.e., perfect clustering) between the obtained clusters and the true labels.

n	50	50	50	100	100	100
m	10	100	1000	10	100	1000
$q = 0$	1	1	1	1	1	1
$q = 0.025$	1	1	1	1	1	1
$q = 0.050$	1	1	1	1	1	1
$q = 0.075$	1	1	1	1	1	1
$q = 0.100$	1	1	1	1	1	1
$q = 0.125$	1	1	1	1	1	1
$q = 0.150$	1	1	1	1	1	1
$q = 0.175$	1	1	1	1	1	1
$q = 0.200$	1	1	1	1	1	1
$q = 0.225$	1	1	1	1	1	1
$q = 0.250$	0.36	1	1	0.18	0.36	0.16
$q = 0.275$	0.22	0.24	1	0.09	0.16	0.14
$q = 0.300$	0.14	0.44	0.38	0.06	0.11	0.10
$q = 0.325$	0.16	0.22	0.34	0.13	0.06	0.10
$q = 0.350$	0.20	0.18	0.22	0.08	0.06	0.08
$q = 0.375$	0.14	0.20	0.16	0.08	0.06	0.08
$q = 0.400$	0.12	0.14	0.16	0.07	0.06	0.11
$q = 0.425$	0.18	0.10	0.10	0.05	0.08	0.07
$q = 0.450$	0.16	0.10	0.14	0.05	0.06	0.06
$q = 0.475$	0.10	0.18	0.14	0.07	0.06	0.08
$q = 0.500$	0.12	0.14	0.12	0.05	0.05	0.05

Table 2.4: Table of matching accuracy in the single Erdős-Rényi background setting with $p = 0.5$, averaged over 10 Monte Carlo iterates

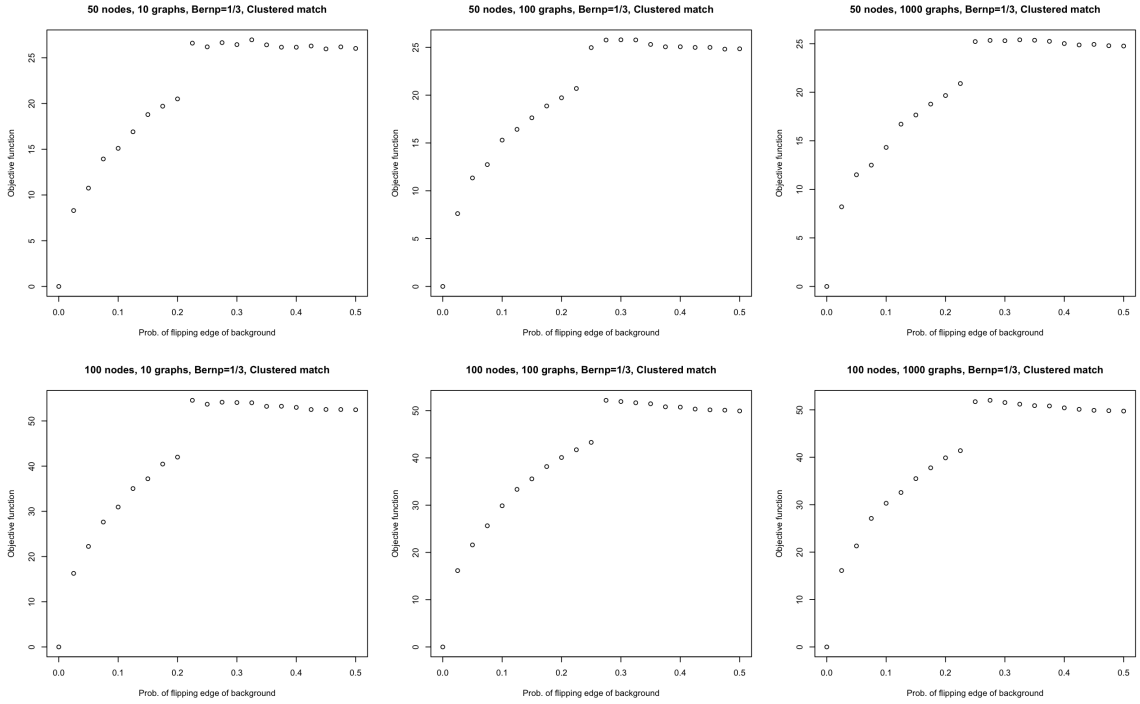

Figure 2.7: With a single background $B \sim \mathrm{ER}(n, 0.3)$, we consider $A, S_i^{(1)} \overset{i.i.d.}{\sim} \mathrm{BF}(B, q)$, and we match A (i.e., $P^* = I_n$) to C using `SGM` with 5 seeds. Varying the number of nodes n ($n = 50$ in the top panels, and $n = 100$ in the bottom panels), and the number of in-sample graphs ($m = 10$ in the left panels, $m = 100$ in the middle panels, and $m = 1000$ in the right panels), we plot the `SGM` objective function value $f = \|A - PCP^T\|_F$ versus the value of the edge perturbation parameter q, averaged over 10 Monte Carlo iterates.

n	50	50	50	100	100	100
m	10	100	1000	10	100	1000
$q = 0$	1	1	1	1	1	1
$q = 0.025$	1	1	1	1	1	1
$q = 0.050$	1	1	1	1	1	1
$q = 0.075$	1	1	1	1	1	1
$q = 0.100$	1	1	1	1	1	1
$q = 0.125$	1	1	1	1	1	1
$q = 0.150$	1	1	1	1	1	1
$q = 0.175$	1	1	1	1	1	1
$q = 0.200$	1	1	1	1	1	1
$q = 0.225$	0.14	1	1	0.10	1	1
$q = 0.250$	0.40	0.50	0.42	0.12	1	0.19
$q = 0.275$	0.20	0.12	0.36	0.12	0.12	0.12
$q = 0.300$	0.20	0.30	0.12	0.07	0.09	0.07
$q = 0.325$	0.22	0.14	0.20	0.06	0.07	0.11
$q = 0.350$	0.14	0.24	0.14	0.08	0.08	0.08
$q = 0.375$	0.20	0.18	0.10	0.05	0.06	0.06
$q = 0.400$	0.10	0.14	0.10	0.05	0.07	0.10
$q = 0.425$	0.16	0.10	0.12	0.06	0.05	0.09
$q = 0.450$	0.12	0.12	0.12	0.06	0.07	0.07
$q = 0.475$	0.10	0.10	0.14	0.05	0.06	0.06
$q = 0.500$	0.14	0.12	0.12	0.06	0.07	0.08

Table 2.5: Table of matching accuracy in the single Erdős-Rényi background setting with $p = 1/3$, averaged over 10 Monte Carlo iterates; similar results are obtained in the $p = 0.5$ setting; see Section 2.8 for detail.

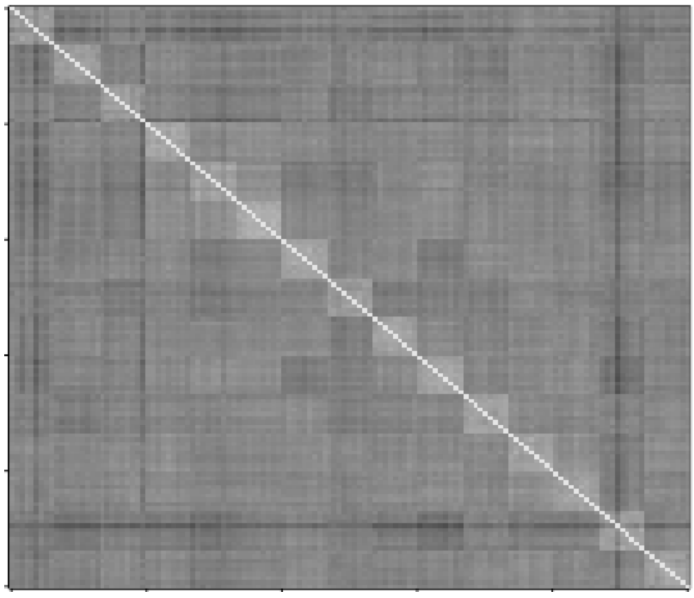

Figure 2.8: Inter-graph distance matrix heatmap for the 135 in-sample brain graphs considered from the HNU1 dataset, where each subject's scans are plotted contiguously (on the 9×9 diagonal block). Larger values in the heatmap are denoted by darker colors.

Chapter 3: Solution diversification in graph matching matched filters

3.1 Introduction

An extension to the GMP is the Subgraph Detection Problem, also known as the Subgraph Matching Problem (SMP) or Subgraph Isomorphism Problem. This extension relaxes the assumption that both graphs have the same number of nodes. In essence, given $G_1 \in \mathscr{G}_m$ and $G_2 \in \mathscr{G}_n$ with $m < n$, the SMP aims to find a subgraph of m nodes in G_2 that is structurally most similar to the template G_1. Note that a general subgraph $G_s = (V_s, E_s)$ of a graph $G = (V, E)$ is defined such that $E_s \subseteq E$ and $V_s \subseteq V$. In contrast, the term induced subgraph imposes the additional condition that for any $i, j \in V_s$, $\{i, j\} \in E_s$ iff $\{i, j\} \in E$. This extension is significant in various applications. For instance, [136] demonstrates the use of SMP on co-authorship networks to extract potential fake reviewers, while [131] discusses SMP with known protein complexes and protein-protein interaction networks to identify new protein complexes. SMP is also valuable in analyzing brain neural networks, where it helps identify specific regions of interest across multiple networks for focused analysis, as shown in [120] and [94]. Additionally, the SMP has been used for activity template detection in large knowledge graphs [81, 82] among myriad other applications in machine learning, social network analysis, computer vision, and pattern recognition [2, 107].

Numerous algorithms have been proposed to detect subgraphs from larger graphs that are isomorphic to the template (i.e., there exists an induced subgraph G_2' and a permutation matrix $P \in \Pi_m$ such that $\|A - PB'P^T\|_F = 0$, where B' is the adjacency matrix of G_2', and A

the adjacency matrix of G_1), with the first notable algorithm presented by [122]. Note that the (perhaps simpler) graph isomorphism problem also has a rich history in the literature, with recent results establishing at worst quasipolynomial complexity for the problem [10]. Detailed explanations and comparisons between state-of-the-art algorithms can be found in survey papers such as [107, 110]. Recently, a series of papers [81, 82, 133] introduced an exhaustive (designed to find all subgraphs of G_2 isomorphic to G_1) tree-based/filtering method that reduces the time required for SMP by eliminating symmetries (referred to as "structural equivalence" and "candidate equivalence") within the graph. The exhaustive nature of the tree-search/filtering based approaches is a key feature that will motivate our modification of the non-exhaustive algorithm of [111] in the following section. It should be noted that the aforementioned methods work well when an isomorphic copy of the template exists in the larger graph, but they often fail when such a promise is absent. In [111, 121], the authors relax the isomorphism requirement and instead aim to find a subgraph that shares the highest amount of structural (and feature-based in the case of [121]) similarity with the template.

Our focus in this chapter will be the matched-filters-based approach (abbreviated `GMMF` for graph-matching matched filters) of [111], in which the authors adapt the Frank-Wolfe-based [46] `SGM` algorithm of [44] by proposing different padding techniques to ensure that the template has the same number of nodes as the larger graph. The validity of their proposed padding methods is supported by both extensive simulations and theoretical justification. However, the `GMMF` algorithm and the adaptation in [92] (lifting the matched-filters approach to richly featured, multiplex networks) rely on efficiently solving iterative linear assignment problem (LAP) subroutines—via the Frank-Wolfe approach—which can be cumbersome in cases where the graphs are very large. Moreover, these algorithms are not designed to exhaustively search the background graph for all close (but perhaps suboptimal) matches, aiming instead to find only the best fitting subgraph(s). In cases where

more solution diversity is desired, this can limit the algorithm's applicability. These two concerns motivate our extensions of the GMMF routine to allow for both more solution diversification and greatly enhanced scalability. Note that code to implement this modified GMMF approach can be found at github.com/jataware/mgmmf.

We begin by introducing a random graph model in which we anchor our study, and provide an overview of the algorithm and modifications we will employ.

3.1.1 Multiple Correlated Erdős-Rényi

Recall the Erdős-Rényi model [38] is one of the most popular network models studied. While assuming all possible edges in the graph exist equally likely and independently, such a model still exhibits rich properties and provides fertile ground for studying graph matching problems. Discussions regarding thresholds of the graph properties for this model can be found in [21, 47] for the homogeneous Erdős-Rényi case and for the inhomogeneous case in [22]. Percolation theories on the Erdős-Rényi model have been proven in [16]. Within the related correlated Erdős-Rényi model, sharp thresholds for graph deanonymization are established in [30, 31, 73, 132], and recent polynomial time algorithms for almost sure exact graph matching (i.e., recovering the optimal solution asymptotically almost surely) have been established in [78] (with almost surely efficient seeded approaches proposed in [84]).

Our present focus on recovering both optimal and near-optimal solutions in the GMMF framework leads us to the following extension of the Erdős-Rényi model, dubbed the Multiple Correlated Erdős-Rényi model. This model is a natural extension of the classical correlated Erdős-Rényi model of [30, 31, 76, 132, 134] and the embedded template model of [111], where here we allow for multiple templates to be embedded in the background. In practice (see Section 3.3.4, it is often the case that there are multiple errorful copies of the template/motif in the background, and this model allows for such structure where the

template matches have varying levels of noise.

Definition 3.1. *(Multiple Correlated Erdős-Rényi) Let $m < n$ be nonnegative integers, with $\Lambda^{(1)} \in [0,1]^{m\times m}$, $\Lambda^{(2)} \in [0,1]^{n\times n}$ probability matrices. Let N be a nonnegative integer, and let $\mathscr{R} = (R_1, R_2, \cdots, R_N)$ be a sequence of symmetric matrices in $[0,1]^{m\times m}$. Two adjacency matrices, A and B, follow the Multiple Correlated Erdős-Rényi Model with parameters Λ_1, Λ_2 and $\mathscr{R}$ if*

i. For all $u, v \in \{1, \cdots, m\}$, $A_{uv} \overset{ind.}{\sim}$ Bernoulli$(\Lambda^{(1)}_{uv})$, and for all $u, v \in \{1, \cdots, n\}$, $B_{uv} \sim$ Bernoulli$(\Lambda^{(2)}_{uv})$;

ii. There exist induced subgraphs $(B^{(1)}, \cdots, B^{(N)})$ of B, each with m vertices that are not necessarily disjoint among these subgraphs, such that for $i = 1, \cdots, N$, and $u, v \in \{1, \cdots, m\}$, $A_{uv}, B^{(i)}_{uv} \sim$ Bernoulli$(\Lambda^{(1)}_{uv})$ and

$$correlation(A_{uv}, B^{(i)}_{uv}) = (R_i)_{uv};$$

iii. All edges in $B \setminus \{B^{(1)}, \cdots, B^{(N)}\}$ are independent and are independent of all edges in $B^{(1)}, \cdots, B^{(N)}$. Furthermore, the collections $\{B^{(i)}_{uv}\}_{i=1}^N$ are independent of each other as $\{u, v\} \in \binom{\{m\}}{2}$ varies, where $\binom{S}{j}$ denotes the collection of all possible subsets of order j from the set S. Note that the edges within each collection $\{B^{(i)}_{uv}\}_{i=1}^N$ can have nontrivial dependence.

In [111], the authors defined a similar Correlated Erdős-Rényi Model, which is a special case of our Multiple Correlated Erdős-Rényi model with the additional assumption that $N = 1$. Allowing N to be greater than one allows us to embed multiple matches of the template A into B, and vary the strengths of the matchings via $\mathscr{R}$. Note that the structure of

the multiple embeddings in Definition 3.1 constrains Λ_2, as multiple copies of Λ_1 need to be embedded into Λ_2 as principal submatrices (up to reordering).

As mentioned earlier, the proposed Multiple Correlated Erdős-Rényi Model extends the classic correlated Erdős-Rényi and the correlated Erdős-Rényi template models of [76, 111]. This novel model explores the new, yet common phenomenon where multiple solutions (here, embedded templates) to the subgraph matching problem could potentially exist, and where certain nodes are more central to the template structure and should be preserved in multiple recovered templates. For example, consider a supply chain graph focusing on key suppliers. Major companies like Nvidia and Intel would be core nodes because their components are crucial across various products, while smaller or more specialized suppliers, whose products are limited to specific areas, would be less likely to exist in multiple templates. That said, Erdős-Rényi models are often not directly applicable for modeling real data, though correlated Erdős-Rényi models are a standard setting for graph matching theory in the literature. Moreover, the proposed algorithms in this work perform well on real network data that is not Erdős-Rényi, see Sections 3.3.3 and 3.3.4 for details.

3.2 Solution diversification

In order to recover signals with suboptimal R_i structure (i.e., sufficiently entry-wise dominated by another R_j), our approach will make use of vertex-based graph features as done in [92]. These features will be represented in the form of a similarity matrix $S = S^{A,B}$ defined as follows.

Definition 3.2. *The similarity matrix between a pair of graphs $A \in \mathscr{G}_m$ and $B \in \mathscr{G}_n$ (where $m < n$) is a matrix $S = S^{A,B} \in [0,\infty)^{m\times n}$, where for $(i,j) \in \{1,\cdots,m\} \times \{1,\cdots,n\}$, we have $S^{A,B}_{ij}$ represents the similarity score between node $i \in V_A$ and $j \in V_B$. When the context is clear, we shall suppress the indices A,B and simply write S.*

In certain cases, when we have knowledge of the networks only limited to edge structures, there is not much we can do besides the standard matched filter approach. See Theorem 3.1 below and [111]. If we have labels or feature vectors for the nodes of the networks, we can in general try to find some proper distance measures to define our S matrix. When no good distance measure can be defined, we can use the multiplex graph matching matched filters proposed in [92].

To incorporate the node similarities into our matching problem, we adapt the approach of [111]. First, for integer k we will let J_k be the $k \times k$ hollow matrix with all off-diagonal entries equal to 1, and $\oplus$ denote the matrix direct sum defined as $M_1 \oplus M_2 = \begin{bmatrix} M_1 & \mathbf{0} \\ \mathbf{0} & M_2 \end{bmatrix}$. Adopting an appropriate padding scheme:

i. The *centered padding* which matches $\tilde{A} = (2A - J_m) \oplus \mathbf{0}_{n-m,n-m}$ to $\tilde{B} = 2B - J_n$; this seeks the best fitting induced subgraph of B to match to A according to the Frobenius norm GMP formulation. As noted in [92], this is equivalent to minimizing $\|A - PBP^T\|_F$ over $P \in \Pi_{m,n}$, where $\Pi_{m,n} = \{P \in \{0,1\}^{m \times n} \text{ s.t. } \mathbf{1}_m^T P \leq \mathbf{1}_n, P\mathbf{1}_n = \mathbf{1}_m\}$.

ii. The *naive padding* which matches $\hat{A} = A \oplus \mathbf{0}_{n-m,n-m}$ to $\hat{B} = B$; this seeks the best fitting subgraph of B to match to A where the objective to minimize $\|\hat{A} - P\hat{B}P^T\|_F$ over $P \in \Pi_n$.

Note that for any $C, D \in \mathscr{G}_n$

$$\text{argmin}_{P \in \Pi_n} \|C - PDP^T\|_F = \text{argmin}_{P \in \Pi_n} \|CP - PD\|_F = \text{argmax}_{P \in \Pi_n} \text{tr}(CPDP^T).$$

The above relation between the Frobenius form of the objective function and the trace form yields that the naive padding scheme is equivalent to maximizing $\text{tr}(APBP^T)$ over $P \in \Pi_{m,n}$. As in [92], we see that the centered padding scheme is equivalent to minimizing $\|A - PBP^T\|_F$ over $P \in \Pi_{m,n}$, which in trace form is equivalent to maximizing

$2\text{tr}(APBP^T) - \|PBP^T\|_F^2$ over $P \in \Pi_{m,n}$. The extra $\|PBP^T\|_F^2$ term incorporates the penalty for edge/non-edge disagreements that distinguishes the centered from the naive padding. We note here that while the above optimization could be cast as optimizing over $\Pi_{m,n}$, this makes the connection between the Frobenius norm and trace-form of the objective function a bit more nuanced across paddings (see also the discussion in [92]). We choose instead to present the optimization over the full permutation matrices Π_n, where naive padding is then finding $\text{argmax}_{P\in\Pi_n}\text{tr}(\hat{A}P\hat{B}P^T)$ and centered padding finding $\text{argmax}_{P\in\Pi_n}\text{tr}(\tilde{A}P\tilde{B}P^T)$ to ease exposition and highlight the connection between the two forms.

From the definition (see also the discussion in [44]), we see that the centered padding scheme penalizes like graph edit distance, with equal penalty for any extra or removed edges in the recovered templates. This is more useful if additional recovered structure is not desired in the recovered template. The naive padding (which only rewards common edges, and does not penalize missing or extraneous edges) should be used if extra edges/activity in the background is unimportant, and the recovery of the template edges is the paramount task. As an example, consider matching the template $A \in \mathcal{G}_m$ that is an $m/2$ regular graph to the graph B, where B is composed of two subgraphs connected by a single edge: one of which is a copy of A with one missing edge (call this B_A), the other the complete graph on m vertices (call this B_K). Centered padding would match A to B_A and naive padding to B_K. Note that if the graphs are weighted, the naive padding is more easily used, as the optimal centered padding scheme for weighted graphs is still an open research topic.

Write $P = \begin{pmatrix} P_{(1)} \\ P_{(2)} \end{pmatrix}$ where $P_{(1)} \in \mathbb{R}^{m\times n}$ and $P_{(2)} \in \mathbb{R}^{(n-m)\times n}$, we account for the similar-

ity term by seeking the solution to one of:

$$\underset{P\in\Pi_n}{\operatorname{argmax}} \quad \underbrace{\operatorname{tr}\left(\tilde{A}P\tilde{B}P^T\right)+\lambda\operatorname{tr}\left(SP_{(1)}^T\right)}_{:=\tilde{f}(P,\lambda)} \quad \begin{pmatrix}\text{centered}\\ \text{padding}\end{pmatrix}$$

$$\underset{P\in\Pi_n}{\operatorname{argmax}} \quad \underbrace{\operatorname{tr}\left(\hat{A}P\hat{B}P^T\right)+\lambda\operatorname{tr}\left(SP_{(1)}^T\right)}_{:=\hat{f}(P,\lambda)} \quad \begin{pmatrix}\text{naive}\\ \text{padding}\end{pmatrix}$$

where λ is a hyperparameter chosen/tuned by the user, S is the similarity matrix as defined in Definition 3.2 and $P_{(1)}$ is the matrix consisting only the top m rows of the matrix P. The `GMMF` approach then uses multiple random restarts of the following procedure to search B for the best fitting subgraphs to A. We will present the algorithm, adapted from [44, 126], incorporating the gradient of the feature term while ignoring the seeded portion, in the centered padding case, the naive padding setting following mutatis mutandis.

1. Apply centered padding to A and B yielding $\tilde{A}$ and $\tilde{B}$;

2. Considering N_{mc} random restarts,

for $k = 1,2,\cdots,N_{mc}$, do the following

i. Set initialization $P^{(0)} = \gamma\mathbf{1}_k\mathbf{1}_k^T/n+(1-\gamma)P$ where $P\sim$ Unif(Π_n) and $\gamma\sim$Unif[0,1];

ii. While $\|P^{(t)}-P^{(t-1)}\|_F > \eta$ for a specified tolerance $\eta > 0$, do the following

a. Compute the gradient

$$\nabla_P\tilde{f}(P^{(t)},\lambda) = \tilde{A}^TP^{(t)}\tilde{B}+\tilde{A}P^{(t)}\tilde{B}^T+\lambda I_{n\times m}S$$

where $I_{n\times m}$ is the matrix consisting with the first m columns of an $n\times n$ identity matrix.

b. Compute search direction

$$Q^{(t)} = \text{argmax}_{Q \in \mathscr{D}_n} \text{tr}\left[\nabla_P \tilde{f}(P^{(t)}, \lambda)^\top Q\right];$$

via the Hungarian Algorithm proposed in [62] where $\mathscr{D}_n$ is the set of $n \times n$ doubly stochastic matrices;

c. Perform line search in the direction of $Q^{(t)}$ by solving

$$\gamma^* = \text{argmax}_{\gamma \in [0,1]} \tilde{f}(\gamma P^{(t)} + (1-\gamma)Q^{(t)}, \lambda)$$

This step involves optimizing a quadratic function of γ, and an analytical solution is obtained by taking the derivative to find the critical point, followed by comparing the function values at the two boundary points and the critical point.

d. Set $P^{(t+1)} = \gamma^* P^{(t)} + (1-\gamma^*)Q^{(t)}$

iii. Set $P^{(*,k)} = \max_{P \in \Pi_n} \text{tr}\left(P^\top P^{(\text{final})}\right)$;

3. Rank the recovered matchings $\{P^{(*,k)}\}_{k=1}^{N_{mc}}$ by largest to smallest value of the objective function $\tilde{f}(P, \lambda)$; output the ranked list of matches.

In the above algorithm, we can steer the algorithm away from previously recovered solutions (in the random restarts) by biasing the objective function away from these already recovered solutions. Suppose that the k-th random restart returns the solution $P^{(*,k)}$ (with corresponding permutation $\sigma^{(*,k)}$). To accomplish this, we define the mask $M^{k,\varepsilon} \in \mathbb{R}^{m,n}$ via

$$M_{ij}^{k,\varepsilon} = \begin{cases} (1-\varepsilon) & \text{if } j = \sigma^{(*,k)}(i); \\ 1 & \text{else.} \end{cases}$$

As an example, consider $[n] = \{1, 2, \ldots, n\}$; $\sigma^{(*,1)}$ maps $[3] \mapsto [7]$ by fixing $[3]$ identically;

and $\sigma^{(*,2)}$ maps $[3] \mapsto [7]$ by $\sigma^{(*,2)}(1) = 1$, $\sigma^{(*,2)}(j) = j+3$ for $j = 2,3$. Then

$$M^{1,\varepsilon} = \begin{bmatrix} 1-\varepsilon & 1 & 1 & \mathbf{1}_{1\times 4} \\ 1 & 1-\varepsilon & 1 & \mathbf{1}_{1\times 4} \\ 1 & 1 & 1-\varepsilon & \mathbf{1}_{1\times 4} \end{bmatrix}$$

$$M^{2,\varepsilon} = \begin{bmatrix} 1-\varepsilon & \mathbf{1}_{1\times 3} & 1 & 1 & 1 \\ 1 & \mathbf{1}_{1\times 3} & 1-\varepsilon & 1 & 1 \\ 1 & \mathbf{1}_{1\times 3} & 1 & 1-\varepsilon & 1 \end{bmatrix}$$

In the next random restart, we apply the mask to the current similarity matrix $S^{(k,\varepsilon)}$ via $S^{(k+1,\varepsilon)} = M^{k,\varepsilon} \circ S^{(k,\varepsilon)}$ (note: $S^{(1,\varepsilon)} = S$) where "$\circ$" represents the matrix Hadamard product. Considering the previous example, let S be a 3×7 matrix, then

$$\begin{aligned} S^{(3,\varepsilon)} &= M^{1,\varepsilon} \circ M^{2,\varepsilon} \circ S \\ &= \begin{bmatrix} (1-\varepsilon)^2 S_{11} & S_{12} & S_{13} & S_{14} & S_{15} & S_{16} & S_{17} \\ S_{21} & (1-\varepsilon)S_{22} & S_{23} & S_{24} & (1-\varepsilon)S_{25} & S_{26} & S_{27} \\ S_{31} & S_{32} & (1-\varepsilon)S_{33} & S_{34} & S_{36} & (1-\varepsilon)S_{36} & S_{37} \end{bmatrix} \end{aligned}$$

After penalization, we then seek to solve one of

$$\underset{P\in\Pi_n}{\operatorname{argmax}} \quad \underbrace{\operatorname{tr}\left(\tilde{A}P\tilde{B}P^T\right) + \lambda \operatorname{tr}\left(S^{(k+1,\varepsilon)} P_{(1)}^T\right)}_{:=\tilde{f}_{\varepsilon,k+1}(P,\lambda)} \begin{pmatrix} \text{centered} \\ \text{padding} \end{pmatrix} \tag{3.1}$$

$$\underset{P\in\Pi_n}{\operatorname{argmax}} \quad \underbrace{\operatorname{tr}\left(\hat{A}P\hat{B}P^T\right) + \lambda \operatorname{tr}\left(S^{(k+1,\varepsilon)} P_{(1)}^T\right)}_{:=\hat{f}_{\varepsilon,k+1}(P,\lambda)} \begin{pmatrix} \text{naive} \\ \text{padding} \end{pmatrix} \tag{3.2}$$

The masks effectively slightly down-weight the similarity scores for the previously recovered matrices. Note that an overly draconian choice of $\varepsilon \approx 1$ may have the effect of

steering the algorithm away entirely from recovered solutions, and might not allow for overlapping solutions to be returned. This is suboptimal in the case where a few key edges/vertices are expected to appear in many recovered templates.

