

RESEARCH

Open Access



Multiplex graph matching matched filters

Konstantinos Pantazis^{1*} , Daniel L. Sussman², Youngser Park³, Zhirui Li¹, Carey E. Priebe⁴ and Vince Lyzinski¹

*Correspondence:
kpantazi@umd.edu

¹ Department
of Mathematics, University
of Maryland, College Park,
College Park, USA
Full list of author information
is available at the end of the
article

Abstract

We consider the problem of detecting a noisy induced multiplex template network in a larger multiplex background network. Our approach, which extends the graph matching matched filter framework of Sussman et al. (IEEE Trans Pattern Anal Mach Intell 42(11):2887–2900, 2019) to the multiplex setting, utilizes a multiplex analogue of the classical graph matching problem to use the template as a matched filter for efficiently searching the background for candidate template matches. The effectiveness of our approach is demonstrated both theoretically and empirically, with particular attention paid to the potential benefits of considering multiple channels in the multiplex framework.

Keywords: Multiplex graphs, Graph matching, Correlation network models, Matched filters

Introduction and background

Multilayer networks have proven to be useful models for capturing complex relational data where multiple types of relations are potentially present between vertices in the network (Boccaletti et al. 2014; Kivelä et al. 2014). Classical, single layer (or monoplex) networks, have proven popular for representing complex interactions (i.e., edges) amongst a collection of objects of interest (vertices). In the multilayer setting, there are multiple vertex layers and edges within a layer can encode different relationship/interaction types. Multilayer graphs are often distinguished by the structure across layers; in the present work we consider multiplex networks where inter-layer relations can only exist amongst vertices representing the same object across the layers. For example, in connectomes (i.e., brain networks) different edge layers/modalities can represent different synapse types between neurons (e.g., chemical versus electrical gap junction synapses amongst the neurons in the *C. elegans* connectome White et al. 1986); in social networks different edge modalities can capture relationships/friendships in different social network platforms (Goga et al. 2015); in scholarly networks, different edge modalities can capture co-authorship across multiple classification categories (Ng et al. 2011). Inasmuch as there is complimentary signal across the layers in the multilayer network (which is the case in many applications, see for example Mucha et al. 2010; Kivelä et al. 2014; Chen et al. 2016), leveraging the signal across the different layers of the network can lead to better, more robust performance than working within any single network modality. In this paper, we address

the question of how we can practically (and theoretically) leverage the signal across multiple network layers in the inference setting of noisy subgraph detection.

Formally, the noisy subgraph detection problem can be stated as follows. For integer $n > 0$, let \mathcal{G}_n be the space of all n -vertex labeled networks; note that unless otherwise specified, all graphs considered will be unweighted. Given a dissimilarity $d : \mathcal{G}_m \times \mathcal{G}_m \mapsto \mathbb{R}$ (so that d is non-negative, $d(g, g) = 0$ for all $g \in \mathcal{G}_m$, and small (resp., large) values of d are indicative of highly similar (resp., dissimilar) networks), a template subgraph $g \in \mathcal{G}_m$, and a background graph $h \in \mathcal{G}_n$ with $n > m$, the goal is to find the subgraphs $g' \in \mathcal{G}_m$ of h minimizing $d(g, g')$. Common dissimilarities d considered in the literature include variants of graph edit distance (Ebsch 2020) and the graph matching distance (Chan and Cheney 2020; Sussman et al. 2019) (see “The graph matching distance” section for detail). The $\min_{g' \subset h} d(g, g') = 0$ setting (where $g' \subset h$ denotes g' is a subgraph of h) is closely related to the subgraph isomorphism problem—given a template g , determine if an isomorphic copy of g exists in the larger network h and find the isomorphic copy (or copies) if it exists. This NP-complete problem has been the subject of voluminous research, with notable results in the setting where g is small or structured. Common approaches include those based on efficient tree search (Ullmann 1976), color coding (Alon et al. 1995, 2008), graph homomorphisms (Fomin et al. 2012), rule-based/filter-based matchings (Cordella et al. 2004; Moorman et al. 2018), constraint satisfaction (Larrosa and Valiente 2002; Zampelli et al. 2010), among others; for a survey of the literature circa 2012, see Lee et al. (2012). The potential to have $\min_{g' \subset h} d(g, g') > 0$ (this is what is meant by “noisy”; note this is also referred to as inexact matching in the literature) for a given template separates the noisy subgraph detection task from the classical subgraph isomorphism problem, as this accounts for the reality that relatively large, complex subgraph templates may only errorfully occur in the larger background network. These errors may be due to missing edges/vertices in the template or background, and arise in a variety of real data settings (Priebe et al. 2015). When g is large and complex and/or $\min_{g' \subset h} d(g, g') > 0$, there are relatively fewer methods in the literature for attacking this problem; see, for example, Tu et al. (2020), Sussman et al. (2019), Du et al. (2017), DePiero and Krout (2003), Lladós et al. (2001). Moreover, for many d , in the setting of complex templates g there is often a single element (up to vertex reordering) in $\arg \min_{g' \subset h} d(g, g')$ and the noisy subgraph detection problem reduces to detecting this unique minimizer.

The inference task we consider in this manuscript is the problem of detecting (possibly multiple copies of) a noisy multiplex template network in a multiplex background network. Letting \mathcal{M}_n^c be the set of all c -layer, n -vertex multiplex networks (see Definition 1), we can define the problem formally as follows. Given a dissimilarity $d : \mathcal{M}_m^c \times \mathcal{M}_m^c \mapsto \mathbb{R}$ (so that d is non-negative, $d(\mathbf{g}, \mathbf{g}) = 0$ for all $\mathbf{g} \in \mathcal{M}_m^c$, and small (resp., large) values of d are indicative of highly similar (resp., dissimilar) networks), a template subgraph $\mathbf{g} \in \mathcal{M}_m^c$, and a background graph $\mathbf{h} \in \mathcal{M}_n^c$ with $n > m$, the goal is to find the subgraphs $\mathbf{g}' \in \mathcal{M}_m^c$ of \mathbf{h} minimizing $d(\mathbf{g}, \mathbf{g}')$. With the rise of network modeling of complex, multilayer data, there has been a recent emphasis on multiplex inference methods. This is, indeed, the case with the noisy multiplex subgraph

detection problem, as there have been numerous recent papers dedicated to solving this problem; see, for example Yang and Liu (2016), Takes et al. (2017), Kopylov and Xu (2019), Kopylov et al. (2020), Moorman et al. (2021, 2018). This multilayer subgraph detection problem presents a host of additional challenges versus its single layer counterpart, including how to adapt monoplex dissimilarities to the multilayer setting (see Kivelä and Porter 2017 for the challenges in defining the general multilayer graph isomorphism), and how to appropriately leverage the signal across the multiple layers rather than more streamlined approaches that collapse the network to a single layer.

Our approach to approximately solving the multiplex noisy subgraph problem adapts the graph matching matched filter (GMMF) method proposed in Sussman et al. (2019) to the multiplex setting. The basic idea of the GMMF is to use the template graph \mathbf{g} to scan the background for similarly (according to d) structured subgraphs in \mathbf{h} ; this is similar in spirit to the matched filters for pattern detection in image analysis (e.g., Chaudhuri et al. 1989) and the convolutional filters in image processing and convolutional neural networks (Krizhevsky et al. 2012; O'Shea and Nash 2015). In our Multiplex GMMF (which we denote \mathbf{M} -GMMF), the scanning mechanism determining how the template is used to search the background is a multiplex adaptation of the Frank-Wolfe (Frank and Wolfe 1956) based \mathbf{FAQ} algorithm of Vogelstein et al. (2014) (see “[Multiplex \$\mathbf{FAQ}\$](#) ” section for more detail). For the details of \mathbf{M} -GMMF, see “[Multiplex graph matching matched filters](#)” section. In the process of developing the \mathbf{M} -GMMF, we introduce a multiplex graph matching problem (see “[The multiplex graph matching distance](#)” section) that allows us to appropriately define a dissimilarity d that seeks to minimize the number of disagreements between \mathbf{g} and \mathbf{g}' (i.e., optimal induced subgraph match) or maximize the number of common edges across \mathbf{g} and \mathbf{g}' (i.e., optimal subgraph match).

In addition to the algorithmic development above, we further posit a statistical framework for embedding a noisy copy of \mathbf{g} into \mathbf{h} that allows us to study an analogue of multiplex graph matchability; see “[Multiplex template matchability](#)” section. In this model, we demonstrate the theoretical appropriateness of multiplex matched filters for detecting the noisy induced subgraph as well as the benefit of considering multiple channels for improved graph de-anonymization. We further demonstrate these theoretical results in a number of simulation and real data settings. In “[Experiments](#)” section, we apply our multiplex matched filter approach in detecting a hidden template in a multiplex social media network and we compare \mathbf{M} -GMMF to other multiplex graph matching algorithms on a template discovery task in a large semantic property graph.

Notation: The following notation will be used throughout. For an integer $n > 0$, we will define $[n] := \{1, 2, \dots, n\}$, let $\mathbf{1}_n \in \mathbb{R}^n$ (resp. $I_n \in \mathbb{R}^{n \times n}$) be the vector (resp. diagonal matrix) with all entries identically set to 1, let J_n be the $n \times n$ matrix with all diagonal entries identically set to 0 and all off-diagonal entries set to 1. Let $\mathbf{0}_{k,\ell}$ (resp., $\mathbf{0}_k$) be the $k \times \ell$ (resp., $k \times k$) matrix with all entries identically set to 0. The set of $n \times n$ permutation matrices is denoted by Π_n . For a graph $g \in \mathcal{G}_n$, we let $V(g)$ denote the vertex set of g and $E(g)$ denote the edge set of g . A graph g' is a subgraph of g (denoted $g' \subset g$) if $V(g') \subset V(g)$ and $E(g') \subset E(g)$.