Our algorithm uses a simple gradient descent-based optimization, which is computationally fast when compared to more complex approximations to NP-hard GMP solution. To avoid the pitfalls of first-order methods (e.g., local maxima) that could lead to suboptimal solutions, we, in practice, run the algorithm multiple times (easily parallelized) with random starting points as our Monte Carlo simulations. This allows us to better explore the objective function and leverages the speed of each Frank-Wolfe iterate.

Remark 3.1. *We can also apply the mask $M^{k,\varepsilon}$ directly to the gradient or to the initialization in the* `GMMF` *algorithm outlined above. In the gradient penalizing case, step (a.) for the $(k+1)$-st random restart becomes*

$$\nabla_P^{(k+1),\varepsilon} \tilde{f}(P^{(t)},\lambda) = M^{1,\varepsilon} \circ \cdots \circ M^{k,\varepsilon} \circ \left(\tilde{A}^T P^{(t)} \tilde{B} + \tilde{A} P^{(t)} \tilde{B}^T + \lambda I_{n\times m} S\right)$$

In the initialization penalizing, the mask is directly applied to $P^{(0)}$ followed by rescaling to ensure double stochasticity. Similar ideas of penalizing the gradient to diversify solutions when solving optimization problems, particularly to find weaker or flatter optimizers, exist in the literature (e.g., [14, 137]). We will consider only the similarity penalization in the theory below because this approach can be easily incorporated into the analysis of the graph matching objective function without need for delving into the optimization steps. We also note that the penalty constant ε could be replaced by a sequence of penalties $\{\varepsilon_k\}$ if we are expecting high level of template overlap in the embedded templates. Yet another approach would be to penalize the edge weights of the recovered templates in the large network. Assuming a high correlation between edge structures and node similarities, the result should be similar to our approach. However, the efficiency of penalizing edges could

be suboptimal due to the significantly higher number of edges compared to nodes in the denser regime.

3.2.1 Theoretical benefits of down-weighting

We next provide theoretical justification for the down-weight masking in the setting where there are two overlapping embedded templates in the background (i.e., where $N = 2$ in Definition 3.1). The case where $N > 2$ follows from repeated applications of the case where $N = 2$, as does the case of no template overlap. In the case where we expect to find only one recovered template, [111] provides a detailed characterization of methods and conditions for detecting such a recovered template with high probability (sans similarity S). Throughout the rest of the manuscript, all graphs and parameters should be indexed by the number of nodes in the larger graph, n. To improve readability, we will suppress the subscript n whenever the context is clear.

We consider Multiple Correlated Erdős-Rényi graphs with the following structure. We will consider $\Lambda_1 = pJ_m$ and $\Lambda_2 = pJ_n$. Note that our theories can be easily extended to inhomogeneous Erdős-Rényi graphs by using 0 and 0.25 as the lower and upper bounds of the variance of any Bernoulli random variables. While in practice, almost no network is purely Erdős-Rényi, such models are particularly useful for theoretically studying matchability phase transitions, and are a standard setting for deriving graph matching theoretical results [30, 31, 76, 132]. Indeed, homogeneous Erdős-Rényi provides a difficult theoretical setting as there is no heterogeneity correlation across graphs [42], and the matching signal is entirely contained in the edge-correlation R. If there was signal in both the edge structure Λ and in R, as is common in most real-world networks, the practical difficulty would likely be reduced significantly as additional statistics (e.g., degree sequence, graph bottlenecks, centrality, etc.) could be more easily leveraged to match the template. Here, we consider

$$A = \begin{array}{c} \\ m-k \\ k \end{array} \begin{array}{c} \begin{array}{cc} m-k & k \end{array} \\ \begin{pmatrix} A^{11} & A^{12} \\ (A^{12})^T & A^{22} \end{pmatrix} \end{array}; \qquad B = \begin{array}{c} \\ 2m-k \\ n-2m+k \end{array} \begin{array}{c} \begin{array}{cc} 2m-k & n-2m+k \end{array} \\ \begin{pmatrix} C & D \\ D^T & E \end{pmatrix} \end{array}$$

where

$$C = \begin{array}{c} \\ m-k \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{ccc} m-k & k & m-k \end{array} \\ \begin{pmatrix} C^{11} & C^{12} & C^{13} \\ (C^{12})^T & C^{22} & C^{23} \\ (C^{13})^T & (C^{23})^T & C^{33} \end{pmatrix} \end{array}$$

Further, for real numbers $0 < r_3 < r_2 < r_1 < 1$,

$$\text{corr}(A_{ij}, B_{h\ell}) = \begin{cases} r_1 & \text{if } i = h,\ j = \ell,\ i \le m,\ j \le m-k \\ r_1 & \text{if } i = h,\ j = \ell,\ i \le m-k,\ j \le m \\ r_2 & \text{if } i = h,\ j = \ell,\ m-k < i \le m,\ m-k < j \le m \\ r_3 & \text{if } i = h-m,\ j = \ell-m,\ i \le m-k,\ j \le m-k \\ r_3 & \text{if } i = h-k,\ j = \ell-m+k,\ m-k < i \le m, \\ & \quad j \le m-k \\ r_3 & \text{if } i = h-m+k,\ j = \ell-k,\ i \le m-k, \\ & \quad m-k < j \le m \\ 0 & \text{else} \end{cases}$$

This structure ensures that the two embedded copies of the template (each of size m) have a non-trivial overlap of size $k > 0$. The overlap again is designed to model the case where

there are key vertices in the background that appear in multiple template embeddings. Moreover, the case where $k = 0$ is conceptually and theoretically simpler than the $k > 0$ case and follows as a corollary to the theory below.

Observe that with shuffling channels $Q^{(A)} \in \Pi_m$ and $Q^{(B)} \in \Pi_n$ applied to matrices A and B, our model can account for Multiple Correlated Erdős-Rényi graphs such that two embeded templates exist with a shared sub-region of all kinds, aligning with the assumptions that more than one solutions exist and certain nodes must be preserved in all solutions.

We will consider the centered padding scheme, where we match $\widetilde{A} = (2A - J_m) \oplus \mathbf{0}_{n-m,n-m}$ to $\widetilde{B} = (2B - J_n)$. Analogous results can be derived for the naive scheme, which we leave to the reader. In this setting, we will consider S of the form

$$S = \begin{array}{c} \\ m-k \\ k \end{array} \begin{array}{c} \begin{array}{cccc} m-k & k & m-k & n-2m+k \end{array} \\ \begin{pmatrix} S^{11} & S^{12} & S^{13} & S^{14} \\ S^{21} & S^{22} & S^{23} & S^{24} \end{pmatrix} \end{array}$$

where all entries of S are independent and bounded, without loss of generality bounded in $[0,1]$ (e.g., Beta distributed), random variables, and where

$$\begin{aligned} &\text{the diagonal elements of } S^{11} \text{ have mean } \mu_1 \\ &\text{the diagonal elements of } S^{22} \text{ have mean } \mu_2 \\ &\text{the diagonal elements of } S^{13} \text{ have mean } \mu_3 \end{aligned}$$

and all other entries have mean μ_4. Here we will assume that $\mu_1 > \mu_2 > \mu_3 > \mu_4$. Let P^*

and $\tilde{P}$ map A to

$$\begin{array}{c} \\ m-k \\ k \end{array}\begin{array}{c} \begin{array}{cc} m-k & k \end{array} \\ \begin{pmatrix} C^{11} & C^{12} \\ (C^{12})^T & C^{22} \end{pmatrix} \end{array}, \text{ and } \begin{array}{c} \\ m-k \\ k \end{array}\begin{array}{c} \begin{array}{cc} m-k & k \end{array} \\ \begin{pmatrix} C^{33} & (C^{23})^T \\ C^{23} & C^{22} \end{pmatrix} \end{array},$$

respectively, so that of the two embedded templates, the embedding via P^* is stronger (in that the correlation is higher entry-wise as are the similarity scores on average) than that provided by $\tilde{P}$.

The goal is to down-weight/penalize the strongly embedded template so that the optimal solution to Eq. 3.1 is $\tilde{P}$ as opposed to P^*. The features here are the key, as without this ability to down-weight the features (or the gradient) we do not expect to find $\tilde{P}$ by solving Eq. 3.1. Indeed, if we can only observe the edges of A and B, without any further information provided by S the results of [111] provides the following theorem:

Theorem 3.1. *Let A and B be graphs as described above. Assuming we can only observe the edges of A and B but have no additional knowledge about the vertex-based graph features, then with probability at least $1-n^{-2}$, we have that $\underset{P\in\Pi_n}{\operatorname{argmax}} \operatorname{tr}\left(\tilde{A}P\tilde{B}P^T\right) = \{P^*\}$.*

If the strongly embedded template is penalized, then S is weighted as follows. Here, $S^{(2,\varepsilon)}$ is set to be (where "$\circ$" is the matrix Hadamard product)

$$\begin{array}{c} \\ m-k \\ k \end{array}\begin{array}{c} \begin{array}{cccc} m-k & k & m-k & n-2m+k \end{array} \\ \begin{pmatrix} (\mathbf{1}_{m-k,m-k}-\varepsilon I_{m-k})\circ S^{11} & S^{12} & S^{13} & S^{14} \\ S^{21} & (\mathbf{1}_{k,k}-\varepsilon I_k)\circ S^{22} & S^{23} & S^{24} \end{pmatrix} \end{array}$$

We next state our main result, which is for the Multiple Correlated Erdős-Rényi model where the entries of S are independent, and bounded (in $[0,1]$). Note the proof can be

found in Section 3.5.1.

Theorem 3.2. *Let A and B be two graphs constructed as above. If there is a constant $\alpha \in [1/2, 1)$ such that*

i. $m - k = \Theta(m)$; $m^{1-\alpha} = \omega(\log^4 n)$

ii. $\lambda = m^{\alpha}$;

iii. $(r_1 - r_3) \ll m^{\alpha-1}$; r_3, r_2, r_1 *are bounded away from* 0 *and* 1;

iv. $\mu_3 > (1-\varepsilon)\mu_1$ *and* $(1-\varepsilon)\mu_2 > \mu_4$; *the differences* $\mu_3 - (1-\varepsilon)\mu_1$, $(1-\varepsilon)\mu_2 - \mu_4$, *and* $\mu_3 - \mu_4$ *are bounded away from 0;*

v. p *is bounded away from* 0 *and* 1;

then if $\widetilde{\Pi}$ is the set of permutations perfectly aligning the weakly embedded template (i.e., of the form $\tilde{P} \oplus Q$), we have

$$\mathbb{P}(argmax_{P \in \Pi} \tilde{f}_{\varepsilon,1}(P, \lambda) \subset \widetilde{\Pi}) \geq 1 - e^{-\omega(\log n)}.$$

Note that when $\lambda = 0$, the objective function becomes the standard one considered in [111], and Theorem 3.1 applies. However, when $\lambda > 0$ is sufficiently large, the feature similarity becomes crucial in the objective function. By increasing ε, we force the global optimizer to move away from the first recovered template, thus expecting to recover a different in-sample subgraph. When λ is too large, the noise in the $S^{(2,\varepsilon)}$ matrix (provided by the entries with mean μ_4) can swamp the signal and suboptimal recovery is possible. Note also that there are myriad combinations of parameter growth conditions under which Theorem 1 (or analogues) will hold, and our theorem is not claiming full generality. We will not mine these conditions further herein. We point out here that p as a function of n

being bounded away from 0 implies that the graph we consider here is dense, as in much of the other graph matching literature. Additionally, we could encode the diameter of the graphs into our objective function to eliminate disconnected matches, if any occur.

By properly realizing the asymptotic notation of ω as a fixed function and thus expressing m in terms of $\log n$, we can easily translate our results into a finite sampling result. Note that most papers discussing graph matchabilities in the literature prove their results in an asymptotic context, meaning that the asymptotic behavior tells us what to expect for large enough networks, as seen in real data.

Remark 3.2. *Beyond independent entries for S, we could consider S as a similarity between vertex features. One possible approach to define such a similarity matrix S is via well-constructed distance functions. In these cases, a function of two variables $s(a,b)$ (resp., $d(a,b)$ for a distance/dissimilarity function, where, for example, we could then define $s(a,b) = 1/d(a,b)$) is defined such that s increases (resp., d decreases) as $|a-b|$ becomes smaller, e.g., $s(a,b) = e^{-|a-b|}$. For each node $i \in V_1 = \{1,2,\cdots,m\}$ and $j \in V_2 = \{1,2,\cdots,n\}$, if we model random vertex features via $X_i, Y_j \sim F$, we can then define $S_{ij} = s(X_i, Y_j)$ for all i, j. Note that Theorem 3.2 relies only on the fact that the means of the Beta random variables are well-separated, so it can be easily adapted to account for these vertex-dependent similarities, as long as the scores satisfy certain tail decay conditions and the expectation of similarity scores for node pairs under different recovery schemes are bounded away from each other.*

We have developed an approach for recovering weaker signal versions of the embedded templates with high probability, under mild conditions. Our approach builds upon the method introduced in [111], where we initially recover the strongest in-sample template. Subsequently, we select a sufficiently large value for λ and gradually increase ε until we identify an additional in-sample template. An important aspect is that once a second recov-

ered template is found, we can iterate the algorithm by penalizing the similarity for both recovered templates. This allows us to discover more templates until we have exhausted all possibilities or until the penalty coefficient term for overlapping signal in unrecovered templates becomes too small compared to the noise in the non-signal nodes.

We next proceed to further demonstrate the validity of our proposed method via experiments in the Multiple Correlated Erdős-Rényi model and via two real data experiments.

3.3 Experimental results

Our proposed modification of the `GMMF` algorithm offers a simple and efficient solution for solution diversification. Moreover, incorporating a multiple of the tr $\left(SP_{(1)}^T\right)$ term into the existing Frank-Wolfe iterations allows us to efficiently incorporate the feature information. Next, we will address scalability issues arising from the line search in steps (ii.a) and (ii.b) of the `GMMF` algorithm. To tackle this, note that the gradient $\nabla_P \tilde{f}(P^{(t)}, \lambda)$ forms an $m \times n$ matrix. In practice, we often have $m \ll n$ (i.e., m is significantly smaller than n). In the linear assignment search step (ii.b), each vertex in A can only be assigned to one of the vertices in B with the top m values in the gradient. Therefore, for the allowable matchings for vertex i in A, we compute the m largest entries of row i of $\nabla_P \tilde{f}(P^{(t)}, \lambda)$. All other row i entries can be discarded. After performing this partial sorting operation for each of the m rows (each row costing $O(n)$ time using the `Introselect` algorithm [85]) the resulting matrix of allowed assignments is at most $m \times m^2$ and the Hungarian algorithm [62] applied to the Linear Assignment Problem (LAP) on this rectangular matrix is of complexity $O(m^4)$ [98]. Moreover, solving the LAP on this reduced matrix ensures the same result as solving it on the full $\nabla_P \tilde{f}(P^{(t)}, \lambda)$. This approach effectively reduces the complexity of the LAP solver subroutine from $O(n^3)$ to $O(mn + m^4)$, and we observe substantial speedups in practice. We note that this complexity reduction technique was also discussed in [20].

The discussion above focuses only on the LAP subroutine of the `GMMF` algorithm since it is often a computational bottleneck in the procedure.

Due to the intractability of computing the exact graph matching solution in all but small cases, in a few of the experiments below we make use of seeded vertices in the graph matching subroutine. Seeded vertices are those whose correct alignment is known a priori. In this case (assuming for the moment that the seeding maps vertices $\{1,2,\ldots,s\}$ in A to $\{1,2,\ldots,s\}$ in B), the graph matching seeks to optimize $\tilde{f}(P,\lambda)$ over P of the form $P = I_s \oplus Q$ (or $\hat{f}(P,\lambda)$ in the naive padding case) using the `SGM` algorithm proposed in [44]. In practice, seeds are often expensive to compute—indeed in the real data experiment on template detection in knowledge graphs below, we have no seeded vertices—though a few seeds can often lead to dramatically increased performance. In the simulations below, the seeded approach helps overcome the computational intractability of the matching subroutine and are quite useful for demonstrating the utility of our solution diversification step.

3.3.1 Two overlapping templates

Our first experiment verifies the proposed algorithm in our multiple correlated Erdős-Rényi model with $N = 2$. In our setup, we take $\Lambda^{(1)} = \mathbf{0.8}_{50\times 50}, \Lambda^{(2)} = \mathbf{0.8}_{500\times 500}$. To generate graphs with tractable edge correlations, we create an Erdős-Rényi background graph with edge probability u. We then sample induced graphs from the background graph by flipping the existence of each edge independently with probability v. Appendix 7.2 of [70] provides a formula for the correlation value as a function of the edge existence probability of the background graph, u, and the edge flipping probability, v. Choosing v to be $0.0074, 0.0168, 0.0326$ and 0.0495 yields the correlation values used in our follow-up experiments as approximately $0.954, 0.897, 0.803$ and 0.706, respectively (due to computational precision, these are estimates of $0.95, 0.9, 0.8, 0.7$ resp.). Now let

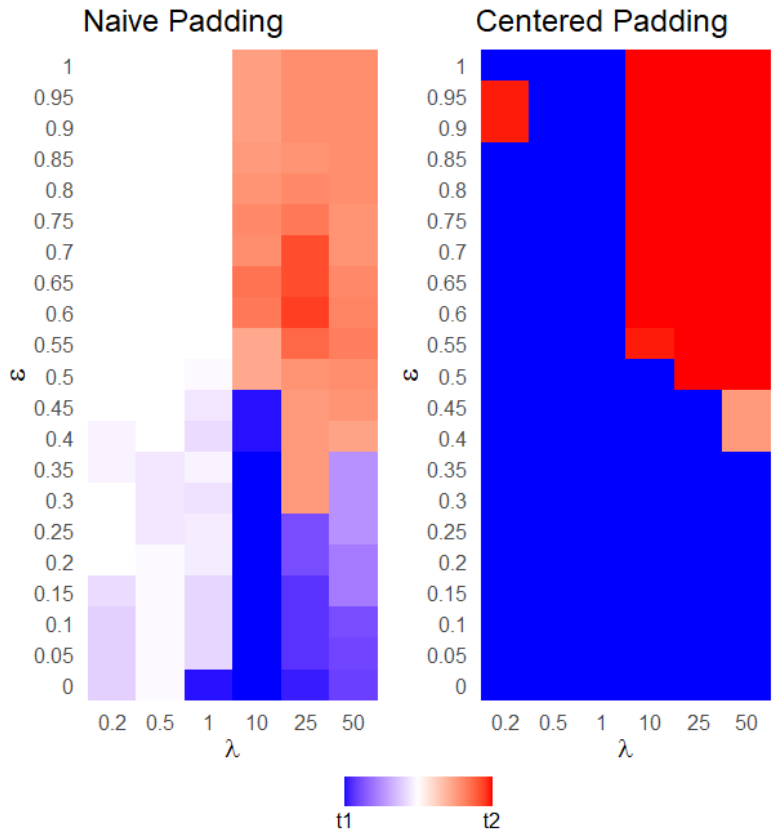

Figure 3.1: We fix $k = 10$ and use the seeded `GMMF` algorithm to match A with B using 5 seeds randomly selected from the overlapping nodes of $B^{(1)}$ and $B^{(2)}$ as described in Section 3.3.1. We plot the recovering results over ε (here ε is used to penalize the stronger of the two embedded templates) and λ, averaged by 20 Monte-Carlo simulations. In the figures, stronger colors represent better recovery of the embedded templates, and t1 (blue) stands for template 1, t2 (red) stands for template 2, with white squares corresponding to the case when none of the two templates was recovered or equal amounts of each template were recovered among the 20 simulations.

$$R_1 = \begin{array}{c} \\ m-k \\ k \end{array} \begin{array}{c} \begin{array}{cc} m-k & k \end{array} \\ \begin{pmatrix} R_1^{11} & R_1^{12} \\ (R_1^{12})^T & R_1^{22} \end{pmatrix} \end{array}; R_2 = \begin{array}{c} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{cc} k & m-k \end{array} \\ \begin{pmatrix} R_2^{22} & R_2^{12} \\ (R_2^{12})^T & R_2^{22} \end{pmatrix} \end{array}$$

where R_1^{11} and R_1^{12} are matrices with all entries set to 0.954; R_2^{12} and R_2^{22} are matrices

with all entries set to 0.803; and R_1^{22} is a matrix with all entries set to 0.897. Note the overlap in R_1 and R_2 is there to make sure the induced subgraphs $B^{(1)}$ and $B^{(2)}$ have k overlapping nodes. For the similarity matrix S, we set

$$S = \begin{array}{c} \\ m-k \\ k \end{array} \begin{array}{c} \begin{array}{cccc} m-k & k & m-k & n-2m+k \end{array} \\ \begin{pmatrix} S^{11} & S^{12} & S^{13} & S^{14} \\ S^{21} & S^{22} & S^{23} & S^{24} \end{pmatrix} \end{array}$$

where all entries of S are independent Beta random variables, such that the diagonal elements of S^{11} are i.i.d. Beta($\mu_1 = 0.6$); the diagonal elements of S^{22} are i.i.d. Beta($\mu_2 = 0.55$); the diagonal elements of S^{13} are i.i.d. Beta($\mu_3 = 0.5$); and all other entries are sampled i.i.d. from Beta($\mu_4 = 0.1$). Note that for Beta distribution with parameters α, β, we have $\mu = \frac{\alpha}{\alpha+\beta}$. We randomly sample $\alpha \sim U(0,1)$ and use the specified μ to calculate the corresponding β. Other combinations of parameters are also explored and plotted, see Section 3.6.1.

We fix $k = 10$ (see Section 3.6.1 for the case of $k = 15, 40$), and use the seeded `GMMF` algorithm with 5 seeds randomly selected from the overlapping nodes of $B^{(1)}$ and $B^{(2)}$. In Figure 3.1 we use the naive padding (left) and the centered padding (right) and plot results over numerous choices of ε (here ε is used to penalize the stronger of the two embedded templates) and λ, averaged over 20 Monte-Carlo simulations. In the figures, stronger colors represent better recovery of the embedded templates, and t1 (blue) stands for template 1, t2 (red) stands for template 2, with white squares corresponding to the case when none of the two templates was recovered or equal amounts of each template were recovered among the 20 simulations. From the figures, we see that when ε is small we recover the stronger embedded template (B_1), and as ε increases we move away from the stronger embedded template, and—provided a suitable value of λ—we successfully recover the weaker embed-

ded template as desired. From the plots, we can see the centered padding outperforms the naive padding for recovering the second template in this multiple correlated Erdős-Rényi model, which aligns with the results proven in [111]. The phenomenon is clearer for larger k (see, for example, Figure 3.6.1 in Section 3.6.1), where the naive padding detects either only template 1 or nothing (denoted by the white in the plot) for whatever λ we choose.

3.3.2 Three overlapping templates

To better illustrate the iterative feature of the proposed algorithm, we construct a multiple correlated Erdős-Rényi model with $N = 3$ and apply our algorithm to attempt to recover all of the three embedded templates. In this experiment, we take $\Lambda^{(1)} = \mathbf{0.8}_{50\times 50}, \Lambda^{(2)} = \mathbf{0.8}_{500\times 500}$,

$$R_1 = \begin{matrix} \\ m-k \\ k \end{matrix} \begin{matrix} \begin{matrix} m-k & k \end{matrix} \\ \begin{pmatrix} R_1^{11} & R_1^{12} \\ (R_1^{12})^T & R_1^{22} \end{pmatrix} \end{matrix}; R_2 = \begin{matrix} \\ k \\ m-k \end{matrix} \begin{matrix} \begin{matrix} k & m-k \end{matrix} \\ \begin{pmatrix} R_1^{22} & R_2^{12} \\ (R_2^{12})^T & R_2^{22} \end{pmatrix} \end{matrix}; R_3 = \begin{matrix} \\ k \\ m-k \end{matrix} \begin{matrix} \begin{matrix} k & m-k \end{matrix} \\ \begin{pmatrix} R_1^{22} & R_3^{12} \\ (R_3^{12})^T & R_3^{22} \end{pmatrix} \end{matrix},$$

where R_1^{11} and R_1^{12} are matrices with all entries set to 0.954; R_2^{12} and R_2^{22} are matrices with all entries set to 0.803; R_3^{12} and R_3^{22} are matrices with all entries set to 0.706; and R_1^{22} is a matrix with all entries set to 0.897. Note again this structure ensures the induced subgraphs $B^{(1)}, B^{(2)}$ and $B^{(3)}$ have exactly k pairwise overlapping nodes, and to make the $B^{(1)}$ a better probabilistic match (i.e., a stronger embedding) than $B^{(2)}$ which is a better probabilistic match than $B^{(3)}$.

For the similarity matrix S, we set S to be

$$
\begin{array}{c} \\ m-k \\ k \end{array}
\begin{array}{c}
\begin{array}{ccccc} m-k & k & m-k & m-k & n-3m+2k \end{array} \\
\begin{pmatrix} S^{11} & S^{12} & S^{13} & S^{14} & S^{15} \\ S^{21} & S^{22} & S^{23} & S^{24} & S^{25} \end{pmatrix}
\end{array}
$$

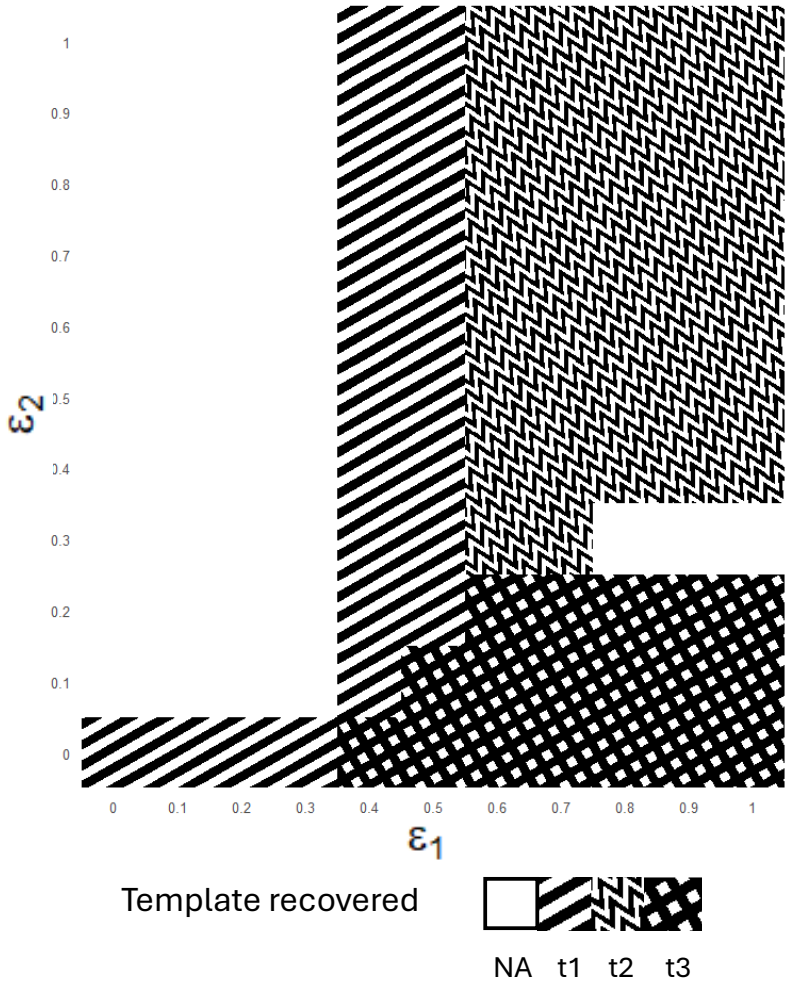

Figure 3.2: We fix $k = 10, \lambda = 25$ and use the seeded `GMMF` algorithm with the centered padding to match A with B using 5 seeds randomly selected from the overlapping nodes of $B^{(1)}, B^{(2)}$ and $B^{(3)}$, where $B^{(1)}, B^{(2)}$ and $B^{(3)}$ are induced subgraph of B such that graphs A and B follows multiple correlated ER model as described in Section 3.3.2. We plot the recovering results over ε_1 (penalty applied to the diagonal elements of $S^{(11)}, S^{(22)}$) and ε_2 (penalty applied to the diagonal elements of $S^{(13)}, S^{(22)}$), averaged by 20 Monte-Carlo simulations. In the figure, the different patterns represent which template was recovered (in majority): t1 for template 1, t2 for template 2, and t3 for template 3, with white squares corresponding to the case when none of the three templates was recovered.

where all entries of S are independent Beta random variables, such that the diagonal elements of S^{11} are i.i.d. Beta($\mu = 0.7$), the diagonal elements of S^{22} are i.i.d. Beta($\mu = 0.6$), the diagonal elements of S^{13} are i.i.d. Beta($\mu = 0.55$), and the diagonal elements of S^{14} are i.i.d. Beta($\mu = 0.5$), and all other entries are sampled from Beta($\mu = 0.1$). Again for all the Beta distributions, we randomly sample $\alpha \sim U(0,1)$ and use the specified μ to calculate the corresponding β.

In Figure 3.2, we fix $k = 10, \lambda = 25$ (see Section 3.6.2 for other combinations of (k, λ)), and use the seeded `GMMF` algorithm with the centered padding (the naive padding behaved sub-optimally for recovering template 3, see Section 3.6.2 for details) and 5 seeds randomly selected from the overlapping nodes of $B^{(1)}, B^{(2)}$ and $B^{(3)}$. We plot the recovering results over ε_1 (penalty applied to the diagonal elements of S^{11}, S^{22}) and ε_2 (penalty applied to the diagonal elements of S^{13}, S^{22}), averaged by 20 Monte-Carlo simulations. In the figure, the different patterns represent which template was recovered (in majority): t1 for template 1, t2 for template 2, and t3 for template 3, with white squares corresponding to the case when none of the three templates was recovered. Note that when $\varepsilon_1 \in [0, 0.3]$, we have not recovered template 2, thus it would be impossible to have a second penalty term associated with ε_2, the corresponding parts are left white and should not be interpreted. As we can see, when both $\varepsilon_1, \varepsilon_2$ are small, we recover the strongest embedded template; as ε_1 increases, we move away from the strongest embedded template to the second strongest embedded template; finally when both $\varepsilon_1, \varepsilon_2$ get large enough, we recovered the third embedded template as desired.

3.3.3 MRI Brain data

We now apply the proposed algorithm to a real data set of human connectomes from [139], where we consider the BNU1 test-retest connectomes processed via the pipeline

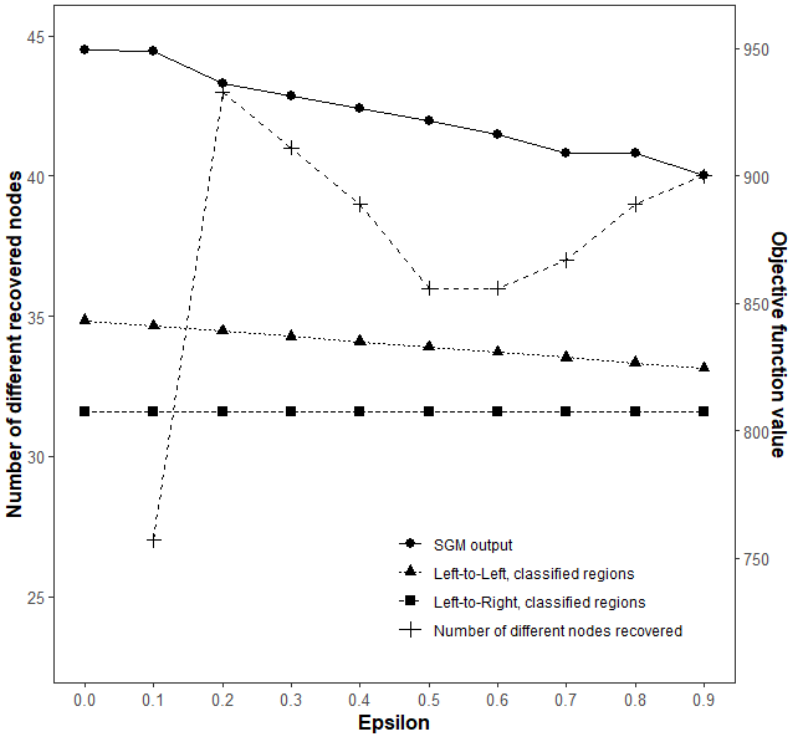

Figure 3.3: We run our proposed algorithm using the seeded `GMMF` algorithm with 500 restarts and 5 seeds selected from the node pairs $\{(s_j, w_j)\}_{j=1}^{6}$ as described in 3.3.3, taking the result with highest objective function value (Eq. 3.2) as the output. For each ε, we compute the GM objective function value (left axis) of the resulting matrix with the template; we also computed the objective function value with respect to the alignment given by the template to the same classified brain region in the left hemisphere in B (Left–to–Left in the plot), as well as the objective function value given by the template to the symmetric region from the right hemisphere in B (Left–to–right in the plot). Also for $\varepsilon > 0$, we calculated the number of novel nodes recovered in each matching compared to the subgraph detected with $\varepsilon = 0$ (right axis).

at [58] (see `http://fcon_1000.projects.nitrc.org/indi/CoRR/html/bnu_1.html` and `https://neurodata.io/mri/` for more detail). The dataset consist of test-retest DTI data for each patient processed into connectome graphs. Moreover, the brain graphs here are segmented into regions of interest and contain (x, y, z) DTI coordinates. We chose patient "subj1" for our present experiment. In the first connectome, which contains 1128 nodes, we select a region (region 30) with 46 nodes from the left hemisphere to act as our template. We then extracted the subgraph induced by the selected region and considered

it as our graph A. The entire graph of the other brain scan from the same patient, which contains 1129 nodes, was designated as our graph B. Our goal is to recover both the same region of interest in the left hemisphere (we consider this the "strong" embedded template) and the corresponding region (region 65) in the right hemisphere (the "weak" embedded template).

To construct the similarity matrix, we consider the (x, y, z) coordinates of each node in the processed MRI scan from graphs A and B. Subsequently, we randomly selected 12 nodes, denoted as $s = \{s_1, \ldots, s_{12}\}$, from graph A. For nodes $s_1, \ldots, s_6$, we identified the corresponding nodes $w_1, \ldots, w_6$ from the same region, same hemisphere, in graph B. Whereas for nodes $s_7, \ldots, s_{12}$, we identified the corresponding nodes $w_7, \ldots, w_{12}$ from the corresponding region in the other hemisphere relative in graph B. Note, these are not seeded vertices, but are simply used to construct the similarity matrix S. Informally, we consider each pair of (s_i, w_i) as a "bridge" which has distance 0; allowing us to define a suitable distance across hemispheres for any nodes $u \in A$ and $v \in B$ as follows. Setting the distance between corresponding seeded nodes across hemispheres to be 0, we define the distance via

$$d(u, v) = \min_{j \in [12]} \{\|u - s_i\|_2 + \|v - w_i\|_2\}$$

Here, $\|\cdot\|_2$ represents the standard Euclidean distance. We defined our similarity matrix such that $S_{ij} \propto -d(u_i, v_j)$.

We executed our proposed algorithm using the seeded `GMMF` algorithm with 500 restarts, with 5 seeds selected out of the node pairs $\{(s_j, w_j)\}_{j=1}^{6}$, so the seeds exist in the left-hemisphere only; and we selected the result with the highest objective function value (in Eq. 3.2 using naive padding with $\lambda = 0.1$ as the final match; see Section 3.6.3 for the case of $\lambda = 1$). Naive padding worked well with the irregular structure of the brain networks here, and we are actively researching whether centered or naive padding is more appropriate

in non edge-independent models. For each ε, (plotted in Figure 3.3) we compute the GM objective function value (right axis) of the resulting matrix with the template; we also computed the objective function value with respect to the alignment given by the template to the same classified brain region in the left hemisphere in B (Left–to–Left in the plot), as well as the objective function value given by the template to the symmetric region from the right hemisphere in B (Left–to–Right in the plot). Also for $\varepsilon > 0$, we calculated the number of novel nodes recovered in each matching compared to the subgraph detected with $\varepsilon = 0$ (left axis, "Number of different nodes recovered" in the plot). As expected, the objective function value obtained from the output of the seeded `GMMF` algorithm is better than the ground truth alignment. Furthermore, by increasing ε beyond 0.1, we observed a deviation from the original recovered template, leading to the discovery of a new subgraph matching the template close to optimally. We comment that the decrease in the objective function value based on the alignment provided by the classified brain regions across the scans is a result of the seeds in the `SGM` algorithm, where the similarity scores between these 5 seeds pairs decrease as ε increases.

We close this example mentioning that by judiciously encoding neuronal information via the feature similarity matrix, the performance of the template recovery increases dramatically. The power of the similarity formalism is that it enables incorporation of any feature information for which similarities can be computed. For example, in the knowledge graph example of Section 3.3.4 the similarity encodes both numeric/quantitative and semantic/qualitative (i.e., ontological) features together. We note however that an adversarial S could potentially break our approach, as it could violate our working assumption of positively correlated edge structures and node similarities.

3.3.4 Template discovery in TKBs

For our second real data example, we consider the transactional knowledge base (TKB) of [96]. The graph is constructed from a variety of information sources including news articles, Reddit, Venmo, and bibliographic data. Moreover, nodes and edges are richly attributed. Node attributes include a unique node ID, node type (according to a custom ontology), free text value, entry ID (used to identify the node in the Wikidata Knowledge Base), date and latitude/longitude. Edge attributes include a unique edge ID, edge type (according to a custom ontology), and edge argument (providing additional edge information). See [96] for more information on the construction of this network and details on the custom ontological structure.

Along with the large background graph, [96] describes the creation of multiple signal templates (with varying levels of noise) to search for in the background. In addition to perfectly aligned templates (i.e., background subgraphs isomorphic to the template), templates are embedded with different and varying noise levels, necessitating noisy template recovery.

The full graph has 14,220,800 nodes and 157,823,262 edges. For each template, we do some simple preprocessing of the graphs that reduces their size (for instance, removing node types that do not appear in the template, removing dangling edges, etc). This preprocessing yields the pruned graphs that are fed into our matching algorithm, which have approximately 13×10^6 nodes and 32×10^6 edges. As in [92], we create a multiplex network from this TKB by dividing edge types (from the different sources) and different ontological edge types into multiple weighted graph layers (weighted based on a measure of ontological similarity), and using node features to define a node–to–node similarity matrix. Note that we use naive padding here, as the edge structure is naturally weighted and optimal centering in the weighted case is nuanced and the subject of present study. The

multiplex adaptation of the GMMF procedure can be found in [92], and amounts to adapting the Frank-Wolfe approach to the objective functions

$$\underset{P\in\Pi_n}{\operatorname{argmax}} \sum_i \operatorname{tr}\left(\left[A^{(i)}\oplus 0_{n-m,n-m}\right]PB^{(i)}P^T\right)+\lambda \operatorname{tr}\left(SP_{(1)}^T\right)$$

where $A^{(i)}$ (resp., $B^{(i)}$) represent the template (resp., background) structure in layer i of the multiplex graph and where nodes with a common label across layers are assumed aligned.

We show the effect of the solution diversification via the following experiment. To measure the fidelity of recovered templates when no ground truth is present, we use the graph edit distance (GED) metric outlined in [37]. We run 32 random restarts of the GMMF algorithm for each template recovery (we plot results for template 1A, 1B, 1C, 1D here; results for templates 2 and 3 can be found in Section 3.6.4), plotting the empirical CDF of the GED of the recovered templates; results are plotted in Figure 3.4. Different penalization values are represented with different colors in the plot. Note that in each template, version A has an isomorphic copy in the background while this is not guaranteed for the other templates (as they have noise introduced in the embeddings). From Figure 3.4, we see that the solution diversification is successful at yielding recovered templates including our optimal fits (the best recovered templates in the $\varepsilon = 0$ case) and templates that are close to optimal by GED. These templates further recover significant signal not recovered in the $\varepsilon = 0$ case; see Table 3.1. In the table, for each vertex in the template we show how many vertices in the background have similarity greater than 0 (i.e., are potential matches)—this is shown in the #>0 row. We see that the solution diversification is successful at recovering additional possible matches in the suboptimal recovered templates. We also see that too severe of a penalty (here the $\varepsilon = 0.01$ case) yields fewer unique nodes and worse GED fits. In summary, when there is no penalty, we consistently detect only a limited number of certain templates. When a penalty term is applied, we successfully

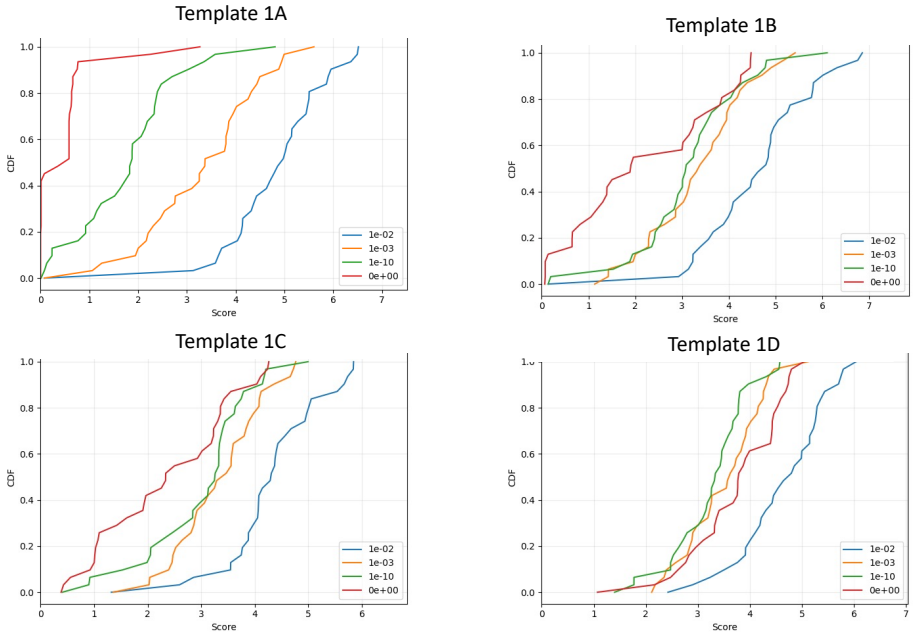

Figure 3.4: We run 32 random restarts of the `GMMF` algorithm for each template recovery, plotting the empirical CDF of the GED of the recovered templates. Different penalization values are represented with different colors in the plot.

recover some new templates. However, as the penalty becomes too large after a certain number of iterations, the noise in the non-signal nodes starts to dominate the signal in the uncovered templates, resulting in some sub-optimal recoveries. While choosing an optimal ε is of paramount importance, we do not have a fully principled recommendation for a best choice. We do recommend smaller penalty combined with more random restarts which achieved our best results. We close this example by noting that, because very few nodes in the background have a similarity greater than 0 to nodes in our templates, it is possible to filter the background by removing other nodes. However, the combinatorics after filtering remain complex; see [82] for more detailed discussions. That said, this filtering could be used to "soft seed" our graph matching (see [41]). We would prefer the soft seeding (where seeds are used to initialize the matching but not fixed throughout), as it is possible that for some nodes in the background, the similarity is incorrectly calculated as 0 but the matching should still be made, making a hard seed filtering step perhaps less desirable.

3.4 Conclusion and discussion

In this chapter, we have introduced a workflow for iteratively identifying multiple instances of noisy embedded templates within a large graph. Our approach extends the matched-filters-based method for noisy subgraph detection by considering both the edgewise structure and node feature similarities. By incorporating these factors, we have achieved a more diversified and scalable approach to effectively uncover embeddings of noisy copies of graph templates. The theoretical analysis of our algorithm demonstrates that, under the assumption of a strong correlation between the edgewise structural similarities and nodewise feature similarities, our approach can successfully identify multiple embedded templates within a large network. To validate the effectiveness of our proposed workflow, we conducted experiments using simulations based on the Multiple Correlated Erdős Rényi models, as well as real-world data sets such as human brain connectomes and the TKB dataset.

Furthermore, we present several intriguing questions that merit further exploration. In particular, the manuscript assumes an agreement between edge-structural similarity and node feature similarity. It would be valuable to investigate scenarios where such an

node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$\varepsilon = 0$	1	8	1	3	20	1	1	1	3	1	14	2	1	1	5	1	1	1	1	2	3	1	3
$\varepsilon = 0.01$	1	23	2	15	32	1	1	1	8	1	24	5	4	1	24	1	1	1	1	3	16	1	14
$\varepsilon = 10^{-3}$	4	21	20	29	32	5	5	4	27	1	26	5	27	3	31	1	4	4	3	3	27	5	26
$\varepsilon = 10^{-10}$	4	32	30	32	31	5	5	4	32	5	30	5	32	4	32	4	20	4	3	3	32	5	32
$\#>0$	4	32	32	32	32	5	5	4	32	5	32	5	32	4	32	4	32	4	3	3	32	5	32

node	24	25	26	27	28	29	30	31	32	33	34	35	36	37
$\varepsilon = 0$	19	1	1	1	3	4	1	3	4	1	2	4	1	1
$\varepsilon = 0.01$	32	1	2	2	5	18	1	3	25	1	2	5	1	1
$\varepsilon = 10^{-3}$	31	12	5	4	5	27	4	13	31	4	15	5	5	6
$\varepsilon = 10^{-10}$	32	29	5	4	5	32	4	28	32	4	31	5	5	21
$\#>0$	32	32	5	4	5	32	4	32	32	4	32	5	5	32

Table 3.1: In Template 1A, we list for each node how many vertices in the background have similarity greater than 0 (i.e., are potential matches)—this is shown in the $\#>0$ row). We then show for the different penalization levels, how many of these possible matches were recovered across the 32 random restarts.

agreement is absent, specifically identifying sharp parameter thresholds that lead to edge-structure dominated recovery, node-feature dominated recovery, and mixed-effect recovery. In all three cases, it is crucial to establish robust measures for evaluating the correctness of edge-wise matching between the template and the recovered template. Additionally, we highlight the issue of overlapping nodes between two embedded templates. The reliability of our algorithm relies on the ratio of overlapping parts between the templates being moderate. However, if the ratio is excessively high, penalizing already recovered templates may lead to sub-optimal results. To address this concern, it would be beneficial to develop methods that specifically target penalization on the non-overlapping regions while preserving the signal of the overlapping region, thereby enhancing the algorithm's performance.

3.5 Proof of Theorems

3.5.1 Proof of Theorem 2:

We restate Theorem 3.2 here before providing a proof.

Theorem 2: Let A and B be two graphs constructed as above. If there is a constant $\alpha \in [1/2, 1)$ such that

i. $m - k = \Theta(m)$; $m^{1-\alpha} = \omega(\log^4 n)$

ii. $\lambda = m^{\alpha}$;

iii. $(r_1 - r_3) \ll m^{\alpha-1}$; $r_1 > r_2 > r_3$ are bounded away from 0 and 1;

iv. $\mu_3 > (1-\varepsilon)\mu_1$ and $(1-\varepsilon)\mu_2 > \mu_4$; the differences $\mu_3 - (1-\varepsilon)\mu_1$, $(1-\varepsilon)\mu_2 - \mu_4$, and $\mu_3 - \mu_4$ are bounded away from 0;

v. p is bounded away from 0 and 1;

then if $\widetilde{\Pi}$ is the set of permutations perfectly aligning the weakly embedded template, we have that

$$\mathbb{P}(\text{argmax}_{P\in\Pi}\tilde{f}_{\varepsilon,1}(P,\lambda)\subset\widetilde{\Pi})\geq 1-e^{-\omega(\log n)}.$$

where we recall $\tilde{f}_{\varepsilon,1}(P,\lambda)$ is the objective function defined as $\tilde{f}_{\varepsilon,1}(P,\lambda)=\text{tr}\left(\tilde{A}P\tilde{B}P^T\right)+\lambda\,\text{tr}\left(S^{(2,\varepsilon)}P_{(1)}^T\right)$

Proof. The proof contains three steps. First, we calculate the expected contributions to the differences of the objective functions between (any one of) the permutations that recover the weakly embedded template, $\tilde{P}$, and another permutation P, due to the edge structures and the node features. See Propositions 3.1 and 3.2 below. Then, we use our assumptions to demonstrate that the actual difference must concentrate enough around its mean and be bounded away from 0 through the McDiarmid's inequality. See Theorem 3.3 below. Finally, by applying a union bound on all possible permutations, we establish that the probability of the difference being bounded away from 0 for all possible permutation matrices is at least $1-e^{-c_n}$ where c_n is proportional to the expected difference calculated in the first step. Combining this with the fact that the expected difference diverges to infinity as n increases, the result is established. See Theorem 3.4 below. □

Let P^* be (any one of) the permutation that maps A to the strongly embedded template in B, $\widetilde{P}$ be (any one of) the permutation that maps A to the weakly embedded template in B, and P be an arbitrary permutation in Π_n that does not map A to the weakly embedded template in B. Let $C_v=p(1-p)$. Finally let T_1 be the set of nodes that P correctly aligns within the strongly embedded template, and T_2 the set of nodes that P matches correctly to the weakly embedded template.

We first consider the contribution to the objective function of the edge disagreement induced by P. Define $D_E(P):=\text{tr}\left(\widetilde{A}\widetilde{P}\widetilde{B}\widetilde{P}^T\right)-\text{tr}\left(\widetilde{A}P\widetilde{B}P^T\right)$, which measures the objective

function difference due to edge structures between the permutation that recovers the weakly embedded template, $\tilde{P}$, and any other permutation matrix P.

Proposition 3.1. *Denote the counts of the correctly recovered template edges via P and $\tilde{P}$ with the following table.*

P \ $\widetilde{P}$	Recovers edges in $\binom{T_1}{2} \cap \binom{T_2}{2}$	Recovers edges in $\binom{T_2}{2} \setminus \binom{T_1}{2}$
Recovers edges in $\binom{T_1}{2} \setminus \binom{T_2}{2}$	0	h_1
Recovers edges in $\binom{T_1}{2} \cap \binom{T_2}{2}$	j_1	0
Recovers edges in $\binom{T_2}{2} \setminus \binom{T_1}{2}$	0	h_2
Misaligned template edges	j_2	h_3

Then $j_1 + j_2 = k(k-1)/2, 0 \leq h_1, h_2 \leq \frac{m(m-1)}{2} - \frac{k(k-1)}{2} = \frac{(m-k)(m+k-1)}{2}$ *and* $\sum_{i=1}^{3} h_i = \frac{(m-k)(m+k-1)}{2}$. *Further,*

$$\mathbb{E}(D_E(P)) = 8C_v \{ j_2 r_2 + h_1(r_3 - r_1) + h_3 r_3 \} \tag{3.3}$$

Proof. The equalities and inequalities involving j's and h's are trivial by counting. Also, we have

$$\begin{aligned}
\mathbb{E}\left\{ \operatorname{tr}\left(\widetilde{A} P \widetilde{B} P^T \right) \right\} =& \mathbb{E}\left\{ \sum_{i=1}^{m} \sum_{j=1}^{m} (2A_{ij} - 1)(2B_{\sigma_P(i)\sigma_P(j)} - 1) \right\} \\
=& \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{E}\left[4A_{ij} B_{\sigma_P(i)\sigma_P(j)} - 2A_{ij} - 2B_{\sigma_P(i)\sigma_P(j)} + 1 \right]
\end{aligned}$$

Thus,

$$\mathbb{E}\left\{ \operatorname{tr}\left(\widetilde{A} P \widetilde{B} P^T \right) \right\} = -8 \binom{m}{2} C_v + 4 \sum_{i=1}^{m} \sum_{j=1}^{m} r_{g_{ij}} C_v + m^2$$

where, recall that we use $\binom{S}{2}$ to denote the collection of all 2-element subsets of the set S,

$$g_{ij} = \begin{cases} 1, & \text{if } \{i,j\} \in \binom{T_1}{2} \setminus \binom{T_2}{2}; \\ 2, & \text{if } \{i,j\} \in \binom{T_1}{2} \cap \binom{T_2}{2}; \\ 3, & \text{if } \{i,j\} \in \binom{T_2}{2} \setminus \binom{T_1}{2}; \\ 0, & \text{otherwise.} \end{cases}$$

Note that the constant terms $-8\binom{m}{2}C_v$ and m^2 cancel when taking differences. Also, j_1 and h_2 terms vanish since they correspond to cases where $r_{g_{ij}}$ values agree for P and $\tilde{P}$, so we know

$$\mathbb{E}(D_E(P)) = \mathbb{E}\left\{\operatorname{tr}\left(\widetilde{A}\widetilde{P}\widetilde{B}\widetilde{P}^T\right) - \operatorname{tr}\left(\widetilde{A}P\widetilde{B}P^T\right)\right\} = 8C_v\{j_2r_2 + h_1(r_3 - r_1) + h_3r_3\}$$

□

Now, we consider S of the form

$$S = \begin{array}{c} \\ m-k \\ k \end{array} \begin{array}{c} \begin{array}{cccc} m-k & k & m-k & n-2m+k \end{array} \\ \begin{pmatrix} S^{11} & S^{12} & S^{13} & S^{14} \\ S^{21} & S^{22} & S^{23} & S^{24} \end{pmatrix} \end{array}$$

where all entries of S are independent—bounded in $[0,1]$—random variables, and where

the diagonal elements of S^{11} have mean μ_1

the diagonal elements of S^{22} have mean μ_2

the diagonal elements of S^{13} have mean μ_3

and all other entries have mean μ_4.

If the strongly embedded template is penalized, then S is weighted via (where "$\circ$" is the matrix Hadamard product)

$$S^{(2,\varepsilon)} = \begin{array}{c} \\ m-k \\ k \end{array} \begin{array}{c} \begin{array}{cccc} m-k & k & m-k & n-2m+k \end{array} \\ \begin{pmatrix} (\mathbf{1}_{m-k,m-k} - \varepsilon I_{m-k}) \circ S^{11} & S^{12} & S^{13} & S^{14} \\ S^{21} & (\mathbf{1}_{k,k} - \varepsilon I_k) \circ S^{22} & S^{23} & S^{24} \end{pmatrix} \end{array}$$

We next consider the contribution to the objective function of the features/similarity induced by P. Like before, we define $D_F(P) := \operatorname{tr}\left(S^{(2,\varepsilon)}\tilde{P}^T\right) - \operatorname{tr}\left(S^{(2,\varepsilon)}P^T\right)$ which measures the objective function difference due to node similarities between the permutation that recovers the weakly embedded template, $\tilde{P}$, and any other permutation matrix P.

Proposition 3.2. *Denote the counts of the recovered template nodes via P and $\tilde{P}$ with the following table*

P \ $\tilde{P}$	Recovers $T_1 \cap T_2$	Recovers $T_2 \setminus T_1$
Recovers $T_1 \setminus T_2$	0	b_1
Recovers $T_1 \cap T_2$	a_1	0
Recovers $T_2 \setminus T_1$	0	b_2
Misaligned template node	a_2	b_3

Then $0 \le a_1 \le k, a_1 + a_2 = k, 0 \le b_1, b_2 < m-k$ and $\sum_{i=1}^{3} b_i = m-k$. Also,

$$\binom{a_1}{2} = j_1, \quad \binom{a_2}{2} + a_1 a_2 = j_2, \quad \binom{b_2}{2} + b_2 a_1 = h_2,$$
$$\binom{b_1}{2} + b_1 a_1 = h_1, \quad \binom{b_3}{2} + b_1 b_2 + b_1 b_3 + b_2 b_3 + a_2(m-k) + b_3 a_1 = h_3.$$

where j's and h's are the counts from the table of Proposition 3.1 above. Moreover,

$$\mathbb{E}(D_F(P)) = a_2[(1-\varepsilon)\mu_2 - \mu_4] + b_1[\mu_3 - (1-\varepsilon)\mu_1] + b_3[\mu_3 - \mu_4]. \tag{3.4}$$

Proof. The equalities and inequalities involving a's and b's, as well as the relationship between a's b's and j's, h's are trivial by counting.