The graph matching distance

Both the single and multilayer noisy subgraph detection problems defined above rely heavily on a suitable definition of graph dissimilarity d to measure the goodness of the fit of the recovered subgraph match. A common dissimilarity used in the literature is the *graph matching distance* d_{GM} defined as follows: Let $g, h \in \mathcal{G}_n$ (recalling \mathcal{G}_n is the space of n -vertex, labelled graphs; here these are assumed to be unweighted) and let A be the adjacency matrix for g , and B the adjacency matrix for h ; we then define

$$d_{\text{GM}}(g, h) = \min_{P \in \Pi_n} \|AP - PB\|_F = \min_{P \in \Pi_n} \sqrt{\|A\|_F^2 - 2 \text{trace}(APB^T P^T) + \|B\|_F^2},$$

where Π_n is the set of $n \times n$ permutation matrices. Note that d_{GM} is a pseudo-distance (i.e., $d_{\text{GM}}(g, g) = 0$ for all $g \in \mathcal{G}_n$, d_{GM} is symmetric, and satisfies the triangle inequality) as it is clear that $d_{\text{GM}}(g, h) = 0$ can hold for $g \neq h$. See Bento and Ioannidis (2019) for a study of a number of related graph matching (pseudo) distances.

There is an immense literature devoted to solving the related graph matching problem: Given graphs $g, h \in \mathcal{G}_n$ (with respective adjacency matrices A and B), find $P \in \Pi_n$ such that $d_{\text{GM}}(g, h) = \|AP - PB\|_F$; see Conte et al. (2004), Foggia et al. (2014), Emmert-Streib et al. (2016), Yan et al. (2016) for a thorough review of the current graph matching literature. Briefly, the graph matching literature is composed of two (intertwined) main thrusts: algorithmic development, with popular methods including optimization/relaxation-based approaches (e.g., Zaslavskiy et al. 2009; Umeyama 1988; Vogelstein et al. 2014; Klau 2009; Zhang and Tong 2016), tree-search (e.g., Pinheiro et al. 2016), deep learning-based approaches (e.g., Zanfir and Sminchisescu 2018), representation-learning approaches (e.g., Heimann et al. 2018; Du et al. 2020), spectral approaches (e.g., Feizi et al. 2015; Fan et al. 2020; Egozi et al. 2012; Singh et al. 2008), and structured factorization-based approaches (e.g., Zhou and De la Torre 2012; Zhang et al. 2019), to name but a few. The second thrust of the graph matching literature is focused on graph de-anonymization/graph matchability. Stated succinctly, the problem of graph matchability is as follows: given a latent alignment P between two networks g and h (with respective adjacency matrices A and B), is it the case that $d_{\text{GM}}(g, h) = \|AP - PB\|_F$; i.e., will the optimal “graph matching” permutation recover the latent alignment? The latent alignment between g and h is often achieved by correlating the networks g and h edge-wise (so that h can be thought of as a noisy version of g), and the matchability literature is often focused on deriving sharp phase-transitions for when $d_{\text{GM}}(g, h) \neq \|AP - PB\|_F$ in terms of decaying correlation across networks; see, for example, Ding et al. (2018), Kazemi et al. (2015a, b), Pedarsani and Grossglauser (2011), Onaran et al. (2016), Lyzinski and Sussman (2020), Cullina and Kiyavash (2017, 2016), Barak et al. (2018), Cullina et al. (2018) for a litany of graph matchability results in the single layer setting.

In the noisy subgraph detection problem outlined above, where $g \in \mathcal{G}_m$ and $h \in \mathcal{G}_n$ for $n > m$ (with respective adjacency matrices A and B), there are a number of ways of adapting d_{GM} to the setting where the graphs are different sizes (see, for example, Appendix F of Bento and Ioannidis 2019). If we desire our dissimilarity to penalize extraneous structure (i.e., non-edges mapped to edges) and reward both common edges and common non-edges, we can seek a subgraph g' (with adjacency matrix B') of h that minimizes $\min_{P \in \Pi_m} \|AP - PB'\|_F$. This is effectively minimizing $d_{\text{GM}}(g, \cdot)$ over subgraphs of

h of order m , which amounts to finding the closest (according to d_{GM}) induced subgraph of h to g . Equivalently, this can be cast as seeking an element P in

$$\Pi_{m,n} = \{Q \in \{0, 1\}^{m \times n} \mid Q\bar{1}_n = \bar{1}_m, \bar{1}_m^T Q \leq \bar{1}_n^T\}$$

minimizing $\|A - PBP^T\|_F$ (see Remark 1 below for detail). If we desire our dissimilarity to only reward recovered structure, we can seek a subgraph g' (with adjacency matrix B') of h that maximizes $\max_{P \in \Pi_m} \text{trace}(AP(B')^T P^T)$. This is effectively minimizing

$$d_{GM}^{(2)}(g, g') = \left(\max_{P \in \Pi_m} \text{trace}(AP(B')^T P^T) \right)^{-1}$$

over subgraphs g' of h of order m . Equivalently, this can be cast as seeking an element P in $\Pi_{m,n}$ maximizing $\text{trace}(APB^T P^T)$ (see Remark 1 below) for detail, which amounts to finding the closest (closest in that it has the most common edges) subgraph of h to g . Note that these two distances/dissimilarities can be obtained from d_{GM} directly by appropriately weighting and padding g and h ; see Fishkind et al. (2015), Sussman et al. (2019) and Remark 1.

Remark 1

Let $g \in \mathcal{G}_m$ with adjacency matrix A and $h \in \mathcal{G}_n$ with adjacency matrix B . In Fishkind et al. (2019), the method for finding the closest (according to d_{GM}) induced subgraph of h to g was to match

$$(2A - \bar{1}_m \bar{1}_m^T) \oplus \mathbf{0}_{n-m} = \begin{pmatrix} 2A - \bar{1}_m \bar{1}_m^T & \mathbf{0}_{m,n-m} \\ \mathbf{0}_{n-m,m} & \mathbf{0}_{n-m} \end{pmatrix}$$

to $2B - \bar{1}_n \bar{1}_n^T$ using d_{GM} . For $Q \in \Pi_n$, write

$$Q = \begin{pmatrix} P \\ Q' \end{pmatrix}$$

where $P \in \Pi_{m,n}$. Then, matching $(2A - \bar{1}_m \bar{1}_m^T) \oplus \mathbf{0}_{n-m}$ to $2B - \bar{1}_n \bar{1}_n^T$ is equivalent to maximizing

$$\begin{aligned} & \text{trace}([(2A - \bar{1}_m \bar{1}_m^T) \oplus \mathbf{0}_{n-m}]Q(2B - \bar{1}_n \bar{1}_n^T)^T Q^T) \\ &= \text{trace}(P^T(2A - \bar{1}_m \bar{1}_m^T)P(2B - \bar{1}_n \bar{1}_n^T)^T) \\ &= 4\text{trace}(APB^T P^T) - 2\text{trace}(\underbrace{(\bar{1}_m \bar{1}_m^T)PB^T P^T}_{=\|PBP^T\|_F^2}) \\ &\quad - 2\text{trace}(\underbrace{AP(\bar{1}_n \bar{1}_n^T)P^T}_{=\|A\|_F^2}) + \text{trace}((\bar{1}_m \bar{1}_m^T)P(\bar{1}_n \bar{1}_n^T)P^T) \end{aligned} \tag{1}$$

and finding the optimal Q for Eq. (1) is equivalent to finding the optimal $P \in \Pi_{m,n}$ minimizing

$$\|A\|_F^2 + \|PBP^T\|_F^2 - 2\text{trace}(APB^T P^T) = \|A - PBP^T\|_F^2.$$

Moreover, the above derivations show us that for $P \in \Pi_{m,n}$, minimizing $\|A - PBP^T\|_F^2$ is not equivalent to maximizing $\text{trace}(APB^T P^T)$. Indeed, in Fishkind et al. (2019), the best fitting (non-induced) subgraph of h to g is found by maximizing (over $Q \in \Pi_n$)

$$\text{trace}((A \oplus \mathbf{0}_{n-m})QB^T Q^T),$$

which is easily seen to be equivalent to maximizing

$$\text{trace}(APB^T P^T)$$

over $P \in \Pi_{m,n}$ as claimed.

Remark 2

The graph matching distance can be easily adapted to the weighted graph setting. Letting \mathcal{W}_n be the set of all n -vertex graphs with nonnegative edge weights, we can associate $g = (V, E, W) \in \mathcal{W}_n$ with a weighted adjacency matrix A so that $A_{i,j} = \mathbb{1}\{(i,j) \in E\}W(\{(i,j)\})$. For $g, h \in \mathcal{W}_n$ with respective weighted adjacency matrices A and B , we define $d_{\text{GM}}(g, h) = \min_{P \in \Pi_n} \|AP - PB\|_F$. Extensions to the setting where $g \in \mathcal{W}_m$ and $h \in \mathcal{W}_n$ are then achieved as above.

Multiplex networks

Before lifting the graph matching distance/problem to the multiplex setting, we first need to define precisely what we mean by a multiplex graph. At the core of this paper, we view the template as being equal or lower order than the background. Moreover, our definition of multiplex networks, ideally, would allow for differing graph orders across the multiple layers within a single graph. To allow for these expected data nuances in the multiplex setting, we consider the following multiplex graph model; see Kivelä et al. (2014) for a thorough overview of this and other multiplex network formulations.