Now, we have

$$\begin{aligned}
\mathbb{E}\left[\operatorname{tr}\left(S^{(2,\varepsilon)}P^T\right)\right] =& \mathbb{E}\left\{\sum_{i=1}^{m}(1-\varepsilon)^{\mathbb{1}\{\sigma_P(i)=\sigma_{P^*}(i)\}}S_{i,\sigma_P(i)}\right\} \\
=& \sum_{i=1}^{m}(1-\varepsilon)^{\mathbb{1}\{\sigma_P(i)=\sigma_{P^*}(i)\}}\mathbb{E}(S_{i,\sigma_P(i)}) \\
=& \sum_{i=1}^{m}(1-\varepsilon)^{\mathbb{1}\{\sigma_P(i)=\sigma_{P^*}(i)\}}\mu_{w_i}
\end{aligned}$$

where

$$w_i = \begin{cases} 1, & \text{if } i \in T_1 \setminus T_2; \\ 2, & \text{if } i \in T_1 \cap T_2; \\ 3, & \text{if } i \in T_2 \setminus T_1; \\ 4, & \text{otherwise.} \end{cases}$$

a_1 and b_2 terms vanish when taking differences since they correspond to cases where $\mathbb{1}\{\sigma_P(i) = \sigma_{P^*}(i)\}$ and μ_{w_i} values agree for P and $\tilde{P}$, so we know

$$\begin{aligned}
\mathbb{E}(D_F(P)) &= \mathbb{E}\left\{\operatorname{tr}\left(S^{(2,\varepsilon)}\tilde{P}^T\right) - \operatorname{tr}\left(S^{(2,\varepsilon)}P^T\right)\right\} \\
&= a_2[(1-\varepsilon)\mu_2 - \mu_4] + b_1[\mu_3 - (1-\varepsilon)\mu_1] + b_3[\mu_3 - \mu_4].
\end{aligned}$$

□

Let $g(P,\tilde{P}) = \tilde{f}_{\varepsilon,1}(\tilde{P},\lambda) - \tilde{f}_{\varepsilon,1}(P,\lambda)$ so that

$$
\begin{aligned}
\mathbb{E}g(P,\widetilde{P}) &= \mathbb{E}(D_E(P) + \lambda D_F(P)) \\
&= 8C_v\Big\{ \left(\binom{a_2}{2} + a_1 a_2 \right) r_2 + \left(\binom{b_1}{2} + b_1 a_1 \right)(r_3 - r_1) \\
&\quad + \left(\binom{b_3}{2} + b_1 b_2 + b_1 b_3 + b_2 b_3 + a_2(m-k) + b_3 a_1 \right) r_3 \Big\} \\
&\quad + \lambda\left(a_2[(1-\varepsilon)\mu_2 - \mu_4] + b_1[\mu_3 - (1-\varepsilon)\mu_1] + b_3[\mu_3 - \mu_4]\right) \\
&\geq 8C_v\Big\{ \underbrace{\left(\binom{b_1 + a_2 + b_3}{2} + (b_1 + a_2 + b_3)(b_2 + a_1) \right)}_{:= n_E} r_3 - \left(\binom{b_1}{2} + b_1 a_1 \right) r_1 \Big\} \\
&\quad + \lambda \underbrace{(a_2 + b_1 + b_3)}_{:= n_F} * \underbrace{\min([(1-\varepsilon)\mu_2 - \mu_4], [\mu_3 - (1-\varepsilon)\mu_1], [\mu_3 - \mu_4])}_{:= c_\varepsilon} \\
&= 8C_v \left(n_E r_3 - \left(\binom{b_1}{2} + b_1 a_1 \right) r_1 \right) + \lambda n_F c_\varepsilon.
\end{aligned}
$$

Considering the case where $b_1 = m-k$ and $a_1 = k$, we see that for the above expectation to be diverging to infinity, it suffices that

$$\lambda c_\varepsilon \gg (r_1 - r_3)(m+k)C_v; \tag{3.5}$$

Equation 3.5 holds true under assumptions ii and iii of Theorem 3.2.

Next, we state the McDiarmid's inequality and use it to show for any $P \in \Pi_n$, $g(P,\tilde{P})$ concentrates around its expectation.

Theorem 3.3 (McDiarmid's inequality [80]). *Let $X_1, \ldots, X_n$ be independent random variables, where $X_i \in \mathscr{X}_i$. Let $f : \mathscr{X}_1 \times \cdots \times \mathscr{X}_n \to \mathbb{R}$ be any function that satisfies the property: there exists $c_1, \ldots c_n$ such that for every $i \in [n]$ and every $(x_1, \ldots, x_n) \in \mathscr{X}_1 \times \ldots \times \mathscr{X}_n$, we*

have

$$\sup_{x_i' \in \mathscr{X}_i} \left| f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) - f\left(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_n\right) \right| \leq c_i$$

Then for any $t > 0$, we have

$$\mathbb{P}\left(f(X_1, \ldots, X_n) - \mathbb{E}\left[f(X_1, \ldots, X_n)\right] \geq t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right)$$

From the forms of $D_E(P)$ and $D_F(P)$, we have that $g(P, \widetilde{P})$ is a function of at most (as P and $\widetilde{P}$ agree on $b_2 + a_1$ template vertices and disagree on the rest)

$$\underbrace{2m - 2b_2 - 2a_1}_{\text{from } D_F(P)} + \underbrace{\binom{m - b_2 - a_1}{2} + (m - b_2 - a_1)(b_2 + a_1)}_{\text{from } D_E(P)} \tag{3.6}$$

random variables, and changing any of the variables from $D_F(P)$ can change $g(P, \widetilde{P})$ by at most 4λ, and changing any of the variables from $D_E(P)$ can change $g(P, \widetilde{P})$ by at most a bounded constant (bounded above by 8 for example). Lastly, note that $m - b_2 - a_1 = b_1 + b_3 + a_2 = n_F$ and

$$\binom{m - b_2 - a_1}{2} + (m - b_2 - a_1)(b_2 + a_1) = \binom{b_1 + a_2 + b_3}{2} + (b_1 + a_2 + b_3)(b_2 + a_1) = n_E.$$

So

$$n_E = n_F \left(\frac{n_F - 1}{2} + b_2 + a_1\right)$$

Assumption iv. of Theorem 3.2 gives that $(1 - \varepsilon)\mu_2 > \mu_4$ and $(1 - \varepsilon)\mu_1 < \mu_3$, and that the differences

$$(1 - \varepsilon)\mu_2 - \mu_4, \quad \mu_3 - \mu_4, \text{ and } \mu_3 - (1 - \varepsilon)\mu_1$$

are bounded away from 0. Further, Eq. (3.5) implies that

$$\begin{aligned}\lambda n_F c_\varepsilon &\gg n_F(r_1-r_3)(m+k)C_v \\ &\gtrsim \binom{b_1}{2}(r_1-r_3)C_v \\ &\gtrsim C_v\left(\left(\binom{b_1}{2}+b_1a_1\right)r_1-r_3n_E\right)\end{aligned}$$

and $\mathbb{E}g(P,\widetilde{P}) = \Omega(\lambda n_F c_\varepsilon)$.

Apply theorem 3.3 with $g(P,\tilde{P})$ as a function of n_E+2n_F random variables from Eq. 3.6 and $c_i = 8$ for random variables from $D_E(P)$ and $c_i = 4\lambda$ for random variables from $D_F(P)$, we see that for n—and hence $m = m_n$—sufficiently large, (where $\xi > 0$ is a constant that can change line–to–line)

$$\begin{aligned}&\mathbb{P}(g(P,\widetilde{P}) \le 0) \le \mathbb{P}(|g(P,\widetilde{P}) - \mathbb{E}[g(P,\widetilde{P})]| \ge \mathbb{E}[g(P,\widetilde{P})]) \\ &\le 2\exp\left\{-\xi\frac{C_v^2\left(n_Er_3-\left(\binom{b_1}{2}+b_1a_1\right)r_1\right)^2+\lambda^2n_F^2c_\varepsilon^2}{n_E+\lambda^2n_F}\right\} \\ &\le 2\exp\left\{-\xi\frac{C_v^2\left(\left(b_3^2+a_2^2+(b_1+b_3+a_2)(b_2+a_1)\right)r_3-b_1^2(r_1-r_3)\right)^2+m^{2\alpha}n_F^2c_\varepsilon^2}{m^{2\alpha}n_F}\right\}\end{aligned} \tag{3.7}$$

Next, we have the following lemma

Lemma 3.1. *Let $H(\cdot)$ be the binary entropy function as defined in [29]. Partition Π_n based on $\Pi_n \ni P \sim Q \in \Pi_n$ if the first m rows of P and Q are exactly equal in to equivalence classes. Then within each partition, our objective function $\tilde{f}_{\varepsilon,1}(\cdot,\lambda)$ takes the same value; moreover, the order of each equivalence class is bounded above by $2^{kH\left(\frac{a_1}{k}\right)+(m-k)H\left(\frac{b_1}{m-k}\right)+(m-k)H\left(\frac{b_2}{m-k}\right)}n^{a_2+b_3}$.*

Proof. Fix an equivalence class C. For any $P,Q \in C$, the contribution due to edge terms are

equal since the bottom $n-m$ rows of both P and Q contribute 0 to the value of the objective function due to padding. Also, the contribution due to feature terms is independent of the bottom $n-m$ rows of the permutation matrix. Thus $\tilde{f}_{\varepsilon,1}(P,\lambda) = \tilde{f}_{\varepsilon,1}(Q,\lambda)$ which proves the objective function takes the same value in C.

Now, by (7.14) from [29]

$$|C| = \binom{k}{a_1}\binom{m-k}{b_1}\binom{m-k}{b_2} n^{a_2+b_3} \leq 2^{kH\left(\frac{a_1}{k}\right)+(m-k)H\left(\frac{b_1}{m-k}\right)+(m-k)H\left(\frac{b_2}{m-k}\right)} n^{a_2+b_3}. \quad (3.8)$$

□

Finally, we apply the union bound on all permutation matrices and get

Theorem 3.4. *With the assumptions of Theorem 3.2, by Lemma 3.1 and Equation 3.7, we have*

$$\mathbb{P}(\exists P \neq \widetilde{P} \; s.t. \; g(P,\widetilde{P}) \leq 0) \leq 2e^{-\omega(\log n)} \quad (3.9)$$

Proof. Apply a union bound over all P with the same counts (a's and b's), modulo the equivalence between permutations with the same first m rows; further apply a union bound over all possible counts of the a's and b's, we have (where $\xi > 0$ is a constant that can change line–to–line)

$$\begin{aligned}
&\mathbb{P}(\exists P \neq \widetilde{P} \text{ s.t. } g(P,\widetilde{P}) \leq 0) \leq \\
&\sum_{a_1=0}^{k}\sum_{b_1=0}^{m-k}\sum_{b_2=0}^{m-k-b_1} 2\exp\Bigg\{-\xi\frac{C_v^2\left(\left(b_3^2+a_2^2+(b_1+b_3+a_2)(b_2+a_1)\right)r_3 - b_1^2(r_1-r_3)\right)^2 + m^{2\alpha}n_F^2c_\varepsilon^2}{m^{2\alpha}n_F} \\
&\quad + kH\left(\frac{a_1}{k}\right)\log(2) + (m-k)H\left(\frac{b_1}{m-k}\right)\log(2) + (m-k)H\left(\frac{b_2}{m-k}\right)\log(2) \\
&\quad + (a_2+b_3)\log(n)\Bigg\} \qquad (3.10)
\end{aligned}$$

To tackle the above Eq. (3.10), we consider several cases:

- If $b_3^2 = \Omega(m^{3/2+\alpha/2})$ or $a_2^2 = \Omega(m^{3/2+\alpha/2})$, then the exponential in Eq. (3.10) can be bounded above by (where $\xi > 0$ is a constant that can change line–to–line)

$$2\exp\left\{-\xi\frac{C_v^2 m^{3+\alpha}}{m^{2\alpha}n_F} + m\log n\right\} \leq 2\exp\left\{-\xi C_v^2 m(\underbrace{m^{2-\alpha} - \log n}_{:=\Theta(m^{2-\alpha})})\right\} = 2e^{-\omega(\log n)} \tag{3.11}$$

- We next consider the case where both $b_3^2 = o(m^{3/2+\alpha/2})$ and $a_2^2 = o(m^{3/2+\alpha/2})$, in this case either $b_1 = \Theta(m)$, $b_2 = \Theta(m)$, or both are $\Theta(m)$.

  - $b_1 = \Theta(m)$ and $b_2 = \Theta(m)$: Then the $b_1 * b_2$ term in the exponent in Eq. (3.10) yields the upper bound (where $\xi > 0$ is a constant that can change line–to–line)

$$2\exp\left\{-\xi\frac{C_v^2 m^4}{m^{2\alpha}n_F} + m\log n\right\} \leq 2\exp\left\{-\xi C_v^2 m(\underbrace{m^{3-2\alpha} - \log n}_{:=\Theta(m^{3-2\alpha})})\right\} = 2e^{-\omega(\log n)} \tag{3.12}$$

  - $b_2 = o(m)$: In this case, $b_1(1+o(1)) = m-k$. Note that if $x(1+o(1)) = y$, then $x/y(1+o(1)) = 1$. If $x \leq y$, then $1 - x/y = o(1)$. As $\lim_{z\to 0} -z\log_2 z = 0$, we then have

$$\begin{aligned} yH(x/y) &= y\left[\frac{x}{y}\underbrace{\left[-\log_2\left(\frac{x}{y}\right)\right]}_{=o(1)} + \underbrace{\left[-\left(1-\frac{x}{y}\right)\log_2\left(1-\frac{x}{y}\right)\right]}_{=o(1)}\right] \\ &= o(y) \end{aligned}$$

Similarly, if $y = \omega(1)$ and $x = o(y)$, then

$$yH(x/y) = y\left[\underbrace{-\frac{x}{y}\log_2\left(\frac{x}{y}\right)}_{=o(1)} + \left(1-\frac{x}{y}\right)\underbrace{\left[-\log_2\left(1-\frac{x}{y}\right)\right]}_{=o(1)}\right]$$
$$= o(y)$$

The $n_F^2 * m^{2\alpha}$ term in the exponent in Eq. (3.10) yields the upper bound (where $\xi > 0$ is a constant that can change line–to–line; note by assumption c_ε is bounded away from 0)

$$2\exp\Big\{-\xi\frac{n_F^2 * m^{2\alpha}c_\varepsilon^2}{m^{2\alpha}n_F} + kH\left(\frac{a_1}{k}\right)\log(2) + (m-k)H\left(\frac{b_1}{m-k}\right)\log(2)$$
$$+ (m-k)H\left(\frac{b_2}{m-k}\right)\log(2) + (a_2+b_3)\log(n)\Big\}$$
$$\leq 2\exp\Big\{-\xi mc_\varepsilon^2 + o(m) + o(m^{3/4+\alpha/4}\log(n))\Big\} = 2e^{-\omega(\log n)} \qquad (3.13)$$

– $b_1 = o(m)$: In this case, $b_2(1+o(1)) = (m-k)$. We make use here of the alternate bound where if $y = \omega(1)$ and $x = o(y)$, then (as $\lim_{z\to 0} -\log_2(1-z)/z = 1/\log(2)$)

$$yH(x/y) = y\left[-\frac{x}{y}\log_2\left(\frac{x}{y}\right) - \left(1-\frac{x}{y}\right)\log_2\left(1-\frac{x}{y}\right)\right]$$
$$= O(x\log_2(y)) + O\left(y\frac{x}{y}\left(1-\frac{x}{y}\right)\right) = O(x\log(y)).$$

We also note that here

$$
\begin{aligned}
&(m-k)H\left(\frac{b_2}{m-k}\right)\\
&= (m-k)H\left(1-\frac{b_1+b_3}{m-k}\right)\\
&= (m-k)\left[-\left(1-\frac{b_1+b_3}{m-k}\right)\log_2\left(1-\frac{b_1+b_3}{m-k}\right)-\left(\frac{b_1+b_3}{m-k}\right)\log_2\left(\frac{b_1+b_3}{m-k}\right)\right]\\
&= O\left((m-k)\frac{b_1+b_3}{m-k}\left(1-\frac{b_1+b_3}{m-k}\right)\right)+O((b_1+b_3)\log_2(m-k))\\
&= O((b_1+b_3)\log(m-k))
\end{aligned}
$$

The $n_F^2 * b_2^2$ term in the exponent in Eq. (3.10) yields the upper bound (where $\xi > 0$ is a constant that can change line–to–line, and we use $kH\left(\frac{a_1}{k}\right) = kH\left(1-\frac{a_2}{k}\right)$ as above)

$$
\begin{aligned}
&2\exp\Big\{-\xi n_F m^{2-2\alpha}+kH\left(\frac{a_1}{k}\right)\log(2)+(m-k)H\left(\frac{b_1}{m-k}\right)\log(2)\\
&\qquad +(m-k)H\left(\frac{b_2}{m-k}\right)\log(2)+(a_2+b_3)\log(n)\Big\}\\
&\leq 2\exp\Big\{-\xi n_F m^{2-2\alpha}+O((b_1+b_3+a_2)\log(m))+n_F\log(n)\Big\} = 2e^{-\omega(\log n)}
\end{aligned}
\tag{3.14}
$$

Therefore, by Equations 3.11, 3.12, 3.13 and 3.14, we have that individual summands of the right-hand-side of Eq. (3.10) is bound above by $2e^{-\omega(\log n)}$, and thus

$$
\begin{aligned}
\mathbb{P}(\exists P \neq \widetilde{P} \text{ s.t. } g(P,\widetilde{P}) \leq 0) &\leq 2e^{-\omega(\log n)}\left(\sum_{a_1=0}^{k}\sum_{b_1=0}^{m-k}\sum_{b_2=0}^{m-k-b_1}1\right)\\
&\leq 2e^{-\omega(\log n)+\ln k+2\ln m} = 2e^{-\omega(\log n)}
\end{aligned}
$$

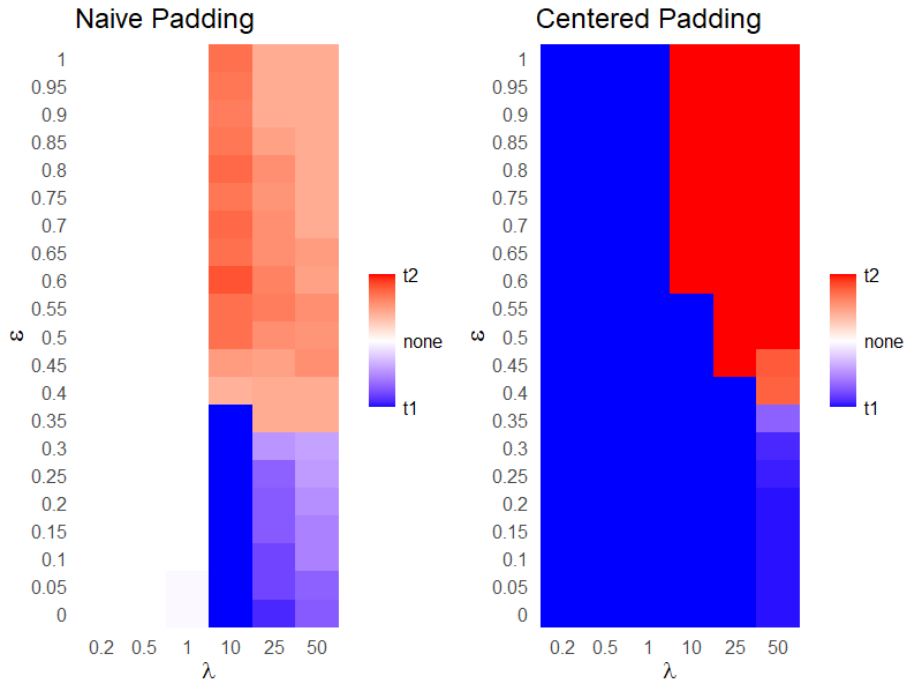

Figure 3.5: We fix $k = 15$ and use the seeded `GMMF` algorithm to match A with B using 5 seeds randomly selected from the overlapping nodes of $B^{(1)}$ and $B^{(2)}$ as described in Section 3.3.1. We plot the recovering results over ε (here ε is used to penalize the stronger of the two embedded templates) and λ, averaged by 20 Monte-Carlo simulations, where blue means the recovered template is closer to $B^{(1)}$ (the stronger embedded template), red means the recovered template is closer to $B^{(2)}$ (the weaker embedded template), and white means there is a tie in the 20 simulations or the recovered template is not close to either $B^{(1)}$ or $B^{(2)}$.

as desired. □

3.6 More Experiments

3.6.1 Additional two overlapping templates experiments

We plot the cases of $k = 15$ and $k = 40$ using the seeded `GMMF` algorithm with 5 seeds randomly selected from the overlapping nodes of $B^{(1)}$ and $B^{(2)}$, for the same parameters as described in 3.3.1.

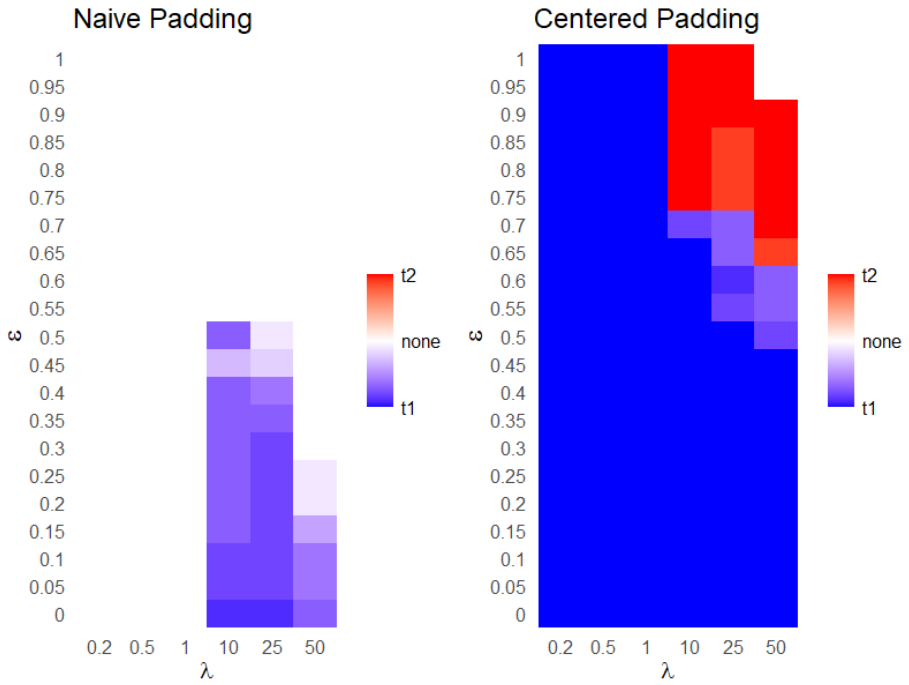

Figure 3.6: We fix $k = 40$ and use the seeded `GMMF` algorithm to match A with B using 5 seeds randomly selected from the overlapping nodes of $B^{(1)}$ and $B^{(2)}$ as described in Section 3.3.1. We plot the recovering results over ε (here ε is used to penalize the stronger of the two embedded templates) and λ, averaged by 20 Monte-Carlo simulations, where blue means the recovered template is closer to $B^{(1)}$ (the stronger embedded template), red means the recovered template is closer to $B^{(2)}$ (the weaker embedded template), and white means there is a tie in the 20 simulations or the recovered template is not close to either $B^{(1)}$ or $B^{(2)}$. Note that the naive padding never recovered anything closer to $B^{(2)}$.

3.6.2 Additional three overlapping templates experiments

We first plot the simulated results where the parameters correspond to Figure 3.2 of 3.3.2 but with the naive padding.

Next, with the same correlation parameters $\{r_j\}_{j=1}^{4}$ and feature mean parameters $\{\mu_j\}_{j=1}^{5}$, we plot the simulated results for $k = 10, \lambda = 10$ and $k = 40, \lambda = 25$ using the centered padding.

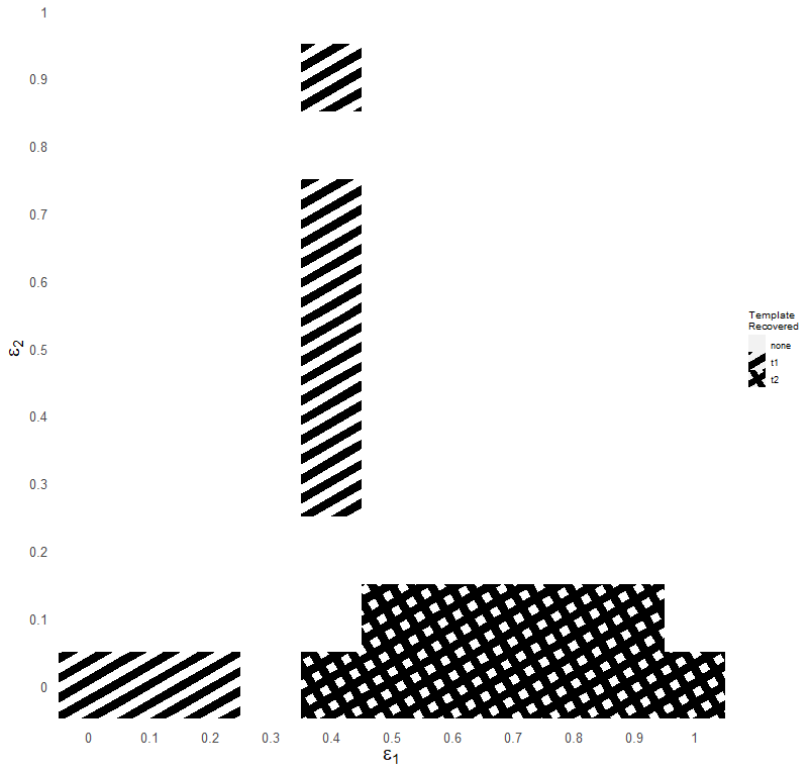

Figure 3.7: We fix $k = 10, \lambda = 25$ and use the seeded `GMMF` algorithm with the naive padding to match A with B using 5 seeds randomly selected from the overlapping nodes of $B^{(1)}, B^{(2)}$ and $B^{(3)}$, where $B^{(1)}, B^{(2)}$ and $B^{(3)}$ are induced subgraph of B such that graphs A and B follows multiple correlated ER model as described in Section 3.3.2. We plot the recovering results over ε_1 (penalty applied to the diagonal elements of $S^{(11)}, S^{(22)}$) and ε_2 (penalty applied to the diagonal elements of $S^{(13)}, S^{(22)}$), averaged by 20 Monte-Carlo simulations. In the figure, the different patterns represent which template was recovered (in majority): t1 for template 1, t2 for template 2, and we fail to recover template 3, with white squares corresponding to the case when none of the three templates was recovered.