Definition 1 The c -tuple $\mathbf{g} = (g_1, g_2, \dots, g_c)$ is an n -vertex multiplex network if for each $i = 1, 2, \dots, c$, we have that $g_i \in \mathcal{G}_{n_i} = \{n_i\text{-vertex labeled graphs}\}$, and the vertex sets $(V_i = V(g_i))_{i=1}^c$ further satisfy the following:

- (i) For each $i \in [c]$, we have that $V(g_i) \subseteq [n]$;
- (ii) $\bigcup_{i=1}^c V(g_i) = [n]$;
- (iii) For each $\mathcal{I} \subset [c]$, we have that $(\bigcup_{i \in \mathcal{I}} V(g_i)) \cap (\bigcup_{j \notin \mathcal{I}} V(g_j)) \neq \emptyset$;
- (iv) The layers are a priori node aligned; i.e., vertices sharing the same label across layers correspond to the same entity in the network.

Note that each vertex $v \in [n]$ need not appear in each channel $i \in [c]$, however, we do require that we cannot partition the graph layers into two sets of vertex-disjoint networks. We will denote the set of c -layer, n -vertex multiplex networks via \mathcal{M}_n^c . We note here that in the subsequent sections we will describe the M-GMMF and its associated primitives in the setting of unweighted multiplex networks. This is done to streamline exposition, and analogues in the weighted graph setting can be defined accordingly.

The multiplex graph matching distance

As mentioned previously, in the single layer setting the two subgraph matching distances—the ones based on d_{GM} and $d_{\text{GM}}^{(2)}$ —can be obtained from d_{GM} via appropriate padding schemes applied to the template g and background h . This will be the approach we take in order to define the analogues of these distances in the multiplex setting, as this will allow us to circumvent the difficulties of differing number of vertices across graphs and across the layers of a given graph. To this end, we consider the following pair of padding schemes (adapted here from Fishkind et al. 2019; Sussman et al. 2019). Letting $\mathbf{g} \in \mathcal{M}_m^c$ and $\mathbf{h} \in \mathcal{M}_n^c$ with $m \leq n$, we define

- i. (Naive Padding) For each $i \in [c]$, define the adjacency matrices $\tilde{A}_i \in \mathbb{R}^{m \times m}$ and $\tilde{B}_i \in \mathbb{R}^{n \times n}$ via

$$\tilde{A}_i(u, v) = \begin{cases} 1 & \text{if } u, v \in V(g_i), \text{ and } \{u, v\} \in E(g_i); \\ 0 & \text{if } u, v \in V(g_i), \text{ and } \{u, v\} \notin E(g_i); \\ 0 & \text{if } u \text{ or } v \in [m] \setminus V(g_i); \end{cases}$$

$$\tilde{B}_i(u, v) = \begin{cases} 1 & \text{if } u, v \in V(h_i), \text{ and } \{u, v\} \in E(h_i); \\ 0 & \text{if } u, v \in V(h_i), \text{ and } \{u, v\} \notin E(h_i); \\ 0 & \text{if } u \text{ or } v \in [n] \setminus V(h_i); \end{cases}$$

Denote $\tilde{\mathbf{A}} = (\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_c)$ and $\tilde{\mathbf{B}} = (\tilde{B}_1, \tilde{B}_2, \dots, \tilde{B}_c)$. This padding scheme effectively appends the number of isolated vertices to each graph layer needed in order to make the number of vertices in all layers equal (equal m in \mathbf{g} and n in \mathbf{h}).

- ii. (Centered Padding) For each $i \in [c]$, define the weighted adjacency matrices $\hat{A}_i \in \mathbb{R}^{m \times m}$ and $\hat{B}_i \in \mathbb{R}^{n \times n}$ via

$$\hat{A}_i(u, v) = \begin{cases} 1 & \text{if } u, v \in V(g_i), \text{ and } \{u, v\} \in E(g_i); \\ -1 & \text{if } u, v \in V(g_i), \text{ and } \{u, v\} \notin E(g_i); \\ 0 & \text{if } u \text{ or } v \in [m] \setminus V(g_i); \end{cases}$$

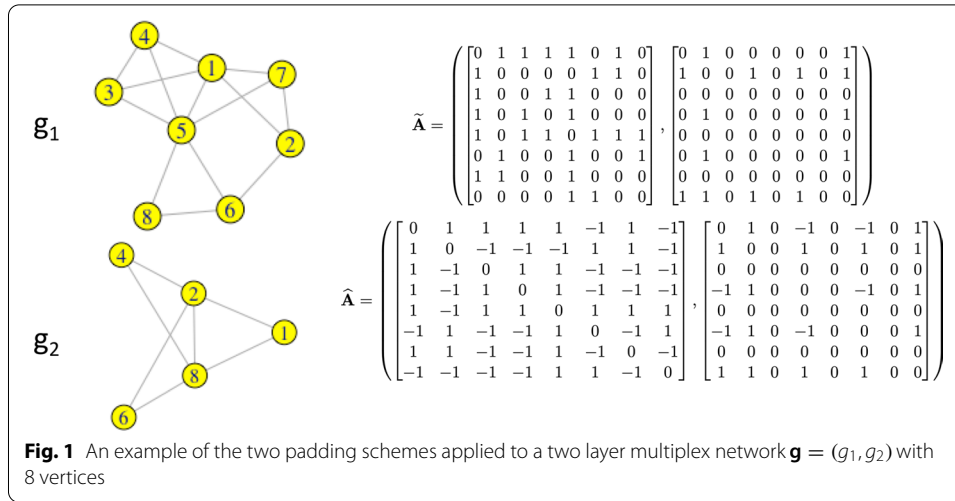
$$\hat{B}_i(u, v) = \begin{cases} 1 & \text{if } u, v \in V(h_i), \text{ and } \{u, v\} \in E(h_i); \\ -1 & \text{if } u, v \in V(h_i), \text{ and } \{u, v\} \notin E(h_i); \\ 0 & \text{if } u \text{ or } v \in [n] \setminus V(h_i); \end{cases} \quad (2)$$

Denote $\hat{\mathbf{A}} = (\hat{A}_1, \hat{A}_2, \dots, \hat{A}_c)$ and $\hat{\mathbf{B}} = (\hat{B}_1, \hat{B}_2, \dots, \hat{B}_c)$. This padding scheme makes the number of vertices in all layers equal, as well as penalizes non-existent edges.

For an example of the two padding schemes applied to a graph in \mathcal{M}_8^2 , see Figure 1.

We are now ready to define the multiplex analogues of our subgraph matching dissimilarities. Let $\mathbf{g} \in \mathcal{M}_m^c$ (with naive and centered paddings given by $\tilde{\mathbf{A}}$ and $\hat{\mathbf{A}}$ resp.,) and $\mathbf{h} \in \mathcal{M}_n^c$ with $m < n$ (with naive and centered paddings given by $\tilde{\mathbf{B}}$ and $\hat{\mathbf{B}}$ resp.,). The multiplex analogue of $d_{\text{GM}}^{(2)}$, that only rewards recovered common edge structure, then seeks a subgraph $\mathbf{g}' \in \mathcal{M}_m^c$ of \mathbf{h} (with corresponding naive and centered paddings given by $\tilde{\mathbf{B}}'$ and $\hat{\mathbf{B}}'$ resp.,) that minimizes

$$d_{\text{M-GM}}^{(2)}(\mathbf{g}, \mathbf{g}') := \left(\max_{P \in \Pi_m} \sum_{i=1}^c \text{trace}(\tilde{A}_i P (\tilde{B}_i')^T P^T) \right)^{-1}.$$



For matrices $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{n-m \times n-m}$, we consider the matrix direct sum $A \oplus B \in \mathbb{R}^{n \times n}$ defined via

$$A \oplus B = \begin{bmatrix} A & \mathbf{0}_{m, n-m} \\ \mathbf{0}_{n-m, m} & B \end{bmatrix},$$

where we recall for integral $k, \ell > 0$, $\mathbf{0}_{k, \ell}$ (resp., $\mathbf{0}_k$) is the $k \times \ell$ (resp., $k \times k$) matrix of all 0's. Finding the m -vertex subgraphs of \mathbf{h} minimizing $d_{\text{M-GM}}^{(2)}(\mathbf{g}, \cdot)$ is equivalent to finding all P in

$$\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|(\tilde{A}_i \oplus \mathbf{0}_{n-m})P - P\tilde{B}_i\|_F^2 = \arg \min_{P \in \Pi_n} \sum_{i=1}^c -\text{tr}((\tilde{A}_i \oplus \mathbf{0}_{n-m})P\tilde{B}_i^T P^T), \quad (3)$$

as each P in Eq. (3) matches \mathbf{g} to a subgraph of \mathbf{h} minimizing $d_{\text{M-GM}}^{(2)}$. This formulation in Eq. (3) effectively seeks to maximize the number of common edges between the multiplex template and multiplex background, where all edges across all channels are weighted equally (see Fishkind et al. 2019; Bento and Ioannidis 2019 for the monoplex analogue).

The multiplex analogue of the dissimilarity that also penalizes extraneously recovered structure seeks an induced subgraph $\mathbf{g}' \in \mathcal{M}_m^c$ of \mathbf{h} that minimizes the centered matching dissimilarity

$$d_{\text{M-GM}}^{(c)}(\mathbf{g}, \mathbf{g}') := \min_{P \in \Pi_m} \sum_{i=1}^c \|\hat{A}_i P - P\hat{B}_i'\|_F^2.$$

Finding the m -vertex subgraphs of \mathbf{h} minimizing $d_{\text{M-GM}}^{(c)}(\mathbf{g}, \cdot)$ is equivalent to finding all P in

$$\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m})P - P\hat{B}_i'\|_F^2 = \arg \min_{P \in \Pi_n} \sum_{i=1}^c -\text{tr}((\hat{A}_i \oplus \mathbf{0}_{n-m})P\hat{B}_i'^T P^T), \quad (4)$$

as each P in Eq. (4) matches \mathbf{g} to a subgraph of \mathbf{h} minimizing $d_{\text{M-GM}}^{(c)}(\mathbf{g}, \cdot)$. If for each i , we have that $V(h_i) = [n] \supset [m_i] = V(g_i)$, then the formulation in Eq. (4) effectively seeks to minimize the number of disagreements (edge mapped to non-edge and vice versa) induced between the background and the matched subgraphs in the template, where all disagreements across all channels are weighted equally. Given the interpretation of Eqs. (3) and (4), the appropriate padding schemes to deploy in practice depends on the underlying problem assumptions and setting. In the sequel, we will let the optimization in Eq. (3) be known as the *naive multiplex graph matching problem* and the optimization in Eq. (4) be known as the *centered multiplex graph matching problem*.

Remark 3

Our formulation of the Multiplex GMP is (assuming channels of equal order across the adjacency matrix lists \mathbf{A} and \mathbf{B})

$$\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|A_i P - P B_i\|_F^2.$$

rather than a formulation weighting the matching in each channel via

$$\arg \min_{P \in \Pi_n} \sum_{i=1}^c \lambda_i \|A_i P - P B_i\|_F^2,$$

for $\lambda_i > 0$. Note that we implemented the Multiplex GMP for different values of λ , though in our experiments below we found that matching results with $\lambda_i = 1$ work suitably well while being more robust than other ad-hoc weighting strategies. As such, we only present the results with $\lambda_i = 1$, though finding optimal weight for a given graph sequence is an important learning task in its own right and is the subject of present research. Moreover, this allows each edge in each template channel to be weighted equally, which may be desirable. In the case that one or more channels is more informative or of higher import than the others, then choosing appropriate λ 's to overweight the matching in those channels may be desirable.

Multiplex graph matching matched filters

Our approach to approximately solving Eqs. (3) and (4) uses the template \mathbf{g} as a filter to search the background \mathbf{h} for approximate matches. Similar to how convolutional and matched filters are used in computer vision tasks, we use the template to scan the background for matching structure, though the mechanism for scanning is not exhaustive search but rather an adaptation of the Frank-Wolfe based matching algorithm FAQ of Vogelstein et al. (2014). Below we provide the details of our multiplex FAQ algorithm, as well as how this is implemented towards our goal of building effective multiplex graph matching matched filters.

Multiplex FAQ

One of the principle motivations for our additive multiplex graph distance formulation is that it allows us to adapt gradient descent based search algorithms to the multiplex setting with ease. One such algorithm that has proven useful in the literature is the FAQ algorithm of Vogelstein et al. (2014). This algorithm—based on the Frank-Wolfe method for constrained gradient descent applied to an indefinite relaxation of the graph matching problem—proceeds as follows (in the monoplex setting): Given graphs $g, h \in \mathcal{G}_n$ with adjacency matrices A, B , a threshold ϵ , and initialization $P^{(0)} \in \mathcal{D}_n$ (where \mathcal{D}_n is the set of doubly stochastic $n \times n$ matrices, i.e., the convex hull of Π_n):

1. While $\|P^{(t)} - P^{(t-1)}\|_F > \epsilon$ at step $t \geq 1$, repeat the following iterative loop:
 - i. Letting $f(P) = -\text{trace}(APB^T P^T)$, compute $\nabla f(P^{(t)}) = -A^T P^{(t)} B - AP^{(t)} B^T$.
 - ii. Find the optimal search direction for gradient descent by finding

$$Q^{(t)} = \arg \min_{Q \in \mathcal{D}_n} \text{trace}(\nabla f(P^{(t)})^T Q);$$

note that this is an instantiation of the famous linear assignment problem (LAP) and can be solved via the Hungarian method in $O(n^3)$ time (Jonker and Volgenant 1987).
 - iii. Set α^* equal to the minimizer of $f(\alpha P^{(t)} + (1 - \alpha)Q^{(t)})$. This line search in the direction of $Q^{(t)}$ can be done efficiently, as it amounts to optimizing a simple quadratic function of α .
 - iv. Set $P^{(t+1)} = \alpha^* P^{(t)} + (1 - \alpha^*)Q^{(t)}$.
2. Project the (possibly) interior point solution $P^{(\text{final})}$ to the set of permutation matrices by solving the following LAP

$$P^* = \arg \min_{P \in \Pi_n} \text{trace}(\nabla f(P^{(\text{final})})^T P).$$

The FAQ algorithm is theoretically principled (Lyzinski et al. 2015), performs well in a number of real-data alignment tasks (see Vogelstein et al. 2014 for a comparison of FAQ to other graph matching approaches), and is flexible enough to easily handle weighted and directed graphs, seeded vertices (Fishkind et al. 2019), matching graphs of different orders (Fishkind et al. 2019; Sussman et al. 2019), and, as we will see below, matching multiplex networks.

The above algorithm extends easily to the multiplex setting as follows. Let $\mathbf{A} = (A_1, \dots, A_c)$ and $\mathbf{A}' = (A'_1, \dots, A'_c)$ be the (possibly padded) adjacency matrix vectors of two multiplex graphs in \mathcal{M}_n^c where each A_i and A'_i is assumed to be an $n \times n$ matrix, we consider minimizing the objective function

$$f_M(P) = - \sum_{i=1}^c \text{trace}(A_i P (A'_i)^T P^T)$$

using the same Frank-Wolfe procedure outlined above. To wit, given a threshold ϵ and initialization $P^{(0)} \in \mathcal{D}_n$, we proceed via

1. While $\|P^{(t)} - P^{(t-1)}\|_F > \epsilon$ at step $t \geq 1$, repeat the following iterative loop:
 - i. Compute $\nabla f_M(P^{(t)}) = -\sum_{i=1}^c A_i^T P^{(t)} A'_i - A_i P^{(t)} (A'_i)^T$.
 - ii. Set $Q^{(t)} = \arg \min_{Q \in \mathcal{D}_n} \text{trace}(\nabla f_M(P^{(t)})^T Q)$
 - iii. Set α^* equal to the minimizer of $f_M(\alpha P^{(t)} + (1 - \alpha)Q^{(t)})$.
 - iv. Set $P^{(t+1)} = \alpha^* P^{(t)} + (1 - \alpha^*)Q^{(t)}$.
2. Lastly compute $P^* = \arg \min_{P \in \Pi_n} \text{trace}(\nabla f(P^{(\text{final})})^T P)$. Return approximate solution P^* .

This multiplex FAQ algorithm (denoted M-FAQ) aims to, ideally, find the global minimizer of $f_M(P)$. We will see in the next section how these ideas can be leveraged towards building our multiplex graph matching matched filter.

Remark 4

One of the advantages of FAQ and M-FAQ is the ease in which it allows seeded vertices to be incorporated (Fishkind et al. 2019). In the setting of graph alignment, seeded vertices are those whose correspondence across networks is a priori known. This is incorporated into M-FAQ as follows. Without loss of generality, letting the first $s < n$ vertices of $\mathbf{g}, \mathbf{g}' \in \mathcal{M}_n^c$ (with respective adjacency matrix lists \mathbf{A} and \mathbf{A}') be seeded via the identity mapping, we can consider incorporating these seeds via:

- i. *A hard seeded approach:* We seek to align \mathbf{g} and \mathbf{g}' by finding

$$\arg \min_{P \in \Pi_{n-s}} \sum_{i=1}^c \|A_i(I_s \oplus P) - (I_s \oplus P)A'_i\|_F^2.$$

This effectively enforces the alignment across seeded vertices must hold in any optimal matching across \mathbf{g} and \mathbf{g}' , thus reducing the dimension of the search space of the graph matching problem. In this case, M-FAQ proceeds as outlined above with the objective function

$$f_M(P) = -\sum_{i=1}^c \text{trace}(A_i(I_s \oplus P)(A'_i)^T(I_s \oplus P)^T),$$

with the computation of the gradient and search steps proceeding analogously.

- ii. *A soft seeded approach:* We seek to align \mathbf{g} and \mathbf{g}' by finding

$$\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|A_i P - P A'_i\|_F^2,$$

and initializing our optimization algorithm at a P of the form $I_s \oplus Q$ (where the form of Q may be dictated by whether the algorithm relaxes the feasible region). This alignment across seeded vertices is encoded into the initialization though this alignment is not necessarily enforced in an optimal matching across \mathbf{g} and \mathbf{g}' . This

is more appropriate in settings where the seeds are potentially noisy or errorful (Fang et al. 2018).

In both the soft and hard seeded approaches, the performance of FAQ and M-FAQ is often dramatically improved even with the inclusion of relatively few seeds (Fishkind et al. 2019; Patsolic et al. 2014).

Multiplex graph matching matched filters

In the setting of multiplex noisy subgraph detection, there may be many subgraphs in \mathbf{h} optimally matching to \mathbf{g} . To account for this (and the presence of spurious local minima for the indefinite $f_M(D)$ for D in \mathcal{D}_n), the multiplex graph matching matched filter (M-GMMF) algorithm proceeds as follows: Use multiple random restarts of the M-FAQ algorithm applied to either Eq. (3) or (4) (depending on the desired structure to recover) to find multiple candidate matches for the template \mathbf{g} in \mathbf{h} . This approach was developed in the monoplex setting (using FAQ) in Sussman et al. (2019), with our contribution herein being the extension to the multiplex setting.