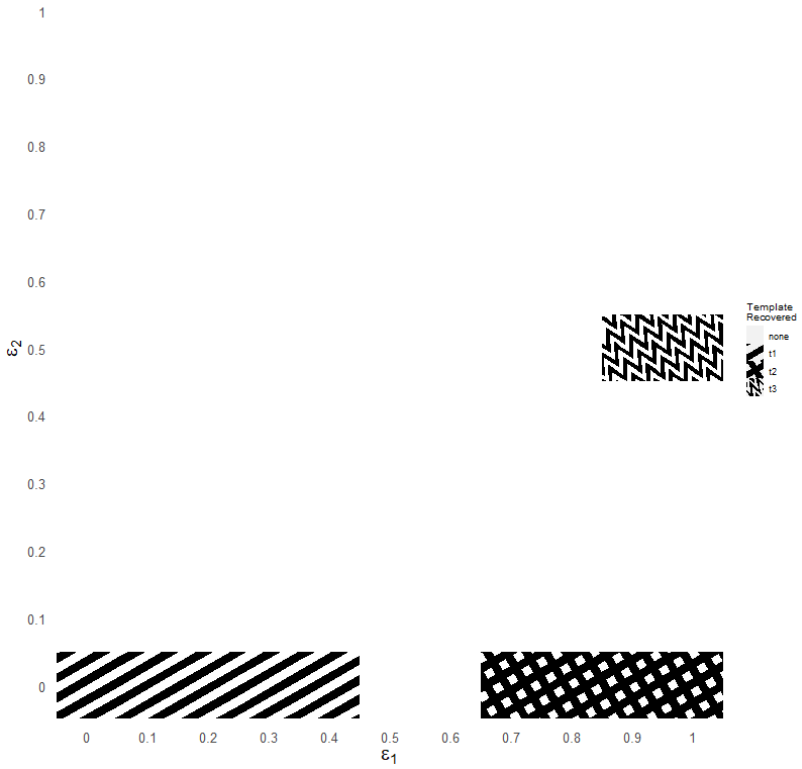

Figure 3.8: We fix $k = 10, \lambda = 10$ and use the seeded `GMMF` algorithm with the centered padding to match A with B using 5 seeds randomly selected from the overlapping nodes of $B^{(1)}, B^{(2)}$ and $B^{(3)}$, where $B^{(1)}, B^{(2)}$ and $B^{(3)}$ are induced subgraph of B such that graphs A and B follows multiple correlated ER model as described in Section 3.3.2. We plot the recovering results over ε_1 (penalty applied to the diagonal elements of $S^{(11)}, S^{(22)}$) and ε_2 (penalty applied to the diagonal elements of $S^{(13)}, S^{(22)}$), averaged by 20 Monte-Carlo simulations. In the figure, the different patterns represent which template was recovered (in majority): t1 for template 1, t2 for template 2, and t3 for template 3, with white squares corresponding to the case when none of the three templates was recovered.

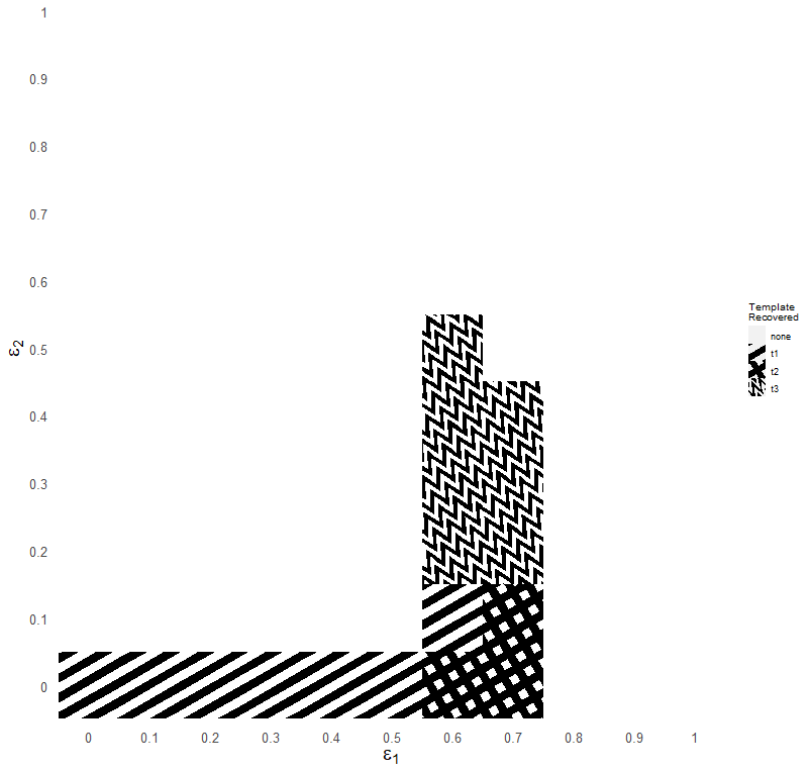

Figure 3.9: We fix $k = 40, \lambda = 25$ and use the seeded `GMMF` algorithm with the centered padding to match A with B using 5 seeds randomly selected from the overlapping nodes of $B^{(1)}, B^{(2)}$ and $B^{(3)}$, where $B^{(1)}, B^{(2)}$ and $B^{(3)}$ are induced subgraph of B such that graphs A and B follows multiple correlated ER model as described in Section 3.3.2. We plot the recovering results over ε_1 (penalty applied to the diagonal elements of $S^{(11)}, S^{(22)}$) and ε_2 (penalty applied to the diagonal elements of $S^{(13)}, S^{(22)}$), averaged by 20 Monte-Carlo simulations. In the figure, the different patterns represent which template was recovered (in majority): t1 for template 1, t2 for template 2, and t3 for template 3, with white squares corresponding to the case when none of the three templates was recovered.

3.6.3 Additional Brain MRI plots

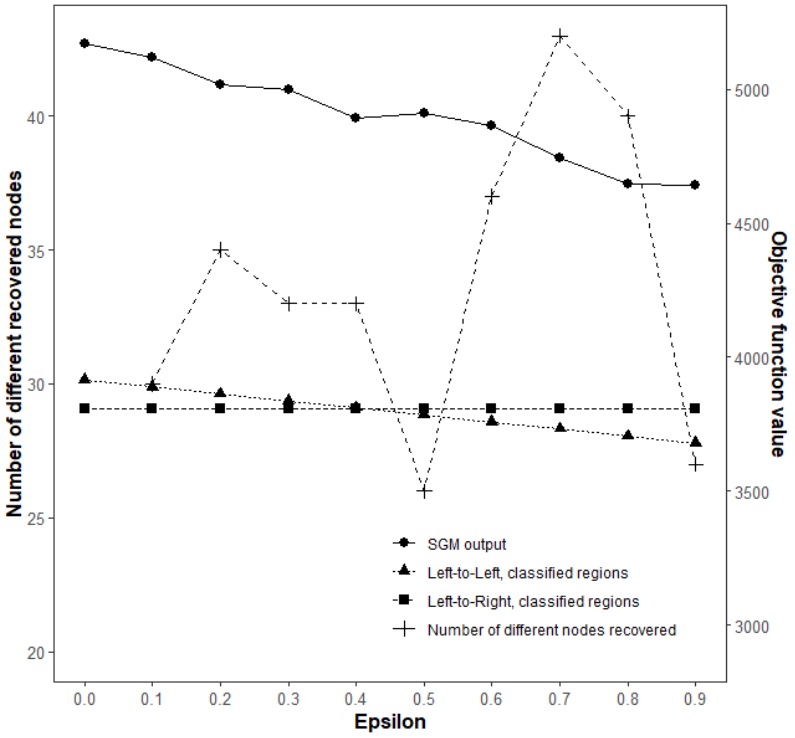

Figure 3.10: We run our proposed algorithm using the seeded `GMMF` algorithm with 500 restarts and 5 seeds selected from the node pairs $\{(s_j, w_j)\}_{j=1}^{6}$ as described in 3.3.3, taking the result with highest objective function value (Eq. 3.2, $\lambda = 1$) as the output. For each ε, we compute the GM objective function value (left axis) of the resulting matrix with the template; we also computed the objective function value with respect to the alignment given by the template to the same classified brain region in the left hemisphere in B (Left–to–Left in the plot), as well as the objective function value given by the template to the symmetric region from the right hemisphere in B (Left–to–right in the plot). Also for $\varepsilon > 0$, we calculated the number of novel nodes recovered in each matching compared to the subgraph detected with $\varepsilon = 0$ (right axis).

3.6.4 Additional TKB templates

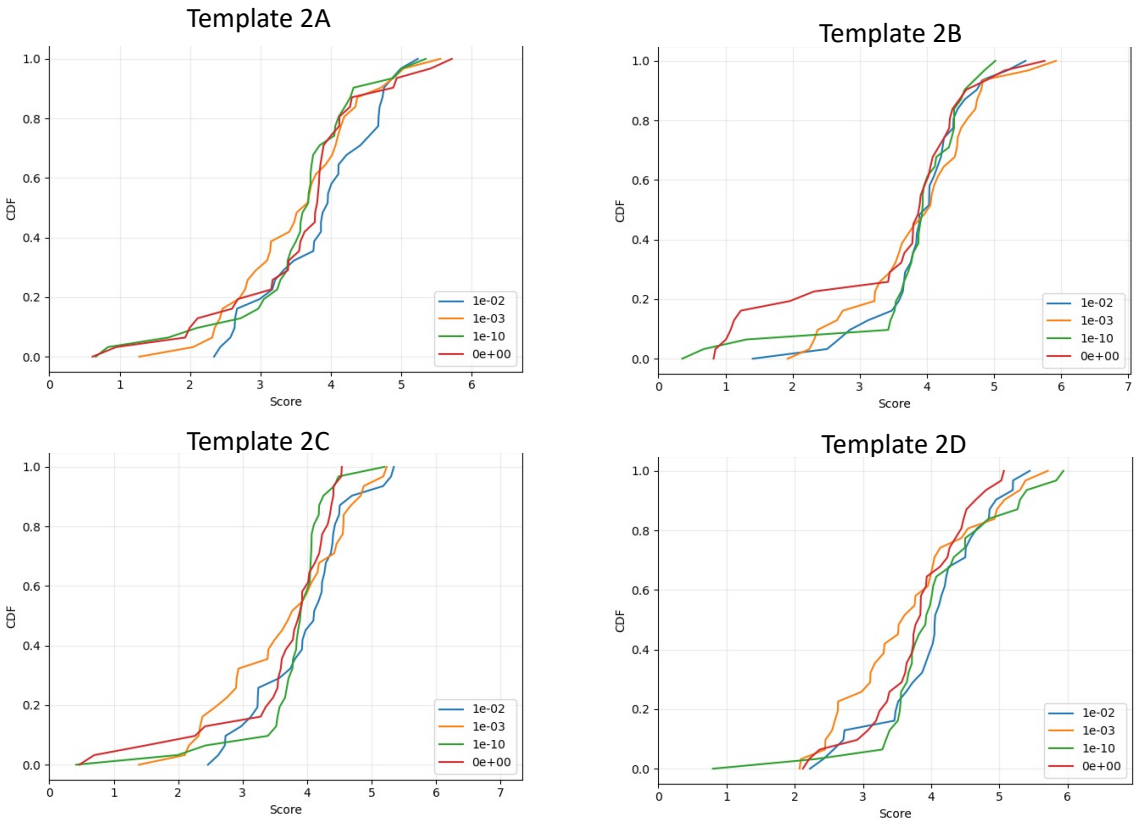

Figure 3.11: We run 32 random restarts of the `GMMF` algorithm for each template recovery, plotting the empirical CDF of the GED of the recovered templates. Different penalization values are represented with different colors in the plot.

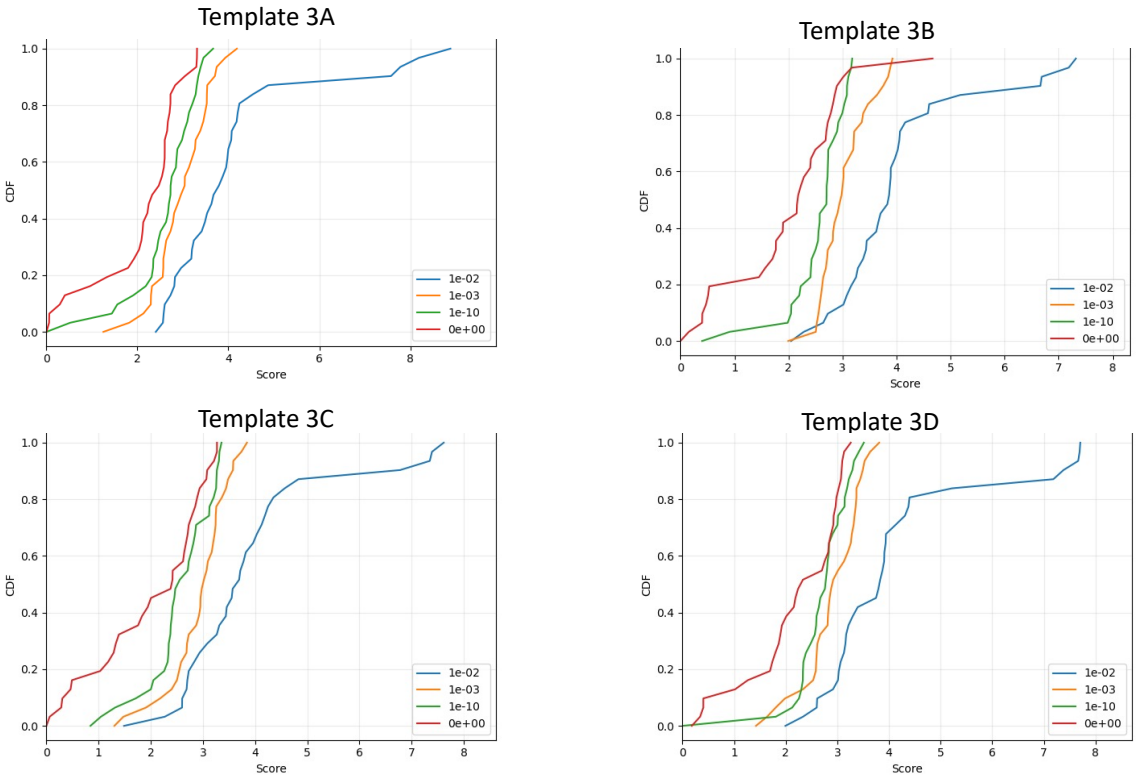

Figure 3.12: We run 32 random restarts of the `GMMF` algorithm for each template recovery, plotting the empirical CDF of the GED of the recovered templates. Different penalization values are represented with different colors in the plot.

Chapter 4: Matching and mixing: Matchability of graphs under Markovian error

4.1 Introduction

Recall that statistical network inference often begins with observed graphs that are assumed to be noisy copies of some underlying background latent position graphs [6, 67, 95, 128]. From these observations, numerous tasks such as classification, hypothesis testing, and more can also be pursued, see [24, 35, 49, 100, 114, 117, 124] and [32]. A crucial assumption many of these methods rely on is the idea of vertices being *a priori aligned* across networks before inference is pursued. In situations where this is not the case, inference can degrade [104] when vertex labels are shuffled or noisily observed. In these cases, graph matching methods can then be applied to attempt to recover the true labeling across graphs and enable the application of subsequent inference methods. Nearly all of the aforementioned theoretical graph matching developments in previous chapters rely on the assumption of independent noise on the edges. However, in real applications such as protein-protein interaction networks [109], disease spreading networks [86], the propagation of computational errors in neural networks [68], or temporal networks [9], it is more realistic to consider time- or node-dependent noise. To address this issue, our focus in the subsequent sections will be to propose some preliminary results on the matchability of noisy graphs with their original counterparts in the presence of spatially and temporally dependent edge-noise processes. Before introducing our model with dependent noise on

edges, we will review some relevant background and definitions.

4.1.1 Lamplighter Walks

A random walk on a graph is defined such that, at each step, one can traverse from the current node to a neighboring node [72]; herein, we will assume our random walks are Markov chains with appropriately defined state space and transition matrix. A key property of Markov chains is that ergodic (i.e., positive recurrent, aperiodic and irreducible) chains converge in the limit to their stationary distribution. A branch of modern Markov chain theory is devoted to analyzing how fast this convergence occurs, motivating the definition of mixing times of Markov chains [4, 66].

Definition 4.1. *Let (X_n) define a discrete time ergodic Marckov chain on state space S with one-step transition matrix P and stationary distribution π. We define the total variation mixing time of the chain via*

$$t_m = \min\left\{t \geq 0 : \max_{x \in S} \left\|P^t(x,\cdot) - \pi\right\|_{TV} < \frac{1}{4}\right\},$$

Intuitively, the mixing time measures the number of steps required for the distribution of states of the Markov chain to become close to its stationary distribution (here $1/4$ can be replaced with any appropriately small constant; $1/4$ is the convention). This concept is important in various applications, including statistical physics, randomized algorithms, and probabilistic combinatorics [66]. Due to its significance, several methods have been employed to analyze mixing times, such as spectral gap analysis and coupling methods [4]. Explicit calculations of mixing times for particular Markov chains have become an active research area. Historically, explicit mixing times have been studied extensively in contexts such as card shuffling processes [15, 83, 129] and interacting particle systems [11, 65]. Moreover, with recent advancements in graph theory, understanding mixing times

on networks has become an essential task; for example, [108] establishes mixing times for time-homogeneous Markov chains on Erdős–Rényi graphs, whereas [55] analyzes how specific graph structures can be leveraged to achieve faster mixing.

The lamplighter walk introduces additional information by placing so-called lamps on the edges of the graph, where the lighter turns the light on or off each time it traverses an edge with a certain probability [66]. By considering the position of the lighter after each step, we obtain a Markov Chain on the space $V(G) \times \{+1, -1\}^{E(G)}$; i.e., on the collection of nodes giving the current position, and $\{+1, -1\}^{E(G)}$ giving whether the lamp at each edge is on $(+1)$ or off (-1). Assuming the lighter walks on a complete graph, the stationary distribution for the chain would be the uniform distribution over $V(G) \times \{+1, -1\}^{E(G)}$ [4]. In [66], there is a detailed discussion of such lamplighter walks; we will only point out here that the mixing time required for the lamplighter on the complete graph is $t_m = \Omega(n^2 \operatorname{polylog}(n))$, where n is the number of nodes in the graph.

Our goal is to define a lamplighter-walk-like process on our graph and to match the resulting graph after each step with the initial graph until the match corrupts, meaning the edge structure signals have been erased by the noise. Notably, the worst-case matching corruption time would be the mixing time, as after the mixing time we have lost all edge signals and are left with only a flat graph. In the sections below, we explore models and conditions under which $\Theta(n^2 \operatorname{polylog}(n))$ steps are necessary and sufficient for matching corruption, as well as scenarios where matching signal can be corrupted before mixing.

4.2 Main Theory

We aim to explore the relationship between the mixing time of Markov chain producing edge noise in the graph and the graph signal anonymization time after which A_0 (the initial graph) and A_t (the noise contaminated graph at time $t \in \mathbb{N}$) are no longer matchable (see

Definition 4.3 for a formal definition of anonymization and matchability). To incorporate Markovian noise into the edge structure of the network, we consider a lamplighter walk on the edges of the random graph.

Definition 4.2. *(Artificial Lamplighter Walks on Edges) Let $G = (V,E)$ be a graph, assumed to be connected, and consider the function $h_G : \binom{V}{2} \mapsto \{-1,1\}$, where $h_G(\{u,v\}) = \mathbb{1}\{\{u,v\} \in E\} - \mathbb{1}\{\{u,v\} \notin E\}$. We will use the nomenclature that edges $e \in E$ with $h_G(e) = 1$ are "on" and $e \in \binom{V}{2} \setminus E$ are "off". We define two types of lamplighter models. Note that by switching the lamps on or off, the lamplighter changes E, and thus the graph G. We use G_0 to denote the initial graph and G_t to denote the graph after time t.*

1. *(Standard Lamplighter Walk on Edges) Let $G_t = (V,E_t)$ be the state of the graph at time t with $G_0 = G = (V,E_0 = E)$. The lamplighter here can be modeled via a time homogeneous Markov chain on $V \times \{-1,1\}^{\binom{V}{2}}$. If the lamplighter is at vertex u at time t, then the lamplighter selects a vertex in $V \setminus \{u\}$ uniformly at random and moves to that vertex. If the lamplighter moves to vertex $v \neq u$ at time $t+1$, then we consider two cases. If $h_{G_t}(\{u,v\}) = 1$, then the graph evolves via $G_{t+1} = (V,E_{t+1})$ where*

$$h_{G_{t+1}}(e) = \begin{cases} h_{G_t}(e) & \text{if } e \neq \{u,v\}; \\ -X_t h_{G_t}(e) + (1-X_t)h_{G_t}(e) & \text{if } e = \{u,v\} \end{cases} \tag{4.1}$$

where $X_t \sim Bern(q_1)$ is independent of the lamplighter walk and of $(G_s)_{s \leq t}$. If $h_{G_t}(\{u,v\}) = 0$, then the graph evolves via $G_{t+1} = (V,E_{t+1})$ where

$$h_{G_{t+1}}(e) = \begin{cases} h_{G_t}(e) & \text{if } e \neq \{u,v\}; \\ -Y_t h_{G_t}(e) + (1-Y_t)h_{G_t}(e) & \text{if } e = \{u,v\} \end{cases} \tag{4.2}$$

where $Y_t \sim Bern(q_2)$ is independent of the lamplighter walk and of $(G_s)_{s\leq t}$. Stated simply, if the lamplighter traverses an "on" edge (resp., an "off" edge), then independently with probability q_1 (resp., q_2) the lamplighter switches the edge from on to off (resp., from off to on), and with probability $1-q_1$ (resp., $1-q_2$) leaves the edge on (resp., off).

2. *(Count-Preserving Lamplighter Walk on Edges) Here the lamplighter can be modeled via a time homogeneous Markov chain on $V \times \{-1,1\}^V$. If the lamplighter is at vertex u at time t, then the lamplighter selects a pair of vertices uniformly at random from $\binom{V\setminus\{u\}}{2}$. If the selected pair is $\{v,w\}$, then we consider the following cases: if $h_{G_t}(\{u,v\}) = h_{G_t}(\{u,w\})$, the lamplighter moves to v or w uniformly at random and $G_{t+1} = G_t = (V,E_t)$; if $h_{G_t}(\{u,v\}) \neq h_{G_t}(\{u,w\})$, the lamplighter moves to v or w uniformly at random, we independently sample $X_t \sim Bern(q)$, and the evolved graph is defined via $G_{t+1} = (V,E_{t+1})$ where*

$$h_{G_{t+1}}(e) = \begin{cases} h_{G_t}(e) & \text{if } e \neq \{u,w\},\{u,v\}; \\ -X_t h_{G_t}(e) + (1-X_t)h_{G_t}(e) & \text{if } e = \{u,w\} \text{ or } \{u,v\}; \end{cases} \tag{4.3}$$

Stated simply, at time t if the lamplighter is at vertex u, then it picks two vertices v and w uniformly at random from $V \setminus \{u\}$, and moves to one of v or w at random. If there is an edge between u and v and not u and w (or vice-versa), then the lamplighter switches both edges with probability q or leaves everything unchanged with probability $1-q$. If there is/is not an edge between u and v and the same holds for u and w, then the lamplighter leaves everything unchanged. In this model, the number of lamps that are on or off will never change, hence the name count-preserving.

Remark 4.1. The standard lamplighter defined above is equivalent to a classical lamp-

lighter (in the language of [66]) on the line graph of K_V, the complete graph on V (i.e., a lamplighter on a graph with $\binom{n}{2}$ nodes, as each element of $\binom{V}{2}$ is a node). Moreover, for this model if L_t is the location of the lamplighter at time t, then the (unordered) pairs $(\{L_{2i}, L_{2i+1}\})_{i=1}^{\infty}$ are independent and uniformly distributed over $\binom{V}{2}$. By combining this with classical results from the coupon collector problem [39], we can easily apply standard results on matchability under i.i.d. noise on edges, as discussed in [132], to this model.

We are interested in our ability to recover the vertex labels of a shuffled G_t via matching it to G_0. To achieve this, we seek conditions on t such that we can still correctly match G_0 with G_t. Often a perfect matching recovering the true correspondence will be too stringent, and we here define a looser notion of matchability

Definition 4.3. *(Ratio Preserved Matchings) Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two* vertex-aligned *graphs with n nodes, and denote their adjacency matrices by A and B. By vertex-aligned, we mean that the true (but potentially unknown) correspondence between the nodes of A and B is the identity map. Let $\alpha, \beta > 0$ be constant, and define $\Pi_{n,k}$ to be the set of permutations that shuffle exactly k vertex labels for $0 < k \leq n$. We then define*

1. ***α-matchability.*** *We say that G_1 and G_2 are α-matchable if the following holds*

$$\mathbb{P}\left(\exists P \in \bigcup_{k \geq n^{\alpha}} \Pi_{n,k} \ \ s.t. \ \ \mathrm{Tr}(APBP^T) = \mathrm{Tr}(AB)\right) \leq o(1).$$

In other words, we say graphs G_1 and G_2 are α-matchable if the probability of finding a permutation shuffling at least n^{α} nodes and that provides a better matching than the true identity mapping in terms of the GMP objective function is asymptotically vanishingly small.

2. ***β-anonymization.*** *We say that G_1 and G_2 are β-anonymized if the following holds*

$$\mathbb{P}\left(\underset{P\in\Pi_n}{arg\,max}\,\mathrm{Tr}(APBP^T)\subset\bigcup_{k=0}^{n^\beta}\Pi_{n,k}\right)\leq o(1).$$

In other words, we say graphs G_1 and G_2 are β-anonymized if the probability of the optimal matching (in terms of the GMP objective function) between them is provided by a permutation that shuffles at most n^β nodes is is asymptotically vanishingly small. In other words, with high probability the optimal matching will shuffle more than n^β nodes.

Define the set $T=\left\{t : I_n\in\arg\min_{P\in\Pi_n}\|G_0-PG_tP^T\|_F\ a.s.\right\}$, and note that by taking $S=\binom{V}{2}$ as our state space and letting L_t represent the vertex traversed by the lamplighter at time t, the sequence $\{(L_{2t},L_{2t+1})\}_{t=0}^{\infty}$ forms a Markov Chain on S. Let t_{cov} be the cover time of that Markov Chain, it is clear that $t\notin T$ for all $t>t_{cov}$, as by this time, all lamps have been randomized, effectively wiping out the vertex-label signal. From [4] and Remark 4.1, we know that $t_{cov}=\Theta(n^2\log n)$ for the Standard lamplighter walk on edges. Also, following Chapters 11 and 19 of [66], let t_m denote the mixing time of the Markov Chain (L_{2t},L_{2t+1}). We have that for the standard lamplighter walk, $t_m\lesssim t_{cov}\lesssim\log(n)t_m$, meaning that the mixing time and the covering time for this particular Markov Chain differ only by logarithmic factors.

Our focus now shifts to exploring models where signal recovery is possible up to t_{cov} and identifying cases where signal recovery fails before t_{cov}. We begin by considering G as an Erdős–Rényi random graph.

4.2.1 Standard Lamplighter Walk on Erdős–Rényi Graphs

We first consider the Standard lamplighter walk on an Erdős–Rényi graph $G_0 = (V, E) \sim ER(n, p)$. Below, we will assume that the initial starting vertex for the lamplighter is uniformly distributed over V. Letting G_t be evolved graph after t steps of the lamplighter walk, we consider here two edge-cases for the parameters q_1, q_2.

In the first case, q_1, q_2 are chosen so that $p(1-q_1) + (1-p)q_2 = p$ (for example, we could set $q_2 = p^2$, $q_1 = p(1-p)$). In this case, if an edge is present with probability p before a visit from the lamplighter, then the edge will be present with probability p after the lamplighter traverses the edge as well. In this case, it is immediate that $G_t = (V, E_t) \sim ER(n, p)$ as well. Moreover, the edgewise correlation between G_t (with adjacency matrix A_t) and G_0 (with adjacency matrix A_0) can be computed via (where $\mathfrak{p}_{t,i,j} = \mathbb{P}(\text{edge } \{i, j\} \text{ is not traversed by the lamplighter by time } t)$)

$$\begin{aligned} \operatorname{corr}(A_{0,i,j}, A_{t,i,j}) &= \frac{\mathbb{E}(A_{0,i,j}A_{t,i,j}) - p^2}{p(1-p)} \\ &= \frac{p\mathfrak{p}_{t,i,j} + p^2(1-\mathfrak{p}_{t,i,j}) - p^2}{p(1-p)} = \mathfrak{p}_{t,i,j} \end{aligned}$$

In Appendix 4.5.1, we show that

$$\mathfrak{p}_{t,i,j} = \exp\left\{-\Theta\left(\frac{t}{n^2}\right)\right\} \tag{4.4}$$

Therefore, as the matchability phase transition here occurs at correlation of order $\sqrt{\frac{\log n}{n}}$, see [73, 76, 132], we see that $t = \omega(n^2 \log n)$ is sufficient for the graphs to be effectively anonymized and that if $t = o(n^2 \log n)$ matchability is preserved.

With the matchings defined in Definition 4.3, we consider the case of the max entropy lamplighter, i.e., when $q_1 = q_2 = 1/2$. In this case, the limiting G_t is ER$(V, 1/2)$ which is

the maximum entropy graph distribution over V). At each step, we again match G_t with G_0, and we have the following theorem (proven in Appendix 4.5.2).

Theorem 4.1. *Let $\{G_t\}_{t\in\mathbb{N}}$ be as described above where $q_1 = q_2 = 1/2$. Let $\alpha, c > 0$ be fixed constants such that $\alpha > 5c$. If $t \leq cn^2 \log n$, then G_0 and G_t are α-matchable and the α-matchability is preserved almost surely. More explicitly, we have*

$$\mathbb{P}\left(\exists P \in \bigcup_{k\geq n^\alpha} \Pi_{n,k} \text{ s.t. } \mathrm{Tr}(A_0 P A_t P^T) \geq \mathrm{Tr}(A_0 A_t)\right) \leq 2exp\left\{-\Omega\left(\frac{n^{2\alpha-5c}}{\log n}\right)\right)$$

For an anonymization bound, we consider the time t_{cov} such that the lamplighter has visited every edge in $\binom{V}{2}$. At this time, $G_{t_{cov}}$ is effectively ER(V,1/2) independent of G_0 and any permutation is equally likely to be optimal. As noted in Remark 4.1, the (unordered) pairs $(\{L_{2i}, L_{2i+1}\})_{i=1}^{\infty}$ are independent and uniformly distributed over $\binom{V}{2}$. Let $\tilde{t}_{cov}$ be the number t such that $(\{L_{2i}, L_{2i+1}\})_{i=1}^{t}$ contains every edge in $\binom{V}{2}$. Therefore, standard coupon collector asymptotics yield that

$$\mathbb{P}(t_{cov} \geq 4n^2 \log(n)) \leq \mathbb{P}(\tilde{t}_{cov} \geq 4n^2 \log(n)) \leq n^2 e^{-4\log n} = n^{-2}.$$

We then have the following (proven in Appendix 4.5.3)

Theorem 4.2. *Let $\{G_t\}_{t\in\mathbb{N}}$ be as described above where $q_1 = q_2 = 1/2$. If $t \geq 4n^2 \log n$, then for any $\beta \in (0,1)$, G_0 and G_t have been β-anonymized and the anonymization is achieved almost surely. More explicitly, we have*

$$\mathbb{P}\left\{argmax_{Q\in\Pi_n} Tr(A_0 Q A_t Q^T) \subset \bigcup_{k=0}^{n^\beta} \Pi_{n,k}\right\} \leq n^{-2} + (2n^\beta)^{-n/(2n^\beta+1)}$$

4.2.2 (Partial) Anonymization Before Mixing for Structured Graphs

We have observed that for simple Erdős–Rényi graphs, there is an agreement between the mixing time of the noise and the anonymization time up to a polylogarithmic factor. Now, we seek to identify models where the matching fails before the noise mixes (at least partially) the network. To explore this, we consider a more structured graph model: the Stochastic Block Model (SBM). The specific model we will examine is

$$\mathrm{SBM}\left(K_n, \Lambda_n = [\Lambda_{i,j;n}], \tau_n, \vec{n} = (n_1, n_2, \cdots, n_{K_n})\right) \subseteq \mathscr{G}_n,$$

Within this SBM model, we define the lamplighter walk as follows

Definition (1 (continued)). *(Artificial Lamplighter Walks on Edges)*

3. *(SBM Lamplighter Walk on Edges) Let $G_t = (V, E_t)$ be the state of the graph at time t with G_0 sampled from the SBM model specified above. The lamplighter here is again modeled by a time homogeneous Markov chain on $\mathscr{C} \times V \times \{-1,1\}^{\binom{V}{2}}$, where $\mathscr{C}$ is the collection of all communities from the graph. If the lighter is at community i (denoted C_i) and vertex $u \in C_i$, then for the next step, the lamplighter stays or leaves community C_i with equal probability (i.e., equal to 1/2).*

    i. *If the lamplighter selects to stay in community C_i, then the lighter selects a vertex $v \in C_i \setminus \{u\}$ and moves to the vertex. If the lamplighter moves to vertex $v \neq u$ at time $t+1$, then we consider two cases. If $h_{G_t}(\{u,v\}) = 1$, then the graph evolves via $G_{t+1} = (V, E_{t+1})$ where*

$$h_{G_{t+1}}(e) = \begin{cases} h_{G_t}(e) & \text{if } e \neq \{u,v\}; \\ -X_t h_{G_t}(e) + (1 - X_t) h_{G_t}(e) & \text{if } e = \{u,v\} \end{cases} \tag{4.5}$$

where $X_t \sim Bern(q_{i;1})$ is independent the lamplighter walk and of $(G_s)_{s\le t}$. If $h_{G_t}(\{u,v\}) = 0$, then the graph evolves via $G_{t+1} = (V, E_{t+1})$ where

$$h_{G_{t+1}}(e) = \begin{cases} h_{G_t}(e) & \text{if } e \neq \{u,v\}; \\ -Y_t h_{G_t}(e) + (1-Y_t) h_{G_t}(e) & \text{if } e = \{u,v\} \end{cases} \tag{4.6}$$

where $Y_t \sim Bern(q_{i;2})$ is independent the lamplighter walk and of $(G_s)_{s\le t}$.

ii. If the lamplighter selects to leave C_i, then the lighter first choose a new community $C_j \neq C_i$ uniformly at random from all $K_n - 1$ non-C_i communities. The lamplighter then pick a node $w \in C_j$ and move to w at time $t+1$, and then consider two cases. If $h_{G_t}(\{u,w\}) = 1$, then the lighter pick at random $w' \in C_j$ such that $h_{G_t}(\{u,w'\}) = 0$ and set $h_{G_{t+1}}(\{u,w\}) = 0$, $h_{G_{t+1}}(\{u,w'\}) = 1$. If $h_{G_t}(\{u,w\}) = 0$, then the lighter pick at random $w' \in C_j$ such that $h_{G_t}(\{u,w'\}) = 1$ and set $h_{G_{t+1}}(\{u,w\}) = 1$, $h_{G_{t+1}}(\{u,w'\}) = 0$.

With the SBM Lamplighter Walk on Edges defined, we have the following result. Note that the proof of Theorem 4.3 can be found in Appendix 4.5.4

Theorem 4.3. *Assume there exist constants $b_1, b_2, b_3 > 0$ and $0 < a_1 < a_2 < a_3$ such that (assuming without loss of generality that $n_1 = \min n_i$)*

$$\begin{aligned} n_1 = b_1 n^{a_1}; \quad n^* := \max_i n_i = b_2 n^{a_3}; \\ \Lambda_{1,j;n} \le b_3 \frac{n^{2a_2}}{K_n n_j n_1} \text{ for all } j \neq 1; \\ \Lambda_{i,j;n} = \omega\left(\frac{\log n}{n_i n_j}\right) \text{ for all } i, j \text{ with } i \neq j \end{aligned}$$

Consider the SBM lamplighter defined above where $q_{i;1} = q_{i;2} = 1/2$ for all i. Define the

event

$$\mathscr{E}_{n,\beta} = \{\beta\text{-anonymization of community 1 after n steps}\},$$

and $N = n^{a_3+a_2}\log^{3/2}(n)$. *We then have for suitable constants* $C_1, C_2 > 0$, *and n sufficiently large*

$$\mathbb{P}(\mathscr{E}_{N,\beta}) \geq 1 - C_1\left(K_n^2 n^{-3} + n^{-2a_1} + e^{-C_2\log^{1/2} n} + K_n n^{a_2-a_3}\right)$$

Note that if $K_n n^{a_2-a_3} = o(1)$ (which holds for example if $a_2 < 1/2$, $a_3 = 3/4$, $K_n = \Theta(n^{1/4})$), then the above theorem says that after $n^{a_3+a_2}\log^{3/2}(n)$ steps, community 1 has been effectively anonymized with high probability. Note that the mixing time of the lamplighter on the full graph is bounded below by the cover time (up to a log factor) of the edges within the largest community, and this cover time is of the order (again, up to a log factor) $n^{2a_3}\log^{3/2}(n)$. Hence, after $n^{a_3+a_2}\log^{3/2}(n)$ steps, with probability near 1 the lamplighter has not mixed globally.

Theorem 4.3 shows that, under the SBM Lamplighter Walk on Edges, the matching corrupts locally before the Markovian noise globally mixes. We note that Theorem 4.1 does not apply here because the community structure introduces different probabilities for walking within or between communities. We conclude this section by noting that, although for the SBM model, matching locally corrupts earlier than the mixing time of the noise, Theorem 4.3 only provides a condition for the inability to achieve exact matching as defined in [132]. According to Theorem 4.3, after $\Theta(n^{a_2+a_3}\,\mathrm{polylog}(n))$ steps, we are only guaranteed anonymization of a constant number of communities, not the entire network. We hypothesize that ensuring even partial global recovery still requires $\Omega(n^2\,\mathrm{polylog}(n))$ time, and further analysis of such bounds is a topic of future work.

4.3 Experiments

We now present both simulated and real data experiments to investigate the relationship between anonymization time and the mixing time. For the simulated data, we also provide regression results that highlight the leading order of the anonymization time. As discussed in Section 4.2, in the standard lamplighter walk, the mixing time t_m and the cover time t_c differ only by a logarithmic factor (with the cover time providing a lower bound in the less standard settings). Since computing the mixing time directly can be challenging in experiments, we use the cover rate as a surrogate to provide a good approximation of the mixing time.

To address the computational intractability of solving the exact graph matching problem, we use the Seeded Graph Matching algorithm introduced by [76] To expedite the experiments for the SBM walk and real data, we initialize the matching algorithm at the graph ground truth—the identity matrix. Note that when local anonymization occurs, the algorithm's optimizer should move away from the ground truth, but not by much. Therefore, this initialization effectively saves computational resources. Our seeds are randomly selected from the set of nodes, and the algorithm ensures that the matching results map these seeded nodes to themselves. Unless stated otherwise, we uniformly select 5% of the nodes as seeds.

4.3.1 Simulated lamplighter walks on ER and SBM graphs

We begin by examining the global lamplighter walk model on Erdős–Rényi graphs. We sample Erdős–Rényi initial graphs with $p = 0.5$ and node counts $n = 49, 100, 144, 225, 324, 729$. For each graph, we perform a global lamplighter walk. For smaller graphs ($n = 49, 100, 144$), we check matching correctness after each move. To save computational resources for larger graphs ($n = 225, 324, 729$), we check matching after a fixed number of steps, specifically

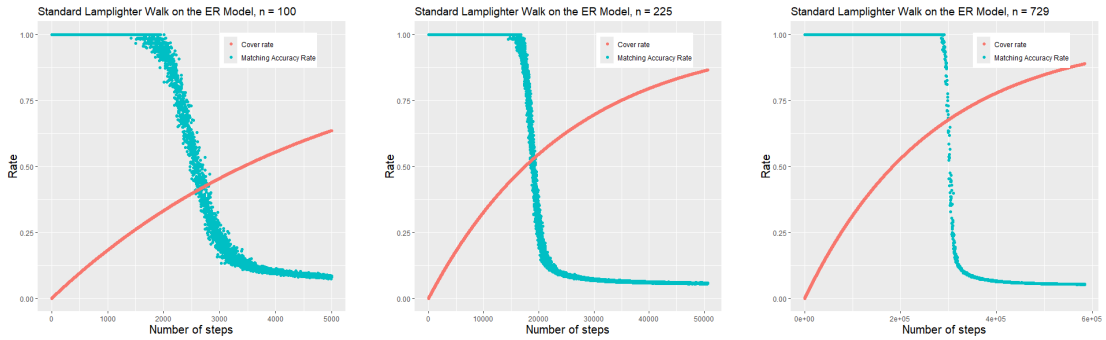

Figure 4.1: We plot matching correctness vs. number of steps for $n = 100$ (left), $n = 225$ (middle) and $n = 729$ (right), the results for other chosen n values can be found in Appendix 4.5.5. For all these plots, we further impose the cover rate of the edges vs number of steps.

$s_n = 3, 30, 300$ for $n = 225, 324, 729$ respectively.

In Figure 4.1, we provide figures showing matching correctness versus the number of steps for $n = 100, 225$ and $n = 729$. Results for the other values of node counts can be found in Appendix 4.5.5. Additionally, we plot the cover rate of the edges against the number of steps for these experiments. We then present, in Figure 4.2, a log-log plot of the number of steps needed until a 0.7-anonymization happens, along with a fitted regression line. The slope of the fitted line is slightly over 2, which aligns with our theoretical result that $\Theta(n^2 \operatorname{polylog}(n))$ steps are required to destroy matchability.

Next, we consider the more structured Stochastic Block Model (SBM) to illustrate a scenario where anonymization occurs before mixing.

As discussed in Section 4.2.2, we sample graphs from the following SBM model:

$$\operatorname{SBM}\left(K = \left\lfloor \frac{n - n^{1/4} - n^{4/3}}{n^{2/3}} \right\rfloor, \Lambda = \frac{\log(n)}{n^{3/4}} J + \operatorname{diag}(1/2), \tau = \tau_n, \vec{n} = (n^{1/4}, n^{2/3} \cdot \mathbf{1}_{K-2}^T, n^{3/4})\right)$$

where $\tau_n(j) = \sum_{k=1}^{K} \mathbf{I}_{\left\{j \geq \sum_{i=1}^{k} \vec{n}_i\right\}}$, $\mathbf{1}_m$ is the $m \times 1$ all 1 vector and $\mathbf{I}$ is the indicator function. It is easy to verify these chosen parameters satisfied the conditions in Theorem 4.3. For the experiments, we select node counts $n = 81, 256, 625$ and perform the SBM lamplighter walk on these sampled networks. Matching is conducted after every step for $n = 81$, after

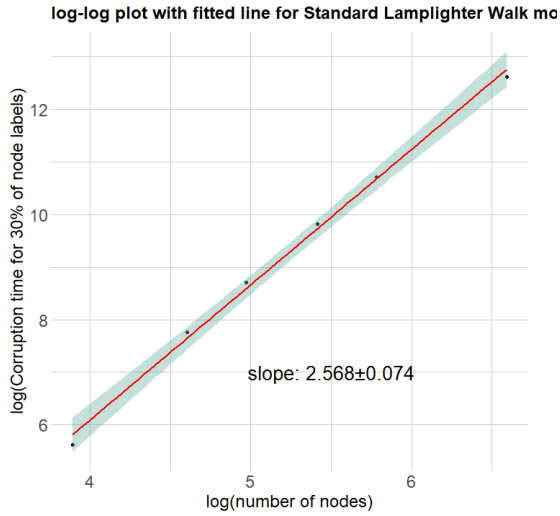

Figure 4.2: We plot the log-log result for steps needed until a 0.7-anonymization happens (recall that a 0.7-anonymization happens when the matching correctness w.r.t. the ground truth is less than $\beta = 0.7$) vs number of nodes, and impose the fitted regression line on these points.

90 steps for $n = 256$, and after 2100 steps for $n = 625$. Additionally, we initialize the matching algorithm at the identity matrix.

In Figure 4.3, we present selected plots for the case $n = 256$: the top-left panel shows the matching correctness versus the number of steps for the entire network; the top-right panel shows matching correctness versus the number of steps for community 1 (the smallest community); the bottom-left panel shows matching correctness versus the number of steps for community 2 (a randomly chosen community with size $n^{2/3}$); and the bottom-right panel shows matching correctness versus the number of steps for community K (the largest community). In all these plots, we also include the edge cover rate against the number of steps. Plots for other communities in the $n = 256$ case, as well as additional plots for the $n = 81$ and $n = 625$ cases, can be found in Appendix 4.5.5.

We observe that the smallest community exhibits noticeably faster anonymization, which aligns closely with our proposed theoretical framework. Additionally, the largest community demonstrates a slower anonymization rate compared to the other communities. It is

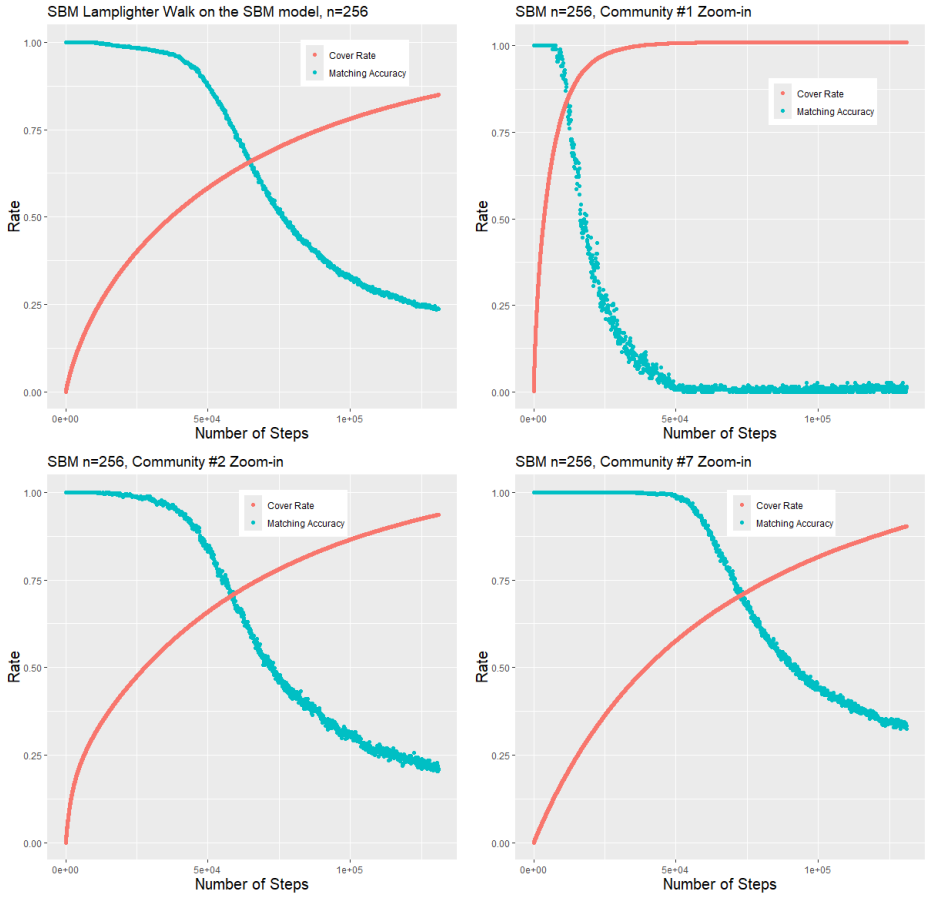

Figure 4.3: We present selected plots for the case $n = 256$: the top-left panel shows the matching correctness versus the number of steps for the entire network; the top-right panel shows matching correctness versus the number of steps for community 1 (the smallest community); the bottom-left panel shows matching correctness versus the number of steps for community 2 (a randomly chosen community with size $n^{2/3}$); and the bottom-right panel shows matching correctness versus the number of steps for community K (the largest community). In all these plots, we also include the edge cover rate against the number of steps. Plots for other communities in the $n = 256$ case, as well as additional plots for the $n = 81$ and $n = 625$ cases, can be found in Appendix 4.5.5.

also important to note that, when comparing this SBM lamplighter setting to the standard lamplighter setting on ER graphs, the transition away from matchability appears to be less sharp.

We also plot a log-log graph of the number of steps required until a $\Theta(n^{1/4})$-anonymization is achieved (here the constant is chosen to be $c = 15$) versus the number of nodes, with a fitted regression line superimposed on these points. As observed, the slope of the regression

Figure 4.4: We plot the log-log graph of the number of steps required until a $\Theta(n^{1/4})$-anonymization is achieved (here the constant is chosen to be $c = 15$) versus the number of nodes, with a fitted regression line superimposed on these points.

line is less than 2, and is about 0.5 less than the slope obtained for the standard lamplighter walk on ER networks. This value aligns with the suggested by our theoretical framework in Section 4.2.2.

4.3.2 Real data experiments

To validate the proposed theoretical results on a real-world network, we apply the count-preserving lamplighter model to a friendship network from [79]. We use the entire network from `https://snap.stanford.edu/data/ego-Facebook.html`, and extract the induced subgraph containing nodes #1921 to #2640. The corresponding adjacency matrix is shown in the left panel of Fig. 4.5. From the adjacency matrix, we observe no clear pattern or structure, so it is reasonable to assume that within this network, individuals are likely to form friendships with one another with equal probability (noting that of course this is not the case). Additionally, the adjacency plot indicates that the graph is relatively dense.

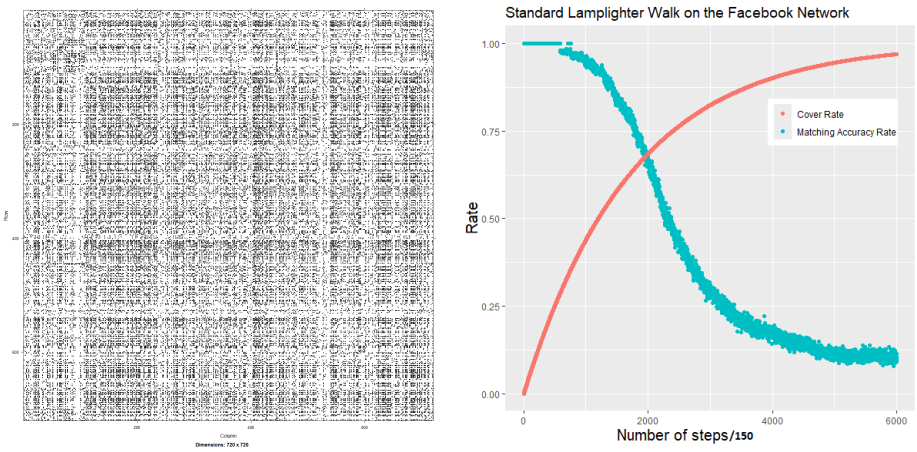

Figure 4.5: Left: The adjacency matrix of the induced subgraph that we perform our lamplighter walk on. Right: We plot the matching correctness vs. iterations for the count-preserving lamplighter walk on the Facebook Network (blue). We further impose the cover-rate vs. iteration curve (red).

On the selected network, we run the standard lamplighter walk for 900,000 steps, performing matching to the original network every 150 iterations. The lamplighter walk introduces noise that can be interpreted as individuals forming new friendships or losing contact with those they have not interacted with recently. We also track the covering rate after each step, defined as:

$$\frac{\#\text{ distinct edges traversed} + \#\text{ distinct edges selected}}{\binom{720}{2}}.$$

In the right panel of Fig. 4.5, we plot the matching correctness versus iteration curve, alongside the covering rate versus iteration curve. The results clearly show a similar pattern to what we observed for the standard lamplighter walk on Erdős–Rényi graphs, as seen in Fig. 4.1.

To further validate the proposed results on a more structured real-world network, we implement the SBM lamplighter walk on an email communication network generated from a large European research institution. In this network, nodes represent individuals, and edges indicate email exchanges between them. The department to which each individual

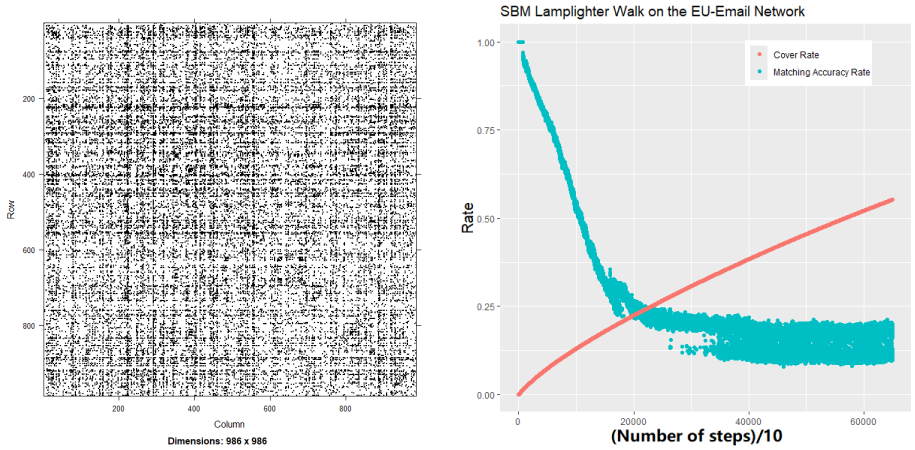

Figure 4.6: Left: The adjacency matrix of the undirected EU Email communication network. Right: We plot the matching correctness vs. iterations for the community structure and count-preserving lamplighter walk on the EU Email Communication Network. We further impose the cover-rate vs. iteration curve.

belongs is provided as a node attribute, allowing us to treat each department as a distinct community. For more detailed information about this network and its applications, refer to the original source [64].

We obtained the network from `https://snap.stanford.edu/data/email-Eu-core.html`. For simplicity and better alignment with our theoretical framework, we made the network undirected. The corresponding adjacency matrix after making the graph undirected is shown in the left panel of Fig. 4.6. As in the Stochastic Block Model (SBM) experiments discussed earlier, we apply the community structure and count-preserving lamplighter walk to this network. To save computational resources, matching with the original network is performed every 220 steps. In the right panel of Fig. 4.6, we plot the matching correctness versus iteration curve, along with the edge cover rate versus iteration curve. The plot shows a similar pattern to what we observed for the community structure and count-preserving lamplighter walk on the simulated SBM graph in Fig. 4.3.

To gain a better understanding of anonymization within communities, Fig. 4.7 shows the matching correctness versus iteration plot and the cover rate versus iteration curve for

the subgraph induced by nodes from community (department) #1 (left panel, 49 members) and by nodes from community (department) #8 (right panel, 49 members). From the plots, we observe that anonymization occurs much earlier in some communities. This observation supports our hypothesis that faster anonymization in structured networks is due to the anonymization of certain communities, while globally corrupting the entire matching still requires time on the same order as the time needed for the noise to globally mix.

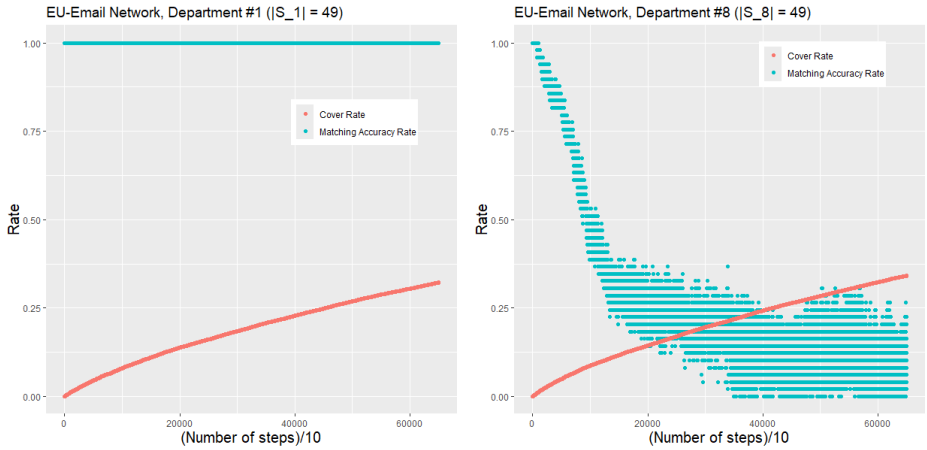

Figure 4.7: We plot the matching correctness vs iteration plot and the cover rate vs iteration curve for only the subgraph induced by nodes from community (department) #15 (left panel) and by nodes from community (department) #1 (right panel)

4.4 Conclusion and Discussion

In this chapter, we explored the relationship between the anonymization time of graph signals and the mixing time of Markovian noise on the graph, across various graph models under different lamplighter-like walk schemes. By examining both theoretical bounds and simulation results, we established that the time required to disrupt the matchability between an original graph and its noisy counterpart aligns closely with the mixing time of the noise, especially in the context of Erdős–Rényi random graphs. Specifically, our results demonstrate that for the standard lamplighter walks on these graphs, the matching

between the original and noisy graphs deteriorates after $\Theta(n^2 \operatorname{polylog}(n))$ steps, consistent with the expected mixing time. However, in some structured graph models, particularly the Stochastic Block Model (SBM), which features inherent community structures, anonymization occurs before the noise has fully mixed. This highlights distinct dynamics in such models. Our findings suggest that in SBM networks with n nodes and communities with sufficiently different sizes, the time required to break matchability locally is approximately $\Theta(n^\alpha \operatorname{polylog}(n))$ for some $\alpha < 2$, significantly earlier than the $\Theta(n^2 \operatorname{polylog}(n))$ time required for the noise to mix. Note that the mixing time depends on the number of edges. Thus, for sparser graphs (with fewer than $\Theta(n^2)$ edges), the mixing time of the noise could be smaller than $\Theta(n^2 \log n)$. Our goal here is to provide a simple example where anonymization occurs faster than mixing; therefore, we do not explore such sparser graphs in this context. This indicates that the presence of community structures can accelerate the anonymization process, reducing the number of steps needed to obscure the original graph's structure.