Given multiplex graphs $\mathbf{g} \in \mathcal{M}_m^c$ and $\mathbf{h} \in \mathcal{M}_n^c$ with $m \leq n$, an appropriate padding scheme (either naive or centered), a tolerance $\epsilon \in \mathbb{R} > 0$, a random initialization parameter γ , and the number of restarts N , M-GMMF proceeds as follows:

1. Pad \mathbf{g} and \mathbf{h} accordingly. In the naive (resp., centered) padding regime, the padded adjacency matrix list of \mathbf{g} is denoted via $\tilde{\mathbf{A}}$ (resp., $\hat{\mathbf{A}}$), and the padded \mathbf{h} via $\tilde{\mathbf{B}}$ (resp., $\hat{\mathbf{B}}$).
2. For $k = 1, 2, \dots, N$, repeat the following
 - i. Create a random initialization for the M-FAQ algorithm via $P^{(0)} \leftarrow \alpha \tilde{\mathbf{1}}_n \tilde{\mathbf{1}}_n^T / n + (1 - \alpha)P$ where $P \sim \text{Unif}(\Pi_n)$, $\alpha \sim \text{Unif}[0, \gamma]$.

Note that this is one possible random initialization scheme (the random jitter of the bar-ycenter $\tilde{\mathbf{1}}_n \tilde{\mathbf{1}}_n^T / n$ of \mathcal{D}_n).

- ii. In the naive (resp., centered) padding regime, find a candidate match for \mathbf{g} via $P_k^* = \text{M-FAQ}(\tilde{\mathbf{A}} \oplus \mathbf{0}_{n-m}, \tilde{\mathbf{B}}, P^{(0)}, \epsilon)$ (resp., $P_k^* = \text{M-FAQ}(\hat{\mathbf{A}} \oplus \mathbf{0}_{n-m}, \hat{\mathbf{B}}, P^{(0)}, \epsilon)$)
3. Rank the matchings $\{P_1^*, P_2^*, \dots, P_N^*\}$ by increasing value of the multiplex graph matching objective function, Eq. (3) or (4), depending on the padding regime selected. This ranked list provides a candidate list of matchings/detected subgraphs for \mathbf{g} in \mathbf{h} .

Note that code implementing the above M-GMMF and M-FAQ procedures can be downloaded as part of our R package, `iGraphMatch`, which is available on CRAN or can be downloaded at <https://github.com/dpmcsuss/iGraphMatch>.

Remark 5

Effectively, the M-GMMF algorithm uses the multiplex template and a multiplex matching algorithm \mathcal{A} (M-FAQ in our implementation) to search Π_n for relevant regions of \mathbf{h} that align well to \mathbf{g} . The multiple restarts in Step 2 of the procedure are needed in the case of

$\mathcal{A} = \text{M-FAQ}$ to account for multiple possible globally best matched and the fact that the objective function is relaxed to an indefinite quadratic program with myriad local minima in the feasible region. For approximate combinatorial \mathcal{A} , the restarts may be appropriate as well, while for continuous, convex relaxation algorithms (see, for example, Bento and Ioannidis 2019), this step may not be necessary if we believe there is a single global minima.

Multiplex template matchability

In order to theoretically understand the efficacy of our M-GMMF approach, we consider the following probabilistic framework (note that real-data experimentation demonstrating the empirical success of M-GMMF is presented in “Experiments” section). Considering a random graph model for both \mathbf{g} and \mathbf{h} in which a stochastically noisy copy of \mathbf{g} has been embedded into \mathbf{h} , we seek to understand to what degree an oracle algorithm (i.e., those that can exactly solve Eq. (3) and (4)) can identify the noisy copy of \mathbf{g} in \mathbf{h} . This offers an important suitability test for M-GMMF, as if even oracle approaches are not able to uncover probabilistically errorful copies of \mathbf{g} , then practically M-GMMF will most likely not be suitable for finding optimal subgraph matches in real data, high noise settings.

Moreover, to understand this problem further, we develop novel results on multiplex graph matchability (for \mathbf{g} and \mathbf{h} of the same order) that may be of independent interest, as well as generative multiplex error models in which to explore this question. The layout of this section is as follows. We develop a pair of error models, the ME and MS models, in “Background/template error models”. Within each error model, we tackle the question of template matchability in “Template matchability” section. Interspersed with these matchability results are simulations further illuminating the corresponding matching theory.

Background/template error models

We first consider a random graph model in the monoplex setting, in which a stochastically noisy copy of a template g has been embedded into the background h . Next, we leverage this model in the multiplex setting by proposing a pair of generative error models wherein the template \mathbf{g} is embedded as an errorful subgraph into the background \mathbf{h} .

Definition 2 (See Arroyo et al. 2018) Consider a graph g with $V(g) \subset [n]$. Let the centered, padded adjacency matrix (as in Eq. (2)) of g be denoted $\hat{A} \in \mathbb{R}^{n \times n}$. Let $E \in [0, 1]^{n \times n}$ be a symmetric, hollow (hollow indicating that the diagonal entries of E are set to 0) matrix. The graph-valued random variable $\mathcal{E}(g)$ with vertex set equal to $V(g)$ and random centered, padded adjacency matrix $\hat{A}_{g,E}$, which models passing g through an errorful channel E , is defined as follows. For each $\{i, j\} \in \binom{[n]}{2}$,

$$\hat{A}_{g,E}(i, j) = \hat{A}(i, j) \cdot (1 - 2X(i, j)),$$

where $X(i, j) \stackrel{\text{ind.}}{\sim} \text{Bern}(E(i, j))$.

The two generative models we then consider for lifting Definition 2 to the multiplex setting are defined below. Each models a particular setting in which multilayer networks can appear in real-data settings.

- i. (*Single Channel Source, Multiplex Error; abbreviated ME*) There is a single non-random background source graph $h \in \mathcal{G}_n$ and a non-random source template $g = h[m] \in \mathcal{G}_m$, and two multi-channel errorful filters $\mathbf{E}^{(1)} = (E_1^{(1)}, \dots, E_c^{(1)})$, with each $E_i^{(1)}$ acting on h , and $\mathbf{E}^{(2)} = (E_1^{(2)}, \dots, E_c^{(2)})$, with each $E_i^{(2)}$ acting on g . We observe $\mathbf{H} = (\mathcal{E}_1^{(1)}(h), \dots, \mathcal{E}_c^{(1)}(h))$ as the multiplex background and $\mathbf{G} = (\mathcal{E}_1^{(2)}(g), \dots, \mathcal{E}_c^{(2)}(g))$ as the multiplex template. By assumption, the errorful filters act independently across channels within \mathbf{G} and \mathbf{H} , and independently across \mathbf{G} and \mathbf{H} . In this model, by construction each $|V(G_i = \mathcal{E}_2^{(i)}(g))| = [m]$ and each $|V(H_i = \mathcal{E}_1^{(i)}(h))| = [n]$. This situation is used to model settings in which the multiplex layers can be thought of as errorful realizations of a single true but unknown network, and the edges across layers are of the same fundamental kind. For example, multiplex connectomes in which each layer graphically represents a (for example) DT-MRI scan for a patient at a different point in time; see, for example, Zuo et al. 2014; Kiar et al. 2018.
- ii. (*Single Channel Errors, Multiplex Source; abbreviated MS*) The non-random background and non-random template source graphs are multiplex. To wit, let $\mathbf{g} \in \mathcal{M}_m^c$ and $\mathbf{h} \in \mathcal{M}_n^c$ satisfy the following: For each $i \in [c]$, let $\hat{\mathbf{C}}$ and $\hat{\mathbf{D}}$ be the centered paddings of \mathbf{g} and \mathbf{h} respectively. We assume then that $\hat{\mathbf{C}}_i = \hat{\mathbf{D}}_i[m]$ (i.e., $\hat{\mathbf{C}}_i$ —the padded adjacency matrix of g_i —is the $m \times m$ principal submatrix of $\hat{\mathbf{D}}_i$ —the padded adjacency matrix of h_i). In this model, there are two multi-channel errorful filters: $\mathbf{E}^{(1)} = (E_1^{(1)}, \dots, E_c^{(1)})$ and $\mathbf{E}^{(2)} = (E_1^{(2)}, \dots, E_c^{(2)})$. For each $i \in [c]$, $E_i^{(1)} \in \mathbb{R}^{n \times n}$ acts on h_i , and $E_i^{(2)} \in \mathbb{R}^{m \times m}$ acts on g_i . We observe

$$\mathbf{H} = (\mathcal{E}_1^{(1)}(h_1), \mathcal{E}_2^{(1)}(h_2), \dots, \mathcal{E}_c^{(1)}(h_c))$$

as the multiplex background, and

$$\mathbf{G} = (\mathcal{E}_1^{(2)}(g_1), \mathcal{E}_2^{(2)}(g_2), \dots, \mathcal{E}_c^{(2)}(g_c))$$

as the multiplex template. As above, the errorful filters act independently across channels within \mathbf{G} and \mathbf{H} , and independently across \mathbf{G} and \mathbf{H} . Note that if the template (resp., background) channels have non-identical vertex sets, then this will be preserved in the errorful template (resp., background). This situation is used to model settings in which we errorfully observe a given multiplex background and template. This is the case in many applications where the layers in \mathbf{G} and \mathbf{H} represent different edge types; e.g., a multiplex social network where the layers represent different friendship modalities (Magnani and Rossi 2011) or a multilayer connectome where the layers represent different synaptic types (Chen et al. 2016).