To validate our theoretical predictions, we performed a series of simulations on both synthetic and real-world networks. The simulations supported our theoretical claims, with regression analysis showing that the slopes of the fitted lines aligned well with the predicted leading order of the anonymization time. Specifically, for Erdős–Rényi graphs, the observed slopes were slightly above 2, consistent with the theoretical $\Theta(n^2 \operatorname{polylog}(n))$ steps. In contrast, for SBM graphs with communities of different sizes, the slope was remarkably smaller compared to the slopes for the Erdős–Rényi graphs, confirming our theoretical predictions.

We further extended our analysis to real-world networks, including a Facebook friendship network and an email communication network from a European research institution. These experiments provided practical insights into how the proposed models perform on real data, demonstrating the applicability of our theoretical results beyond synthetic set-

tings. In particular, the count-preserving lamplighter walk on the Facebook network and the SBM lamplighter walk on the email network both showed that anonymization occurred in line with the expected theoretical bounds, underscoring the soundness of our theories.

The findings of this study open new avenues for future research. First, while our results demonstrate that community structures can accelerate anonymization, our current results are built only based on the edge structure information and it worth exploring the effect of node similarities in this context. Additionally, extending our models to other types of structured graphs, such as hierarchical stochastic block models [27, 69, 77, 90, 93], block models with overlapping communities [3], or graphs with more complex structures [7, 51], could provide deeper insights into the relationship between graph structure and anonymization time. Moreover, if we consider collections of temporal networks, dependencies may emerge in the propagation of signals from one graph to the next, extending beyond the scope of Markovian noise. Further analysis of matchability and other statistical inferences on these functional dependencies could provide invaluable insights for more realistic real-data modeling via networks. Finally, while the Seeded Graph Matching (SGM) algorithm provided a practical solution for matching noisy graphs to their originals, it would be valuable to explore more advanced uses of seed information—beyond simply constraining the solver to a particular subspace of the function space—to improve the efficiency and accuracy of the matching process; see [84] for steps in this direction. In particular, cases where SGM results differ significantly from the ground truth suggest the need for more sophisticated ways to utilize seed information, potentially through statistical learning processes.

4.5 Proof of Theorems

4.5.1 Derivation of Eq. 4.4

We seek to show the given bound on $\mathfrak{p}_{t,i,j}$. We first define the Global Lamplighter Walk analogously to the standard lamplighter, except that the position of the lamplighter does not matter. At the $(t+1)$st step, the lamplighter randomly chooses any $e \in \binom{V}{2}$, and defines $h_{G_{t+1}}(e)$ as in Eqs. 4.1 and 4.2. Consider the three events,

$$\mathscr{E}_{1,t} = \{\text{the lamplighter in the standard walk does not traverse } \{i,j\} \text{ by } t\}$$

$$\mathscr{E}_{2,t} = \left\{\text{the lamplighter in the standard walk does not traverse } \{i,j\} \text{ in } \{(L_{2i}, L_{2i+1})\}_{i=0}^{\lfloor \frac{t-1}{2} \rfloor}\right\}$$

$$\mathscr{E}_{3,t} = \{\text{the lamplighter in the global walk does not traverse } \{i,j\} \text{ by } t\}$$

We then have

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^t = \mathbb{P}(\mathscr{E}_{3,t}) \leq \mathbb{P}(\mathscr{E}_{1,t}) \leq \mathbb{P}(\mathscr{E}_{2,t}) = \left(1 - \frac{1}{\binom{n}{2}}\right)^{\lfloor \frac{t+1}{2} \rfloor}$$

Using the fact that for $n > 3$,

$$\left(1 - \frac{1}{\binom{n}{2}}\right) \leq e^{-\frac{1}{\binom{n}{2}}}, \quad \left(1 - \frac{1}{\binom{n}{2}}\right) \geq e^{-\frac{1}{\binom{n}{2}-1}},$$

we have that

$$\mathfrak{p}_{t,i,j} \leq \exp\left\{-\frac{\lfloor \frac{t+1}{2} \rfloor}{\binom{n}{2}}\right\}, \quad \mathfrak{p}_{t,i,j} \geq \exp\left\{-\frac{t}{\binom{n}{2}-1}\right\}$$

as desired.

4.5.2 Proof of Theorem 4.1

Here, we will follow standard concentration inequality theory to establish the matchability result. Let $B = A_t$, and consider for a fixed permutation matrix P, with associated permutation σ_P, the following

$$\begin{aligned} S_{n,P} := \mathrm{Tr}(APBP^T) - \mathrm{Tr}(AB) &= \mathrm{Tr}\left(A[PBP^T - B]\right) \\ &= 2 \sum_{\substack{\{i,j\} \in \binom{V}{2} \text{ s.t.} \\ \{i,j\} \neq \{\sigma_P(i), \sigma_P(j)\}}} A_{i,j} \left(B_{\sigma_P(i),\sigma_P(j)} - B_{i,j}\right). \end{aligned}$$

Straightforward computations then yield

$$\begin{aligned} \mathbb{E}(S_{n,P}) &= 2 \sum_{\substack{\{i,j\} \in \binom{V}{2} \text{ s.t.} \\ \{i,j\} \neq \{\sigma_P(i), \sigma_P(j)\}}} \mathbb{E}\left\{\left(B_{\sigma_P(i),\sigma_P(j)} - B_{i,j}\right) \mid A_{i,j} = 1\right\} \mathbb{P}(A_{i,j} = 1) \\ &= 2p \sum_{\substack{\{i,j\} \in \binom{V}{2} \text{ s.t.} \\ \{i,j\} \neq \{\sigma_P(i), \sigma_P(j)\}}} \left\{\mathbb{E}\left(B_{\sigma_P(i),\sigma_P(j)} \mid A_{i,j} = 1\right) - \mathbb{E}\left(B_{i,j} \mid A_{i,j} = 1\right)\right\} \end{aligned}$$

Noting that $\mathfrak{p}_{t,i,j}$ is independent of i, j, we drop these indices and write $\mathfrak{p}_t$ moving forward. We now consider computing each conditional expectation via considering cases for index i and j after t steps.

1. We first consider $\mathbb{E}\left\{B_{\sigma_P(i),\sigma_P(j)} \mid A_{i,j} = 1\right\}$. If $\{\sigma_P(i), \sigma_P(j)\}$ was traversed by the lamplighter by time t, then the edge has probability $1/2$ regardless of the state of $A_{\{\sigma_P(i),\sigma_P(j)\}}$ and $A_{i,j}$. This conditional expectation is then equal to

$$\mathbb{E}\left\{B_{\sigma_P(i),\sigma_P(j)} \mid A_{i,j} = 1\right\} = \frac{1}{2}(1 - \mathfrak{p}_t) + p\mathfrak{p}_t$$

2. We next consider $\mathbb{E}\left\{B_{i,j} \mid A_{i,j} = 1\right\}$. If $\{i, j\}$ was traversed by the lamplighter by

time t, then the edge has probability $1/2$ regardless of the state of $A_{i,j}$; else $B_{i,j} = A_{i,j}$. This conditional expectation is then equal to

$$\mathbb{E}\left\{B_{i,j} \mid A_{i,j} = 1\right\} = \frac{1}{2}(1-\mathfrak{p}_t) + \mathfrak{p}_t$$

Putting the above together, we get that

$$\mathbb{E}(S_{n,P}) = 2\underbrace{\left|\{i,j\} \in \binom{V}{2} \text{ s.t. } \{i,j\} \neq \{\sigma_P(i), \sigma_P(j)\}\right|}_{:=\eta_P} \mathfrak{p}_t p(p-1)$$

If P shuffles k vertices, then $\eta_P \in [(n-2)k/2, nk]$ and this expectation can be bound via

$$(n-2)k\mathfrak{p}_t p(p-1) \leq \mathbb{E}(S_{n,P}) \leq 2nk\mathfrak{p}_t p(p-1)$$

Moreover, $S_{n,P}$ can be realized as a function of the following random variables:

i. The at most $2nk$ edges, $A_{i,j}$, involved in the summation. Changing any of these could change the summation $S_{n,P}$ by at most 4.

ii. The at most $2(t+1)$ random variables dictating the moves and flips of the lamplighter walk. Changing a single move of the lamplighter would change at most 2 of the $B_{i,j}$'s and hence could change the summand by at most 8. Changing a flip of the lamplighter could change at most one $B_{i,j}$ and hence would change the summand by at most 2.

McDiamid's inequality then gives us that for n sufficiently large (recalling that we assume

there exists a constant $c > 0$ such that $t \leq cn^2 \log n$)

$$\begin{aligned}
\mathbb{P}(S_{n,P} \geq 0) &\leq \mathbb{P}(|S_{n,P} - \mathbb{E}(S_{n,P})| \geq |\mathbb{E}S_{n,P}|) \\
&\leq 2\exp\left\{ -\frac{(n-2)^2 k^2 p^2 (p-1)^2 \exp\left\{ -\frac{4t}{n(n-1)-2} \right\}}{8(nk+t+1)} \right\} \\
&\leq 2\exp\left\{ -\frac{(n-2)^2 k^2 p^2 (p-1)^2 n^{-5c}}{8(nk+cn^2 \log n+1)} \right\}
\end{aligned}$$

Let $\Pi_{n,k}$ be the set of permutations shuffling exactly k vertex labels, and define the event

$$\beta_{k,t} = \{\exists P \in \Pi_{n,k} \text{ s.t. } S_{n,P} \geq 0\}.$$

A union over the at most n^k permutation matrices in $\Pi_{n,k}$ yields

$$\mathbb{P}(\beta_{k,t}) \leq 2\exp\left\{ -\frac{(n-2)^2 k^2 p^2 (p-1)^2 n^{-5c}}{8(nk+cn^2 \log n+1)} + k \log n \right\}$$

Note next that (recalling there exists a constant $\alpha > 5c$ such that $k \geq n^\alpha$)

$$\frac{\frac{(n-2)^2 k^2 p^2 (p-1)^2 n^{-5c}}{8(nk+cn^2 \log n+1)}}{k \log n} = \Omega\left(\frac{p^2 (p-1)^2 n^{\alpha-5c}}{\log^2 n} \right) = \omega(1)$$

Therefore, taking a union over such k yields

$$\mathbb{P}\left(\bigcup_{k \geq n^\alpha} \beta_{k,t} \right) \leq 2\exp\left\{ -\Omega\left(\frac{n^{2\alpha-5c}}{\log n} \right) + \log n \right\} \leq 2\exp\left\{ -\Omega\left(\frac{n^{2\alpha-5c}}{\log n} \right) \right\}$$

as desired.

4.5.3 Proof of Theorem 4.2

Abusing notation slightly, we will use Π_n to denote the set of permutation of $\{1,2,\cdots,n\}$ (denoted by lower case Greek letters) and the set of $n \times n$ permutation matrices (denoted by capital Roman letters). We say that permutations σ and τ in Π_n disagree at location $i \in \{1,2,\cdots,n\}$ if $\sigma(i) \neq \tau(i)$. Consider $\mathfrak{n}_\beta = \lfloor n/(2n^\beta+1) \rfloor$ disjoint sets in $\{1,2,\cdots,n\}$, denote these sets via $\{S_i\}_{i=1}^{\lfloor n/(2n^\beta+1)\rfloor}$ For each i, let $\sigma^{(i)}$ be permutation of $\{1,2,\cdots,n\}$ that fixes all elements of $\cup_{j\neq i} S_j$ and is a (in cycle decomposition) cyclic derangement (i.e., a cycle of length $2n^\beta+1$) of S_i. Note that if $(\sigma^{(i)})^k$ denotes the permutation

$$(\sigma^{(i)})^k = \underbrace{\sigma^{(i)} \circ \cdots \circ \sigma^{(i)}}_{k \text{ compositions}},$$

then for all $k_1, k_2 \in \{1,2,\cdots,2n^\beta\}$ with $k_1 \neq k_2$, we have that $(\sigma^{(i)})^{k_1}$ and $(\sigma^{(i)})^{k_2}$ disagree in exactly $2n^\beta+1$ locations. Consider all permutations of the form

$$\sigma = (\sigma^{(1)})^{k_1} \circ (\sigma^{(2)})^{k_2} \cdots \circ (\sigma^{(\mathfrak{n}_\beta)})^{k_{\mathfrak{n}_\beta}} \tag{4.7}$$

for $(k_1,k_2,\cdots,k_{\mathfrak{n}_\beta}) \in \otimes_{i=1}^{k_{\mathfrak{n}_\beta}} \{1,2,\cdots,2n^\beta\}$. Each pair of such σ's with distinct $(k_1,k_2,\cdots,k_{\mathfrak{n}_\beta})$ sequences (i.e., there is at least one j such that k_j differ) disagree in at least $2n^\beta+1$ locations. The set of such permutations (of form in Eq. 4.7 with distinct $(k_1,k_2,\cdots,k_{\mathfrak{n}_\beta})$ sequences) will be denoted $\widetilde{\Pi}^\beta$. Note that the size of $\widetilde{\Pi}^\beta$ is

$$\left|\widetilde{\Pi}^\beta\right| = (2n^\beta)^{\mathfrak{n}^\beta} \geq (2n^\beta)^{n/(2n^\beta+1)}$$

Next note that after time t_{cov}, $A_t \sim \text{ER}(n,1/2)$ independent of A_0 and hence $\text{tr}(A_0 P A_t P^T)$ is equal in distribution to $\text{tr}(A_0 A_t)$ for any permutation $P \in \Pi_n$. For a set of permutations

$\mathscr{Q} \in \Pi_n$, define the action of a permutation $P \circ \mathscr{Q}$ to be the set

$$P \circ \mathscr{Q} = \{Q \in \Pi_n \text{ s.t. } P^{-1}Q \in \mathscr{Q}\}.$$

Hence, if we define

$$\mathscr{E}_{\beta,P} = \left\{ \text{argmax}_{Q \in \Pi_n} \text{Tr}(A_0 Q A_t Q^T) \subset P \circ \left(\bigcup_{k=0}^{n^\beta} \Pi_{n,k} \right) \right\}$$

Then $\mathbb{P}(\mathscr{E}_{\beta,P} \,|\, t > t_{cov})$ is the same for all $P \in \Pi_n$. Moreover, the sets $\{\mathscr{E}_{\beta,P}\}_{P \in \widetilde{\Pi}^\beta \cup \{I_n\}}$ are disjoint, as the sets $P \circ \left(\bigcup_{k=0}^{n^\beta} \Pi_{n,k} \right)$ are disjoint for $P \in \widetilde{\Pi}^\beta \cup \{I_n\}$; if for $P_1, P_2 \in \widetilde{\Pi}^\beta \cup \{I_n\}$, there is a $Q \in P_1 \circ \left(\bigcup_{k=0}^{n^\beta} \Pi_{n,k} \right) \cap P_2 \circ \left(\bigcup_{k=0}^{n^\beta} \Pi_{n,k} \right)$, then there exists $Q_1, Q_2 \in \bigcup_{k=0}^{n^\beta} \Pi_{n,k}$ such that $P_1 = Q_1^{-1} Q_2 P_2$ and hence P_1 and P_2 could disagree in at most $2n^\beta$ locations yielding the contradiction. Therefore $\mathbb{P}(\mathscr{E}_{\beta,I_n} \,|\, t > t_{cov}) \leq (2n^\beta)^{-n/(2n^\beta+1)}$. The Theorem is then proven by considering

$$\mathbb{P}(\mathscr{E}_{\beta,I_n}) \leq \mathbb{P}(t_{cov} \geq t) + \mathbb{P}(\mathscr{E}_{\beta,I_n} \,|\, t > t_{cov})$$

and applying the above bounds.

4.5.4 Proof of Theorem 4.3

For each $i \in [K_n]$, let n_i denote the size of the i-the community. We will make use of the following Chernoff bound (here adapted from Theorem 3.2 of [25]): Let $X_i \sim \text{Bern}(p_i)$ be independent random variables with $X = \sum_i X_i$ and $\mathbb{E}(X) = \sum_i p_i$, then for any $t > 0$,

$$\mathbb{P}(|X - \mathbb{E}(X)| > t) \leq 2\exp\left\{ -\frac{t^2}{2\mathbb{E}(X) + 2t/3} \right\}$$

We apply this to $m_{ij} \sim \text{Bin}(n_i n_j, \Lambda_{i,j})$ defined to be the number of edges between community i and community j (with $m_{ii} \sim \text{Bin}(\binom{n_i}{2}, \Lambda_{i,i})$ defined analogously) to get that with probability at least 1-$K_n^2 n^{-3}$ (as $n_i n_j \Lambda_{i,j} = \omega(\log n)$)

$$\left| m_{ij} - n_i n_j \Lambda_{i,j} \right| \leq \sqrt{14 n_i n_j \Lambda_{i,j} \log n}; \tag{4.8}$$

$$\left| m_{ii} - \binom{n_i}{2} \Lambda_{i,i} \right| \leq \sqrt{14 \binom{n_i}{2} \Lambda_{i,i} \log n}. \tag{4.9}$$

For the remainder of the proof, we will condition on the events in Eq. 4.8–4.9 (call this event $\mathscr{A}_n$), so that between community 1 and community j, there are $m_{1,j} \in n_1 n_j \Lambda_{1,j} \pm \sqrt{14 n_1 n_j \Lambda_{1,j} \log n}$ edges. Recall that $\Lambda_{1,j} \leq b_3 \frac{n^{2a_2}}{K_n n_1 n_j}$ so that there exists a suitable constant $b_3' > 0$ with $m^* = \max_j m_{1,j} \leq b_3' \frac{n^{2a_2}}{K_n}$.

Consider the lamplighter walk on this graph. If in community 1, with probability $1/2$ the lamplighter will move to a uniformly random community other than community 1 in one step. Let us denote the number of steps before leaving community 1 via T_1. Note that

$$\mathbb{P}(T_1 > K_n m^*) = (1-p)^{K_n m^*} \leq \exp(-m^*),$$

and hence T_1 is with very high probability of lower order than our mixing time N defined below. As such, T_1 has little impact on the overall mixing time, and we can assume without loss of generality that the lamplighter starts outside of community 1.

Once the lamplighter first leaves community 1 (consider this time 0 if the lamplighter starts outside community 1), we define a renewal process as follows: We say that a renewal occurs if the lamplighter enters community 1, moves inside community 1 for at least one step and then exits community 1. The interarrival time has distribution equal to

$$\xi_n \overset{\text{dist.}}{=} \sum_{i=1}^{Z_1} (V_i + 1) + Z_2$$

where

i. $\{V_i\}$ are i.i.d. Geo$\left(\frac{1}{2(K_n-1)}\right)$random variables; here V_i represents the amount of time spent outside community 1 between the $i-1$st and ith visits to community 1;

ii. $Z_1 \sim \text{Geo}(1/2)$, independent of $\{V_i\}$, represents the number of visits to community 1 where no moves inside community 1 are taken;

iii. $Z_2 \sim \text{Geo}(1/2)$, independent of $\{V_i\}$ and Z_1, represents the number of moves inside community 1 the lamplighter takes in final visit to community 1 before renewal;

Here $\mathbb{E}(\xi_n) = 4(K_n-1)+2+2 = 4K_n$. Now the $f(n)$-th renewal time (assume $f(n) \in \mathbb{Z} > 0$), $S_{f(n)}$, satisfies $\mathbb{E}\left(S_{f(n)}\right) = f(n)4K_n$. Let

$$N = n^{a_3+a_2}\log^{3/2}(n)$$

and

$$f(n) = n^{2a_2}\log^{3/2}(n),$$

then Markov's inequality tells us (where $R(t)$ is the number of renewals by time t)

$$\mathbb{P}\left(R(N) < f(n)\right) = \mathbb{P}\left(S_{f(n)} > N\right) \leq \frac{4K_n n^{2a_2}\log^{3/2}(n)}{n^{a_3+a_2}\log^{3/2}(n)} = 4K_n n^{a_2-a_3} = o(1).$$

This implies that, with probability converging to 1, we have at least $f(n)$ renewals after N steps.

Between renewals the lamplighter must enter community 1 at least once. This entry is from a uniformly random community, with an edge between that community and community 1 being randomized. Let T_2 denote the number of renewals needed to randomize all edges into community 1. The distribution of T_2 is that of a non-uniform coupon-collector

(as the $m_{1,j}$'s are potentially different), which can be quite complex (see [45] for example); to circumvent, we will stochastically bound $T_2 \overset{st.}{\leq} T_3$ where T_3 is the uniform coupon collector distribution with $K_n m^*$ (where $m^* = \max_j m_{1,j}$) coupons to collect; the upper bound follows as T_3 has more coupons to collect than T_2, each with collection probability no greater than those in T_2. Well known coupon collector asymptotics then provide that (where C is a suitable constant that can change line-to-line)

$$\begin{aligned}
\mathbb{P}(T_2 > f(n)) &\leq \mathbb{P}(T_3 > f(n)) \\
&= \mathbb{P}\left(T_3 > \frac{f(n)}{(K_n m^*)\log(K_n m^*)}(K_n m^*)\log(K_n m^*)\right) \\
&\leq \exp\left(-C\frac{n^{2a_2}\log^{3/2} n}{n^{2a_2}\log n} + 1\right) \\
&\leq \exp\left(-C\log^{1/2} n\right)
\end{aligned}$$

This implies that after $f(n)$ renewals, we have with high probability randomized out all signals between community one and other communities.

Moreover, with each renewal, the lamplighter randomizes at least one edge (chosen uniformly among the $\binom{n_1}{2}$ possible edges in community 1). By Theorem 4.2, after $f(n)$ renewals within community 1, the lamplighter has performed a β-anonymization within community 1 for all constants $\beta \in (0,1)$. Collecting all results, once the edges connecting community 1 have been randomized, and the edges within community 1 have been randomized, community 1 has been effectively anonymized in the graph. Hence, letting

$$\mathscr{E}_{n,\beta} = \{\beta\text{-anonymization of community 1 after n steps}\},$$

we have for suitable constants $C_1, C_2 > 0$, and n sufficiently large

$$\begin{aligned}\mathbb{P}(\mathscr{E}^{\mathscr{C}}_{N,\beta}) &\leq \mathbb{P}(\mathscr{E}^{\mathscr{C}}_{N,\beta} | R(N) \geq f(n), \mathscr{A}_n) + \mathbb{P}(R(N) < f(n) | \mathscr{A}_n) + \mathbb{P}(\mathscr{A}_n^c) \\ &\leq C_1 \left(n^{-2a_1} + e^{-C_2 \log^{1/2} n} + K_n n^{a_2 - a_3} + K_n^2 n^{-3} \right)\end{aligned}$$

After $N = n^{a_3+a_2} \log^{3/2}(n) = o(n^{2a_3} \log n)$ steps, we have a β-anonymization of n^{a_1} nodes in the graph G with probability converging to 1. Note that the largest community is of order n^{a_3}, and that with high probability, at least $\Theta(n^{2a_3} \log n)$ steps are needed to randomize the edges within the large community; hence, after N steps, the lamplighter, with high probability, has not mixed.

4.5.5 Additional Experiments and Plots

Additional Standard Lamplighter on ER model plots

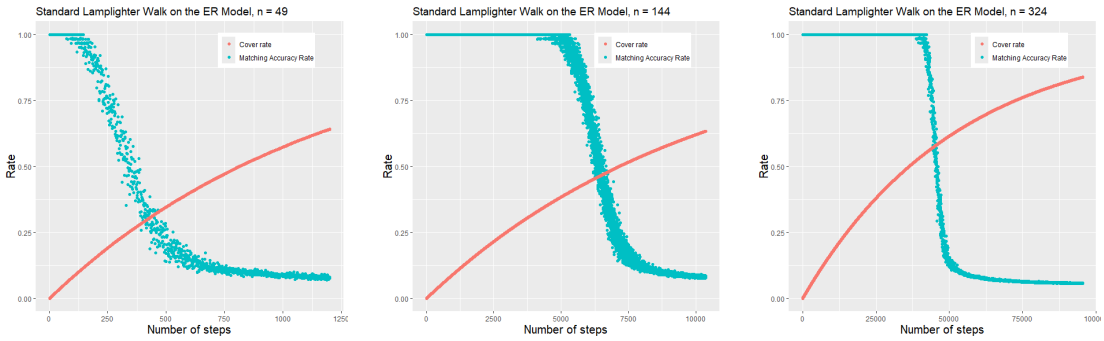

Figure 4.8: We plot matching correctness vs. number of steps for $n = 49$ (left), $n = 144$ (middle) and $n = 324$ (right), where the setup of the model and the parameters chosen are the same as in Section 4.3.1. For these plots, we further impose the cover rate of the edges vs number of steps.