It may be convenient to view g and h (resp., \mathbf{g} and \mathbf{h}) as realizations from graph-valued random variables in the ME (resp., MS) model. In this case, we will assume the actions of the errorful filters on g and h (resp., \mathbf{g} and \mathbf{h}) are also independent of the random g and h (resp., \mathbf{g} and \mathbf{h}).

Considering the models above, in order for our M-GMMF approach to possibly recover the *true* errorful induced subgraph of \mathbf{h} corresponding to \mathbf{g} , we need for the global minimum of the Multiplex GMP to be in $\mathcal{P}_{m,n} := \{I_m \oplus P : P \in \Pi_{n-m}\}$. This is the multiplex analogue of template matchability, i.e., uncovering conditions under which an oracle graph matching will recover a true but unknown latent vertex alignment. Here, that alignment is represented by \mathbf{g} being an errorful version of $\mathbf{h}[m]$.

Template matchability

In this section, we will explore the benefit of considering multiplex versus monoplex networks when considering template matchability in both the MS and ME models. We note here that while the proof mechanism behind the theory underlying the matchability results in the multiplex setting differs only slightly from those used in the monoplex setting of Sussman et al. (2019), we stress that the end results are novel and demonstrate the utility of considering multiple channels in a novel manner. Recall below that for integer $n > 0$, J_n is the $n \times n$ matrix with all diagonal entries identically set to 0 and all off-diagonal entries set to 1.

In the MS model, let $\mathbf{g} \in \mathcal{M}_m^c$ and $\mathbf{h} \in \mathcal{M}_n^c$ be the respective template and background source graphs, with respective centered, padded adjacency matrices given respectively by $\widehat{\mathbf{C}}$ and $\widehat{\mathbf{D}}$ satisfying $\widehat{\mathbf{C}}_i = \widehat{\mathbf{D}}_i[m]$ for all $i \in [c]$. Assume that the errorful filters satisfy for each $i \in [c]$, $E_i^{(1)} = q_i J_n$ and $E_i^{(2)} = s_i J_m$ (where $s_i = s_i(n)$ and $q_i = q_i(n)$ are allowed to vary with n). If $c = 1$, and $s_1, q_1 = 1/2$, then the observed background and template are effectively independent $ER(n, 1/2)$ and $ER(m, 1/2)$ networks, respectively. It is immediate then that the optimal permutation aligning the background to the template will almost surely not be in $\mathcal{P}_{m,n}$. If $c > 1$, can strongly correlated signal present in less noisy channels be used to counteract the excess error caused by noisier channels?

The next result tackles this from a template matchability perspective. The main result in the following theorem (Theorem 1) is that in the correlated model outlined above, if there are sufficiently many positively correlated channels relative to the number of negatively correlated channels, then under mild assumptions on the growth of m , the subgraph of \mathbf{H} minimizing the centered graph matching dissimilarity $d_{\text{M-GM}}^{(c)}(\mathbf{G}, \cdot)$, is, with high probability, uniquely the embedded noisy copy of \mathbf{G} in \mathbf{H} ; note that the proof of Theorem 1 can be found in “Appendix 1.1”. While an oracle procedure for finding the minimizer of $d_{\text{M-GM}}^{(c)}(\mathbf{G}, \cdot)$, is unknown, this result gives credence to the matched filter approach, as the global optimizing alignment sought by the multiplex matched filter will (with high probability) be the *right* alignment.

Theorem 1 *In the MS model framework, suppose that there exist constants $\alpha < 1$, $\beta > 0$, and $n_0 \in \mathbb{Z} > 0$ such that for all $n > n_0$, $m = m(n)$ satisfies $m^\alpha > \beta \log n$. Suppose that for $p_i = p = p(n)$ we have that each $h_i \stackrel{\text{ind.}}{\sim} ER(n, p)$, and that $s_i(n) = s(n) = s < 1/2$. For c_1 channels, let $q_i = q < 1/2$, for c_2 channels, let $q_i = 1/2$, and for $c_3 = c - c_1 - c_2$ channels let $q_i = 1 - q > 1/2$ (where $c_1 = c_1(n)$, $c_2 = c_2(n)$, $c_3 = c_3(n)$, $s = s(n)$ and $q = q(n)$ are allowed to vary with n). Then there exist constants $\gamma, \xi > 0$, and $n_2 \in \mathbb{Z} > 0$ such that if for all $n > n_2$,*

$$mp^2 \geq \xi \log n, \quad (5)$$

$$p(1/2 - s)(1/2 - q)(c_1 - c_3) > \gamma \sqrt{m^{\alpha-1}c}, \quad (6)$$

then for $n > \max(n_0, n_2)$, we have

$$\mathbb{P} \left(\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m})P - P\hat{B}_i\|_F^2 \notin \mathcal{P}_{m,n} \right) \leq 4n^{-2}.$$

If s, q, c, c_1 and c_2 are fixed constants that do not vary with n , we need only require $c_1 > c_3$ rather than Condition (5).

To derive analogous results to Theorem 1 in the ME model, we consider the following setting. Letting $H \sim ER(n, p = p(n)) \in \mathcal{G}_n$ with $p \leq 1/2$, and $G = H[m]$ be the respective background and template source graphs, we again assume that there exist constants $\alpha \leq 1$, $\beta > 0$, and $n_0 \in \mathbb{Z} > 0$ such that for all $n > n_0$, $m = m(n)$ satisfies $m^\alpha \geq \beta \log n$. Further assume that for each $i \in [c = c(n)]$ the errorful filters satisfy,

$$E_i^{(2)}(j, \ell) = \begin{cases} s_i = s_i(n) & \text{if } G(j, \ell) = 1 \\ q_i = q_i(n) & \text{if } G(j, \ell) = 0 \end{cases}$$

$$E_i^{(1)}(j, \ell) = \begin{cases} r_i = r_i(n) & \text{if } H(j, \ell) = 1 \\ t_i = t_i(n) & \text{if } H(j, \ell) = 0. \end{cases}$$

In the ME setting, we then have the following theorem whose proof is presented in “Appendix 1.2”. The result in Theorem 2 is similar to that of Theorem 1: In the correlated ME model outlined above, if there are sufficiently many positively correlated channels (those in c_3 and c_4) relative to the number of negatively correlated channels (those in c_1 and c_2), then under mild assumptions the subgraph of \mathbf{H} minimizing the centered graph matching dissimilarity $d_{\text{M-GM}}^{(c)}(\mathbf{G}, \cdot)$, is again, with high probability, uniquely the embedded noisy copy of \mathbf{G} in \mathbf{H} .

Theorem 2 *With setup as above, suppose that $\alpha < 1$. For*

- $c_1 = c_1(n)$ channels, suppose that $s_i + q_i = 1 + e_1 > 1$ and $r_i + t_i = 1 - e_2 < 1$;
- $c_2 = c_2(n)$ channels, suppose that $s_i + q_i = 1 - e_1 < 1$ and $r_i + t_i = 1 + e_2 > 1$;
- $c_3 = c_3(n)$ channels, suppose that $s_i + q_i = 1 + e_1 > 1$ and $r_i + t_i = 1 + e_2 > 1$;
- $c_4 = c_4(n)$ channels, suppose that $s_i + q_i = 1 - e_1 < 1$ and $r_i + t_i = 1 - e_2 < 1$,

where $e_1 = e_1(n)$ and $e_2 = e_2(n)$ can vary in n and $c = c_1 + c_2 + c_3 + c_4$. Then there exist constants $\gamma, \xi > 0$, and $n_2 \in \mathbb{Z} > 0$ such that if for all $n > n_2$

$$mp^2 \geq \xi \log n, \quad (7)$$

$$pe_1e_2[c_3 + c_4 - c_1 - c_2] > \gamma \sqrt{m^{\alpha-1}c}, \quad (8)$$

then for $n > \max(n_0, n_2)$,

$$\mathbb{P}\left(\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m})P - P\hat{B}_i\|_F^2 \notin \mathcal{P}_{m,n}\right) \leq 6n^{-2}. \quad (9)$$

If e_1 , e_2 , and c are fixed in n , we need only require $c_1 + c_2 < c_3 + c_4$ for the bound in Eq. (9) to hold for sufficiently large n .

As mentioned above, in Theorem 2, the channels in c_3 and c_4 represent those in which the template and background are positively correlated, with those in c_1 and c_2 representing those in which the template and background are negatively correlated. We can then interpret Eq. (8) as asserting that sufficiently many positively correlated channels can overcome the signal loss in negatively correlated channels. This is a unique benefit of multiple layers versus the monoplex setting, as in the single layer setting there is no universal way to overcome the negative impact of anti-correlation on graph matchability. In the following sections, we further empirically explore the effect of negatively correlated/high noise channels on multiplex matchability.

Strength in numbers

Consider $c_2 = c_3 = 0$ in Theorem 1. Condition (6) then reduces to

$$p(1/2 - s)(1/2 - q)\sqrt{c} > \gamma\sqrt{m^{\alpha-1}}, \quad (10)$$

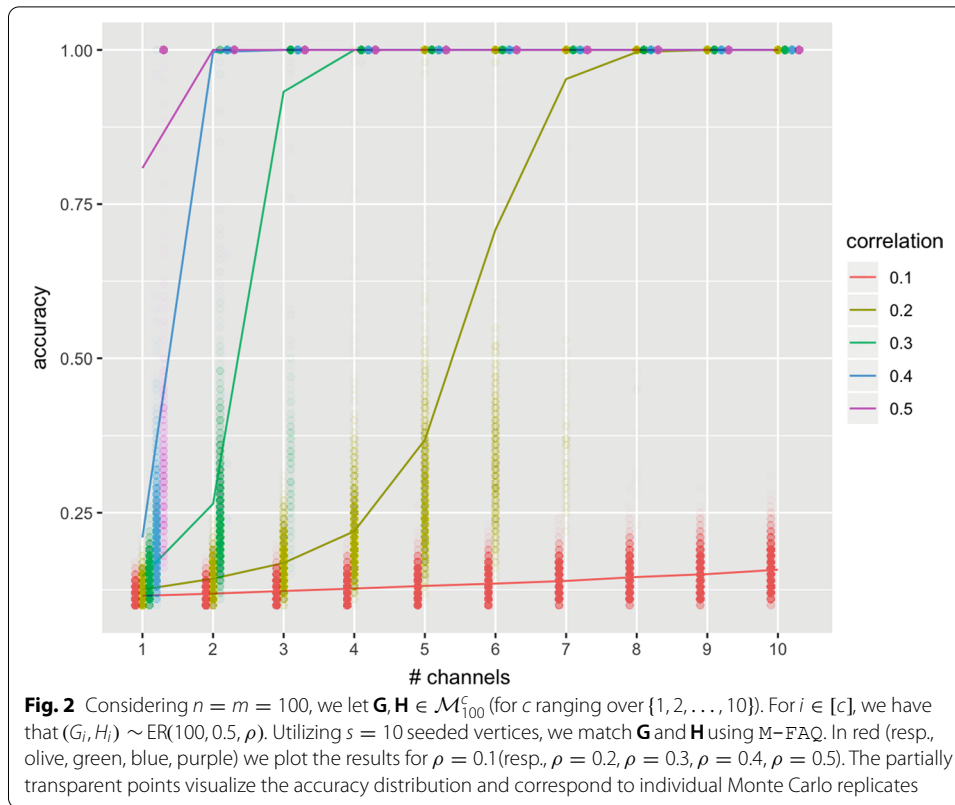
and sparsity in the background (i.e., $p \approx 0$) and large noise values (i.e., close to $1/2$) for s and q can be mitigated by choosing an appropriately large c ; effectively, multiple channels can amplify the weak signal present in each individual channel.

We explore this further in the following experiment. We will look at two different cases, first considering when $m = n$ and then when $m < n$. First, we consider $n = m = 100$ (to mitigate possible effects of template order on matching accuracy), and we let $\mathbf{G}, \mathbf{H} \in \mathcal{M}_{100}^c$ for c ranging over $\{1, 2, \dots, 10\}$. For each $i \in [c]$, we have that $(G_i, H_i) \sim \text{ER}(100, 0.5, \rho)$ (so that G_i and H_i are marginally $\text{ER}(100, 0.5)$ and edges across graphs are independent except that for each $\{j, k\} \in \binom{[100]}{2}$, we have that $\text{corr}(\mathbb{1}\{\{j, k\} \in E(G_i)\}, \mathbb{1}\{\{j, k\} \in E(H_i)\}) = \rho$).

Within this model, the channels are endowed with a natural vertex correspondence across G_i and H_i , namely the identity mapping as identically labeled edges are correlated while edges with different vertex labels are independent. Note that in the $h_i \sim \text{ER}(n, p_i)$ MS model setting, we have that

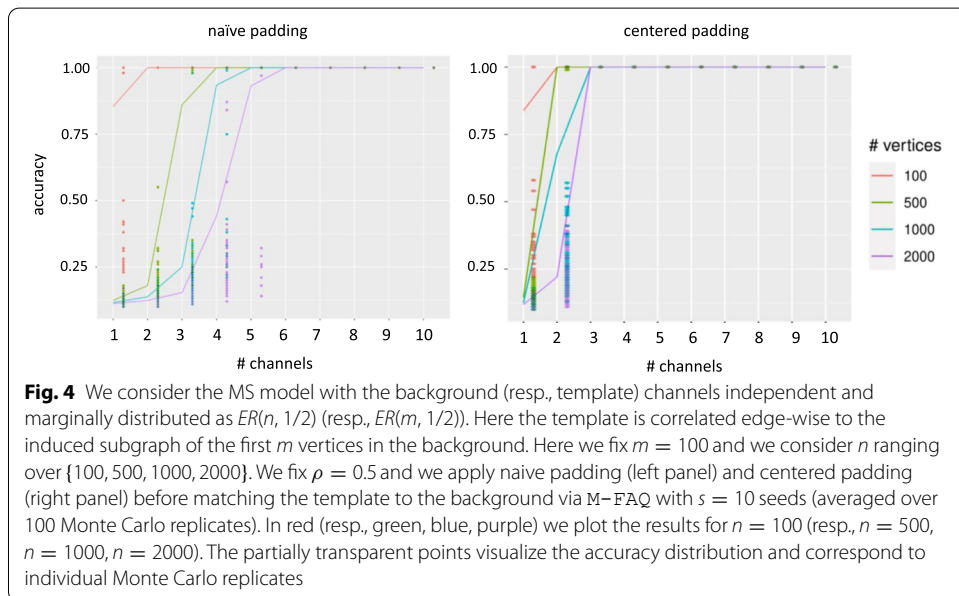
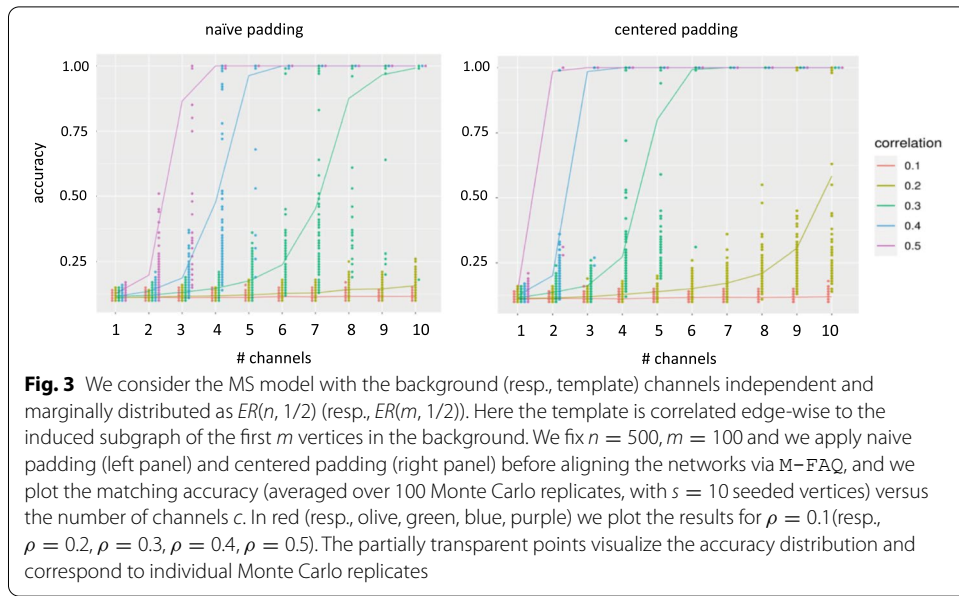
$$\text{Cov}(\mathbb{1}\{j \sim_{g_i} k\}, \mathbb{1}\{j \sim_{h_i} k\}) = p_i(1 - p_i)(1 - 2s_i)(1 - 2q_i),$$

so that the correlation between edges in G_i and H_i can be made positive or negative with judiciously chosen s_i and q_i . Considering ρ varying over $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, we match \mathbf{G} and \mathbf{H} using M-FAQ using $s = 10$ hard seeded vertices (see Remark 4). Results are plotted in Figure 2. In the figure, we plot the mean matching accuracy (i.e., the fraction of vertices whose latent alignment is recovered correctly) of M-FAQ versus c , averaged over 2000 Monte Carlo replicates. For each choice of parameters, we also plot (via the partially transparent points) the accuracy distribution corresponding to the MC replicates. In red (resp., olive, green, blue, purple) we plot the results for $\rho = 0.1$ (resp., $\rho = 0.2$, $\rho = 0.3$, $\rho = 0.4$, $\rho = 0.5$). From Figure 2, we see the expected relationship: in low



correlation settings where M-FAQ is unable to align the monoplex graphs, this can often be overcome by considering $c > 1$. Indeed, in all cases, save $\rho = 0.1$, perfect matching is achieved using $c \geq 8$ channels.

Next, we look at the case when $m < n$. In addition to examining the effect of multiple channels when weak signal is present across channels, we wish to compare the effect of different padding schemes (Naive vs Centered) in terms of the matching accuracy. Here, we are following the MS model with the background (resp., template) channels independent and marginally distributed as $\text{ER}(n, 1/2)$ (resp., $\text{ER}(m, 1/2)$). Here the template is correlated edge-wise to the induced subgraph of the first m vertices in the background (as in the MS model), with edge correlation ρ . We analyze the padding schemes' effectiveness by first varying the values of the correlation $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ while keeping $n = 500$, and $m = 100$ constant (see Figure 3) and second, by varying the background size $n \in \{100, 500, 1000, 2000\}$ while the template size $m = 100$ and the correlation $\rho = 0.5$ remain constant (see Figure 4). Letting the Naive (resp. Centered) padded template/background pair have adjacency matrix lists $(\tilde{\mathbf{G}}, \tilde{\mathbf{H}})$ (resp. $(\hat{\mathbf{G}}, \hat{\mathbf{H}})$), in each figure we consider matching the template to the background using M-FAQ (with 10 hard seeds) while varying c over $\{1, 2, \dots, 10\}$. Results are plotted in Figures 3 and 4. As in Figure 2, we plot the mean matching accuracy (i.e., the fraction of template vertices whose true latent alignment is recovered correctly in the background) of M-FAQ versus c , here averaged over 100 MC replicates. For each choice of parameters, we also plot (via the partially transparent points) the accuracy distribution corresponding to the MC replicates. In Figure 3, in red (resp., olive, green, blue, purple) we plot the results for $\rho = 0.1$ (resp.,



$\rho = 0.2, \rho = 0.3, \rho = 0.4, \rho = 0.5$). In Figure 4, in red (resp., green, blue, purple) we plot the results for $n = 100$ (resp., $n = 500, n = 1000, n = 2000$).