Additional SBM Lamplighter on SBM model plots

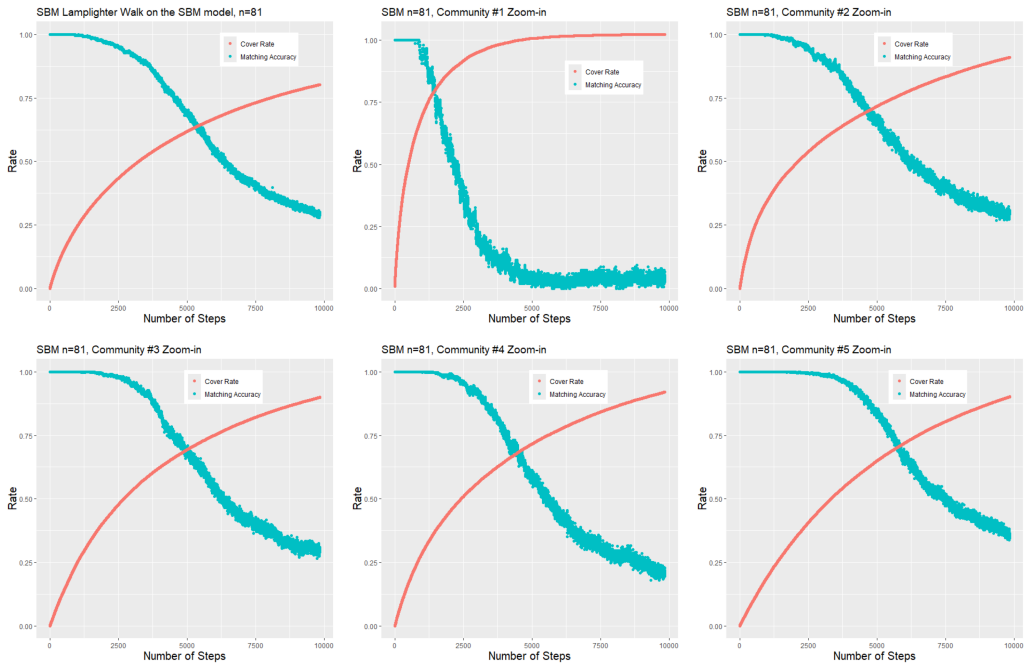

Figure 4.9: SBM Lamplighter Walk plot with $n = 81$, see Section 4.3.1 for the description of the experiment setup. Number of communities $= 5$

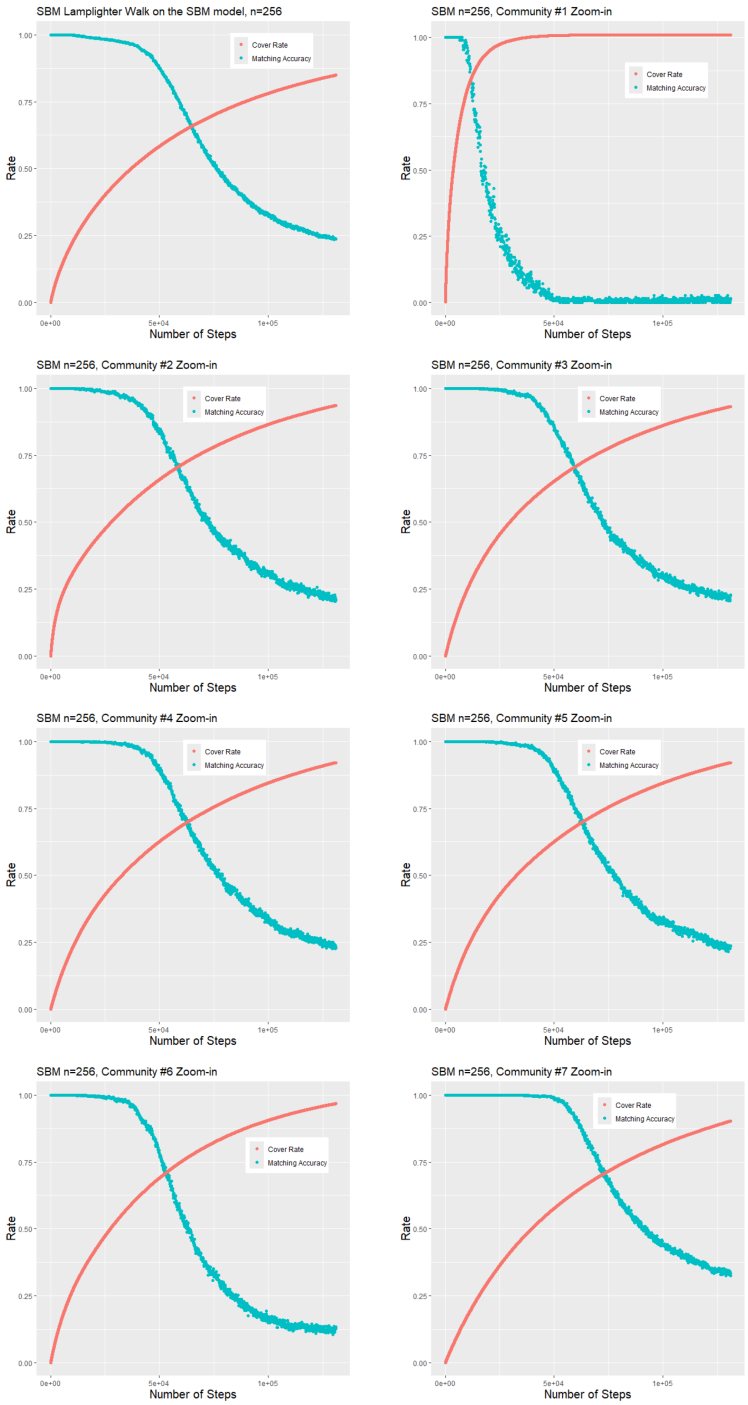

Figure 4.10: SBM Lamplighter Walk plot with $n = 256$, see Section 4.3.1 for the description of the experiment setup. Number of communities $= 7$

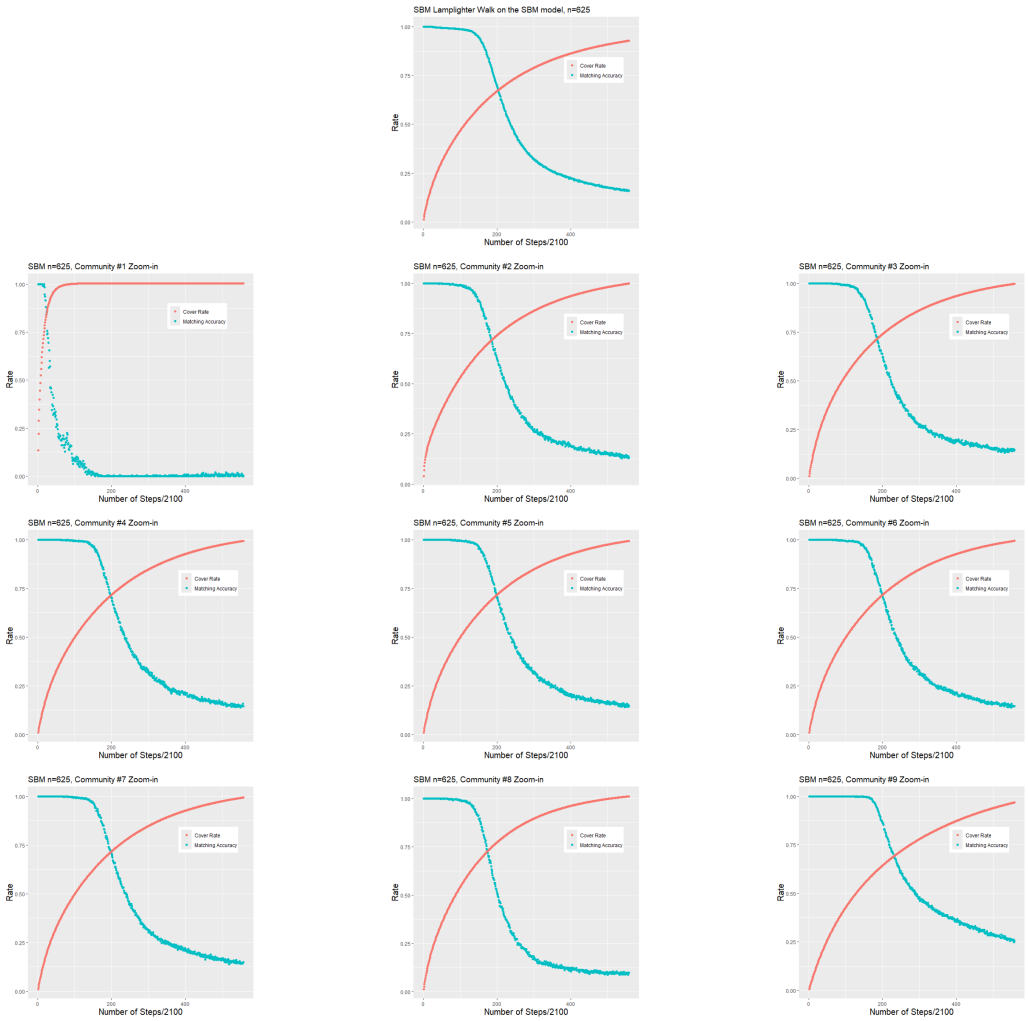

Figure 4.11: SBM Lamplighter Walk plot with $n = 625$, see Section 4.3.1 for the description of the experiment setup. Number of communities $= 9$

Bibliography

[1] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[2] M. A. Abdulrahim and M. Misra. A graph isomorphism algorithm for object recognition. *Pattern Analysis and Applications*, 1:189–201, 1998.

[3] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journ. of Mach. Learn. Res.*, 9:1981–2014, 2008.

[4] D. Aldous and J. A. Fill. Reversible markov chains and random walks on graphs, 2002.

[5] J. Arroyo, A. Athreya, J. Cape, G. Chen, Priebe C. E., and J. T. Vogelstein. Inference for multiple heterogeneous networks with a common invariant subspace. *Journ. of Mach. Learn. Res.*, 22:1–49, 2021.

[6] J. Arroyo, D. L. Sussman, C. E. Priebe, and V. Lyzinski. Maximum likelihood estimation and graph matching in errorfully observed networks. *Journ. of Computational and Graphical Statistics*, pages 1–13, 2021.

[7] A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, Y. Qin, and D. Sussman. Statistical inference on random dot product graphs: a survey. *Journ. of Mach. Learn. Res.*, 18:1–92, 2018.

[8] A. Athreya, C. E. Priebe, M. Tang, V. Lyzinski, D. J. Marchette, and D. L. Sussman. A limit theorem for scaled eigenvectors of random dot product graphs. *Sankhya A*, pages 1–18, 2013.

[9] Avanti Athreya, Zachary Lubberts, Youngser Park, and Carey E Priebe. Euclidean mirrors and dynamics in network time series, 2024.

[10] L. Babai. Graph isomorphism in quasipolynomial time. In *Proc. of the 48th annual ACM symp. on Theory of Computing*, pages 684–697, 2016.

[11] Sven Banisch and Sven Banisch. The voter model with homogeneous mixing. *Markov Chain Aggregation for Agent-Based Models*, pages 57–82, 2016.

[12] A.-L. Barabási. *Network Science*. Cambridge University Press, 2016.

[13] B. Barak, C. Chou, Z. Lei, T. Schramm, and Y. Sheng. (nearly) efficient algorithms for the graph matching problem on correlated random graphs. *Adv. in Neural Infor. Proc. Sys.*, 32:9190–9198, 2019.

[14] D. Barrett and B. Dherin. Implicit gradient regularization. In *Int. Conf. on Learning Representations*, 2021.

[15] Dave Bayer and Persi Diaconis. Trailing the dovetail shuffle to its lair. *The Annals of Applied Probability*, pages 294–313, 1992.

[16] S. Bhamidi, R. Van der Hofstad, and G. Hooghiemstra. First passage percolation on the Erdős–Rényi random graph. *Combinatorics, Probability and Computing*, 20(5):683–707, 2011.

[17] P. J. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. of the National Academy of Sciences of the United States of America*, 106:21068–73, 2009.

[18] P. J. Bickel and A. Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proc. of the National Academy of Sciences*, 106(50):21068–21073, 2009.

[19] P. J. Bickel, A. Chen, and E. Levina. The method of moments and degree distributions for network models. *The Annals of Statistics*, 39(5):2280–2301, 2011.

[20] J Bijsterbosch and A Volgenant. Solving the rectangular assignment problem and applications. *Ann. Oper. Res.*, 181:443–462, 2010.

[21] B. Bollobás. *Random graphs*. Springer, 1998.

[22] B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures and Algorithms*, 31:3–122, 2007.

[23] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. Vishwanathan, A. J. Smola, and H. P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 2005.

[24] Joshua Cape, Minh Tang, and Carey E Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. *The Annals of Statistics*, 47, No. 5, 2405–2439, 2019.

[25] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1):79–127, 2006.

[26] J. Chung, E. Bridgeford, J. Arroyo, B. D. Pedigo, A. Saad-Eldin, V. Gopalakrishnan, L. Xiang, C. E. Priebe, and J. T. Vogelstein. Statistical connectomics. *Annual Review of Statistics and Its Application*, 8:463–492, 2021.

[27] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[28] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *Int. journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004.

[29] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience; 2nd ed., 2006.

[30] D. Cullina and N. Kiyavash. Improved achievability and converse bounds for erdos-renyi graph matching. In *ACM SIGMETRICS Performance Evaluation Review*, volume 44 (1), pages 63–72. ACM, 2016.

[31] D. Cullina and N. Kiyavash. Exact alignment recovery for correlated erdos renyi graphs. *arXiv preprint arXiv:1711.06783*, 2017.

[32] Riet De Smet and Kathleen Marchal. Advantages and limitations of current network inference methods. *Nature Reviews Microbiology*, 8(10):717–729, 2010.

[33] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[34] B. Draves and D. L. Sussman. Bias-variance tradeoffs in joint spectral embeddings. *arXiv preprint arXiv:2005.02511*, 2020.

[35] Xinjie Du and Minh Tang. Hypothesis testing for equality of latent positions in random graphs. *Bernoulli*, 29(4):3221–3254, 2023.

[36] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Adv. in Neural Infor. Proc. Sys.*, volume 28. Curran Associates, Inc., 2015.

[37] C. L. Ebsch, J. A. Cottam, N. C. Heller, R. D. Deshmukh, and G. Chin. Using graph edit distance for noisy subgraph matching of semantic property graphs. In *2020 IEEE international conference on big data (big data)*, pages 2520–2525. IEEE, 2020.

[38] P. Erdős and A. Rényi. Asymmetric graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 14(3–4):295–315, 1963.

[39] Pál Erdős and Alfréd Rényi. On a classical problem of probability theory. *A Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 6(1-2):215–220, 1961.

[40] Z. Fan, C. Mao, Y. Wu, and J. Xu. Spectral graph matching and regularized quadratic relaxations: Algorithm and theory. In *Int. Conf. on Machine Learning*, pages 2985–2995. PMLR, 2020.

[41] F. Fang, D. L. Sussman, and V. Lyzinski. Tractable graph matching via soft seeding. *arXiv preprint arXiv:1807.09299*, 2018.

[42] D. E. Fishkind, L. Meng, A. Sun, C. E. Priebe, and V. Lyzinski. Alignment strength and correlation for graphs. *Pattern Recognition Letters*, 125:295–302, 2019.

[43] D. E. Fishkind, F. Parker, H. Sawczuk, L. Meng, E. Bridgeford, A. Athreya, C. Priebe, and V. Lyzinski. The phantom alignment strength conjecture: practical use of graph matching alignment strength to indicate a meaningful graph match. *Applied Network Science*, 6(1):1–27, 2021.

[44] D.E. Fishkind, S. Adali, H.G. Patsolic, L. Meng, D. Singh, V. Lyzinski, and C.E. Priebe. Seeded graph matching. *Pattern Recognition*, 87:203 – 215, 2019.

[45] Philippe Flajolet, Daniele Gardy, and Loÿs Thimonier. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Applied Mathematics*, 39(3):207–229, 1992.

[46] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

[47] A. Frieze and M. Karoński. *Introduction to random graphs*. Cambridge University Press, 2016.

[48] C. E. Ginestet, J. Li, P. Balachandran, S. Rosenberg, and E. D. Kolaczyk. Hypothesis testing for network data in functional neuroimaging. *Ann. Appl. Stat.*, pages 725–750, 2017.

[49] Cedric E Ginestet, Jun Li, Prakash Balachandran, Steven Rosenberg, and Eric D Kolaczyk. Hypothesis testing for network data in functional neuroimaging. *The Annals of Applied Statistics*, pages 725–750, 2017.

[50] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010.

[51] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journ. of the American Statistical Association*, 97(460):1090–1098, 2002.

[52] A. J. Hoffman and H. W. Wielandt. The variation of the spectrum of a normal matrix. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 118–120. World Scientific, 2003.

[53] P. W. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[54] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[55] Saber Jafarizadeh. Fastest mixing reversible markov chain: Clique lifted graphs and subgraphs. *IEEE Transactions on Signal and Information Processing over Networks*, 6:88–104, 2020.

[56] B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83, 2011.

[57] KDD Proc. of the 17th ACM SIGKDD international Conf. on Knowledge discovery and data mining. *On the privacy of anonymized networks*, 2011.

[58] Gregory Kiar, Eric W Bridgeford, William R Gray Roncai, Vikram Chandrashekhar, Disa Mhembere, Sephira Ryman, Xi-Nian Zuo, Daniel S Margulies, R Cameron Craddock, Carey E Priebe, et al. A high-throughput pipeline identifies robust connectomes but troublesome variability. *bioRxiv*, page 188706, 2018.

[59] E. D. Kolaczyk and G. Csárdi. *Statistical analysis of network data with R*, volume 65. Springer, 2014.

[60] E. D. Kolaczyk, L. Lin, S. Rosenberg, J. Walters, and J. Xu. Averages of unlabeled networks: Geometric characterization and asymptotic behavior. *Ann. Stat.*, 48(1):514–538, 2020.

[61] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[62] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[63] Clement Lee and Darren J Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50, 2019.

[64] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

[65] David A Levin, Malwina J Luczak, and Yuval Peres. Glauber dynamics for the mean-field ising model: cut-off, critical power law, and metastability. *Probability Theory and Related Fields*, 146:223–265, 2010.

[66] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[67] Keith Levin, Avanti Athreya, Minh Tang, Vince Lyzinski, and Carey E Priebe. A central limit theorem for an omnibus embedding of multiple random dot product graphs. In *2017 IEEE international conference on data mining workshops (ICDMW)*, pages 964–967. IEEE, 2017.

[68] Guanpeng Li, Siva Kumar Sastry Hari, Michael Sullivan, Timothy Tsai, Karthik Pattabiraman, Joel Emer, and Stephen W Keckler. Understanding error propagation in deep learning neural network (dnn) accelerators and applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12, 2017.

[69] Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Koen Van den Berge, Purnamrita Sarkar, Peter J Bickel, and Elizaveta Levina. Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*, 117(538):951–968, 2022.

[70] Zhirui Li, Jesús Arroyo, Konstantinos Pantazis, and Vince Lyzinski. Clustered graph matching for label recovery and graph classification. *IEEE Trans. Netw. Sci. and Eng.*, 2023.

[71] Zhirui Li, Ben Johnson, Daniel L Sussman, Carey E Priebe, and Vince Lyzinski. Gotta match'em all: Solution diversification in graph matching matched filters. *IEEE Transactions on Signal and Information Processing over Networks*, 2024.

[72] László Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4, 1993.

[73] V. Lyzinski. Information recovery in shuffled graphs via graph matching. *IEEE Trans. Info. Theory*, 64(5):3254–3273, 2018.

[74] V. Lyzinski, D. E. Fishkind, M. Fiori, J. T. Vogelstein, C. E. Priebe, and G. Sapiro. Relax at your own risk. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 60–73, 2016.

[75] V. Lyzinski and D. L. Sussman. Matchability of heterogeneous networks pairs. *Information and Inference: A Journ. of the IMA*, 9(4):749–783, 2020.

[76] Vince Lyzinski, Sancar Adali, Joshua T Vogelstein, Youngser Park, and Carey E Priebe. Seeded graph matching via joint optimization of fidelity and commensurability. *stat*, 1050:16, 2014.

[77] Vince Lyzinski, Minh Tang, Avanti Athreya, Youngser Park, and Carey E Priebe. Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions on Network Science and Engineering*, 4(1):13–26, 2016.

[78] Cheng Mao, Yihong Wu, Jiaming Xu, and Sophie H Yu. Random graph matching at otter's threshold via counting chandeliers. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1345–1356, 2023.

[79] J. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, volume 2012, pages 548–56, 2012.

[80] C. McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[81] J. D. Moorman, Q. Chen, T. K. Tu, Z. M. Boyd, and A. L. Bertozzi. Filtering methods for subgraph matching on multiplex networks. In *2018 IEEE Int. Conf. on Big Data (Big Data)*, pages 3980–3985, Dec 2018.

[82] J. D. Moorman, T. Tu, Q. Chen, X. He, and A. Bertozzi. Subgraph matching on multiplex networks. *IEEE Trans. Netw. Sci. and Eng.*, 2021.

[83] Ben Morris, Weiyang Ning, and Yuval Peres. Mixing time of the card-cyclic-to-random shuffle. *The Annals of Applied Probability*, 24(5):1835–1849, 2014.

[84] E. Mossel and J. Xu. Seeded graph matching via large neighborhood statistics. *Random Structures & Algorithms*, 57(3):570–611, 2020.

[85] D. R. Musser. Introspective sorting and selection algorithms. *Software: Practice and Experience*, 27(8):983–993, 1997.

[86] M. E. J. Newman. Spread of epidemic disease on networks. *Phys. Rev. E*, 66:016128, 2002.

[87] A. M. Nielsen and D. Witten. The multiple random dot product graph model. *arXiv preprint arXiv:1811.12172*, 2018.

[88] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *Proc. of The 33rd Int. Conf. on Machine Learning*, volume 48 of *Proc. of Machine Learning Research*, pages 2014–2023. PMLR, 20–22 Jun 2016.

[89] G. Nikolentzos, P. Meladianos, and M. Vazirgiannis. Matching node embeddings for graph similarity. In *AAAI*, 2017.

[90] Majid Noroozi and Marianna Pensky. The hierarchy of block models. *Sankhya A*, 84(1):64–107, 2022.

[91] E. Onaran, S. Garg, and E. Erkip. Optimal de-anonymization in random graphs with community structure. *arXiv preprint arXiv:1602.01409*, 2016.

[92] K. Pantazis, D. L. Sussman, Y. Park, Z. Li, C. E. Priebe, and V. Lyzinski. Multiplex graph matching matched filters. *Applied Network Science*, 7(1):1–35, 2022.

[93] T. P. Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X*, 4, 2014.

[94] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar, and S. E. Petersen. Functional network organization of the human brain. *Neuron*, 72(4):665–678, 2011.

[95] C. E. Priebe, D. L. Sussman, M. Tang, and J. T. Vogelstein. Statistical inference on errorfully observed graphs. *Journ. of Computational and Graphical Statistics*, 24(4):930–953, 2015.

[96] S. Purohit, P. Mackey, W. Smith, M. Dunning, M. J. Orren, T. M. Langlie-Miletich, R. D. Deshmukh, A. Bohra, T. J. Martin, and D. J. Aimone. Transactional knowledge graph generation to model adversarial activities. In *2021 IEEE Int. Conf. on Big Data (Big Data)*, pages 2662–2671. IEEE, 2021.

[97] M. Racz and A. Sridhar. Correlated stochastic block models: Exact graph matching with applications to recovering communities. *Adv. in Neural Infor. Proc. Sys.*, 34, 2021.

[98] L. Ramshaw and R. E. Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. *HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1*, 20, 2012.

[99] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journ. of the American Statistical association*, 66(336):846–850, 1971.

[100] Jesús D Arroyo Relión, Daniel Kessler, Elizaveta Levina, and Stephan F Taylor. Network classification with applications to brain connectomics. *The annals of applied statistics*, 13(3):1648, 2019.

[101] K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics*, 39:1878–1915, 2011.

[102] N. Ross. Fundamentals of stein's method. *Probability Surveys*, 8:210–293, 2011.

[103] B. A. Sabarish, R. Karthi, and K. T. Gireesh. Graph similarity-based hierarchical clustering of trajectory data. *Procedia Computer Science*, 171:32–41, 2020. Third Int. Conf. on Computing and Network Communications (CoCoNet'19).

[104] Ayushi Saxena and Vince Lyzinski. Lost in the shuffle: Testing power in the presence of errorful network vertex labels. *Computational Statistics & Data Analysis*, 204:108091, 2025.

[105] C Seshadhri, A. Sharma, A. Stolman, and A. Goel. The impossibility of low-rank representations for triangle-rich complex networks. *Proc. of the National Academy of Sciences*, 117(11):5631–5637, 2020.

[106] N. Shervashidze, P. Schweitzer, E.J.V. Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journ. of Mach. Learn. Res.*, 12(77):2539–2561, 2011.

[107] C. Solnon. Experimental evaluation of subgraph isomorphism solvers. In *Graph-Based Representations in Pattern Recognition: 12th IAPR-TC-15 Int. Workshop, GbRPR 2019, Tours, France, June 19–21, 2019, Proceedings 12*, pages 1–13. Springer, 2019.

[108] Vishal Sood, Sidney Redner, and Dani Ben-Avraham. First-passage properties of the erdős–renyi random graph. *Journal of Physics A: Mathematical and General*, 38(1):109, 2004.

[109] Ulrich Stelzl, Uwe Worm, Maciej Lalowski, Christian Haenig, Felix H Brembeck, Heike Goehler, Martin Stroedicke, Martina Zenkner, Anke Schoenherr, Susanne Koeppen, et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122(6):957–968, 2005.

[110] S. Sun and Q. Luo. In-memory subgraph matching: An in-depth study. In *Proceedings of the 2020 ACM SIGMOD Int. Conf. on Management of Data*, pages 1083–1098, 2020.

[111] D. L. Sussman, V. Lyzinski, Y. Park, and C. E. Priebe. Matched filters for noisy induced subgraph detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(11):2887–2900, 2019.

[112] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journ. of the American Statistical Association*, 107(499):1119–1128, 2012.

[113] Daniel L Sussman, Minh Tang, Donniell E Fishkind, and Carey E Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128, 2012.

[114] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, Y. Park, and C. E. Priebe. A semiparametric two-sample hypothesis testing problem for random graphs. *Journ. of Computational and Graphical Statistics*, 26(2):344–354, 2017.

[115] M. Tang, A. Athreya, D. L. Sussman, V. Lyzinski, and C. E. Priebe. A semiparametric two-sample hypothesis testing for random dot product graphs. arXiv preprint. `http://arxiv.org/abs/1403.7249`, 2014.

[116] M. Tang and C. E. Priebe. Limit theorems for eigenvectors of the normalized laplacian for random graphs. *The Annals of Statistics*, 46(5):2360–2415, 2018.

[117] M. Tang, D. L. Sussman, and C. E. Priebe. Universally consistent vertex classification for latent positions graphs. *The Annals of Statistics*, 41(3):1406–1430, 2013.

[118] Minh Tang, Joshua Cape, and Carey E Priebe. Asymptotically efficient estimators for stochastic blockmodels: The naive mle, the rank-constrained mle, and the spectral estimator. *Bernoulli*, 28(2):1049–1073, 2022.

[119] R. Tang, M. Ketcha, A. Badea, E. D. Calabrese, D. S. Margulies, J. T. Vogelstein, C. E. Priebe, and D. L. Sussman. Connectome smoothing via low-rank approximations. *IEEE Trans. Med. Imaging*, 38(6):1446–1456, 2018.

[120] C. J. Thies, V. H. Metzler, T. M. Lehmann, and T. Aach. Formal extraction of biomedical objects by subgraph matching in attributed hierarchical region adjacency graphs. In *Medical Imaging 2004: Image Processing*, volume 5370, pages 1498–1508. SPIE, 2004.

[121] T. K. Tu, J. D. Moorman, D. Yang, Q. Chen, and A. L. Bertozzi. Inexact attributed subgraph matching. In *2020 IEEE Int. Conf. on Big Data (Big Data)*, pages 2575–2582. IEEE, 2020.

[122] Julian R Ullmann. An algorithm for subgraph isomorphism. *Journ. of the ACM (JACM)*, 23(1):31–42, 1976.

[123] F. Vaca-Ramírez and T. P. Peixoto. Systematic assessment of the quality of fit of the stochastic block model for empirical networks. *arXiv preprint arXiv:2201.01658*, 2022.

[124] J. T. Vogelstein and C. E. Priebe. Shuffled graph classification: Theory and connectome applications. *Journ. of Classification*, 32(1):3–20, 2015.

[125] J. T. Vogelstein, W. G. Roncal, R. J. Vogelstein, and C. E. Priebe. Graph classification using signal-subgraphs: Applications in statistical connectomics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1539–1551, 2013.

[126] J.T. Vogelstein, J.M. Conroy, V. Lyzinski, L.J. Podrazik, S.G. Kratzer, E.T. Harley, D.E. Fishkind, R.J. Vogelstein, and C.E. Priebe. Fast Approximate Quadratic Programming for Graph Matching. *PLoS ONE*, 10(04), 2014.

[127] Lu Wang, Feng Vankee Lin, Martin Cole, and Zhengwu Zhang. Learning clique subgraphs in structural brain network classification with application to crystallized cognition. *Neuroimage*, 225:117493, 2021.

[128] S. Wang, J. Arroyo, J. T. Vogelstein, and C. E. Priebe. Joint embedding of graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(4):1324–1336, 2019.

[129] David Bruce Wilson. Mixing time of the rudvalis shuffle. *Electronic Communications in Probability*, 8:77–85, 2003.

[130] P. J. Wolfe and S. C. Olhede. Nonparametric graphon estimation. ArXiv preprint at `http://arxiv.org/abs/1309.5936`, 2013.

[131] M. Wu, X. Li, C. Kwoh, and S. Ng. A core-attachment based method to detect protein complexes in ppi networks. *BMC Bioinf.*, 10(1), 2009.

[132] Y. Wu, J. Xu, and S. H. Yu. Settling the sharp reconstruction thresholds of random graph matching. *IEEE Trans. Info. Theory*, 68(8), 2022.

[133] D. Yang, Y. Ge, T. Nguyen, D. Molitor, J. D. Moorman, and A. L. Bertozzi. Structural equivalence in subgraph matching. *IEEE Trans. Netw. Sci. and Eng.*, 2023.

[134] Lyudmila Yartseva and Matthias Grossglauser. On the performance of percolation graph matching. In *Proceedings of the first ACM conference on Online social networks*, pages 119–130, 2013.

[135] S. Young and E. Scheinerman. Random dot product graph models for social networks. In *Proc. of the 5th international Conf. on algorithms and models for the web-graph*, pages 138–149, 2007.

[136] S. Zhang, D. Zhou, M. Y. Yildirim, S. Alcorn, J. He, H. Davulcu, and H. Tong. Hidden: Hierarchical dense subgraph detection with application to financial fraud detection. In *SDM*, 2017.

[137] Y. Zhao, H. Zhang, and X. Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *Int. Conf. on Machine Learning*, pages 26982–26992. PMLR, 2022.

[138] Y. Zhou and H.-G. Müller. Dynamic network regression. *arXiv preprint arXiv:2109.02981*, 2021.

[139] X. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos, et al. An open science resource for establishing reliability and reproducibility in functional connectomics. *Scientific data*, 1(1):1–13, 2014.