All figures demonstrate that even though the M-FAQ algorithm is unable to align the multiplex template to its correlated subgraph in the background when $c = 1$, this can often be overcome by considering $c > 1$. Moreover, Figures 3 and 4, show that the Centered Padding scheme achieves better matching accuracy between channels than the Naïve Padding scheme, and that the matching accuracy increases as the correlation increases for both padding schemes. Finally, we observe the unsurprising phenomena in Figure 4 that the matching accuracy decreases as the ratio between the template size and the background size decreases (indeed, larger backgrounds are harder to search!).

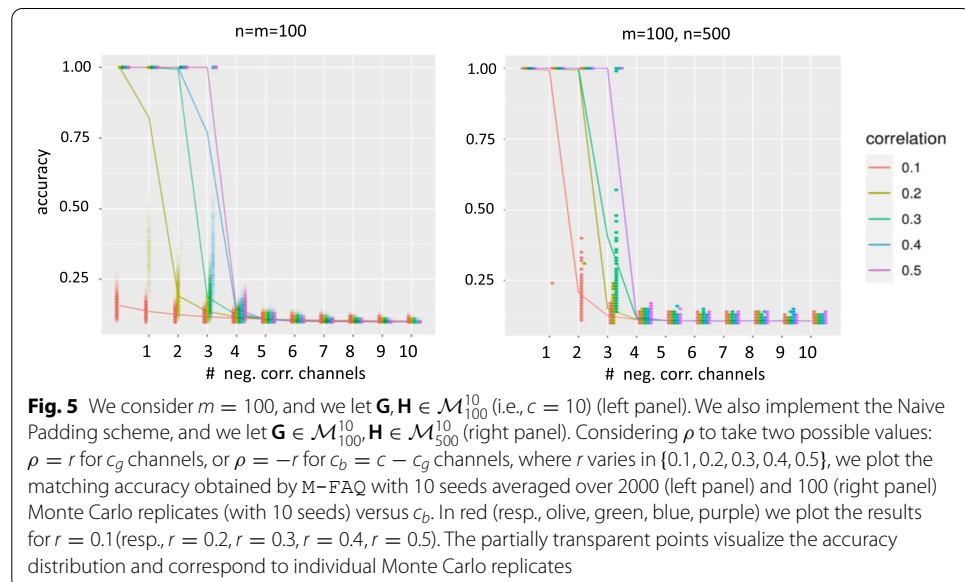
The good outweighs the bad

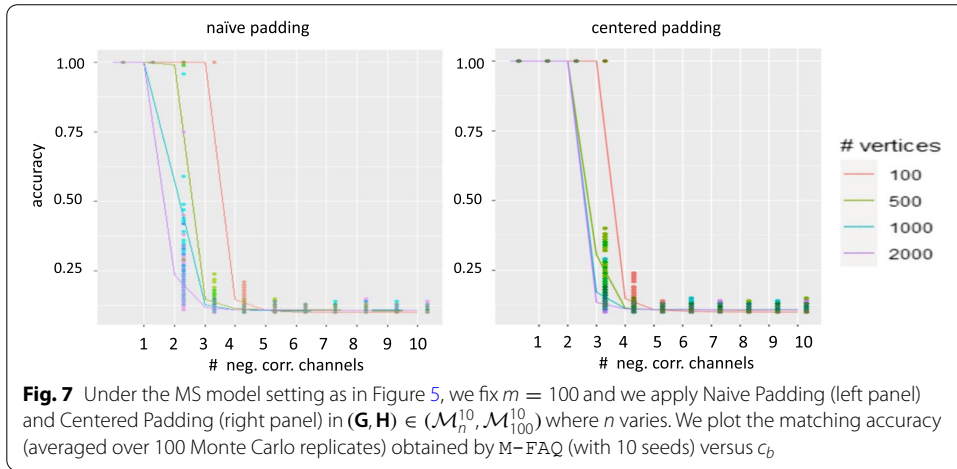
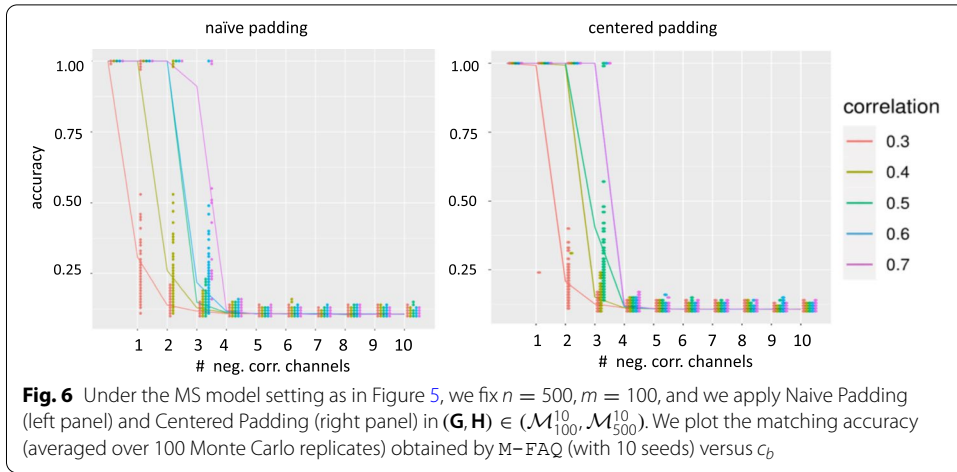
In this section, we explore the ability of the signal in “good” channels to overcome the obfuscating effect of “bad” channels. To wit, consider Condition (6) with $c_3 > 0$. We see that if there are enough channels (i.e., c_1 is sufficiently large) with positive correlation ($s_i, q_i < 1/2$), then the template and background remain matchable even in the presence of (potentially) multiple negatively-correlated channels. We explore this further in the following experiment. As in the previous “Strength in numbers” section, we study the obfuscating effect of negatively-correlated channels for both the $m = n$ and $m < n$ cases.

In Figure 5, we first consider $n = m = 100$ (left panel), and let $\mathbf{G}, \mathbf{H} \in \mathcal{M}_{100}^{10}$ (i.e., $c = 10$), where for $i \in [10]$ we have that $(G_i, H_i) \sim \text{ER}(100, 0.5, \rho)$. Under the same setting (right panel), we let $m = 100$, $n = 500$ and we apply Naive Padding to $(\mathbf{G}, \mathbf{H}) \in (\mathcal{M}_{100}^{10}, \mathcal{M}_{500}^{10})$.

Considering ρ to be either $\rho = r$ (for c_g channels) or $\rho = -r$ (for $c_b = c - c_g$ channels), where r varies in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, we plot the matching accuracy (averaged over 2000 (left panel) and 100 (right panel) Monte Carlo replicates) obtained by M-FAQ (with 10 seeds) versus c_b in Figure 5. For each choice of parameters, we also plot (via the partially transparent points) the accuracy achieved by each MC replicate. In red (resp., olive, green, blue, purple) we plot the results for $r = 0.1$ (resp., $r = 0.2, r = 0.3, r = 0.4, r = 0.5$). From the figure, we see the expected relationship: matching at higher levels of ρ yields better accuracy, and more robustness to channels with negative correlation. Further, we notice that the matching accuracy in the right panel (i.e., $m < n$) is not as good as in the left panel (i.e., $m = n$). To see whether this is universal, or merely an artifact of the padding scheme, we explore the $m < n$ case further in Figures 6 and 7.

To study the effect of different padding schemes (Naive vs Centered) in terms of the matching accuracy. We analyze the padding schemes’ effectiveness first, by varying the values of the correlation $r \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ while keeping n, m constant (see Figure 6) and second, by varying the number of the background vertices





$n \in \{100, 500, 1000, 2000\}$ while the template size $m = 100$ remains the same (see Figure 7). Using the Naive (resp. Centered) padding scheme, we consider matching $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{H}}$ (resp. $\hat{\mathbf{G}}$ and $\hat{\mathbf{H}}$) for c_b ranging over $\{1, 2, \dots, 10\}$. Utilizing $s = 10$ seeds, we match using M-FAQ; results are plotted in Figures 6 and 7. As in Figure 5, we plot the mean matching accuracy (i.e., the fraction of vertices whose latent alignment is recovered correctly) of M-FAQ versus c_b , averaged over 100 MC replicates. For each choice of parameters, we also plot (via the partially transparent points) the accuracy distribution corresponding to the MC replicates. In Figure 6, in red (resp., olive, green, blue, purple) we plot the results for $r = 0.3$ (resp., $r = 0.4$, $r = 0.5$, $r = 0.6$, $r = 0.7$). In Figure 7, in red (resp., green, blue, purple) we plot the results for $n = 100$ (resp., $n = 500$, $n = 1000$, $n = 2000$). From Figures 5 and 6, we observe that matching at higher levels of ρ yields better accuracy, and more robustness to channels with negative correlation even though the channels with negative correlation are more anti-correlated for higher r . Moreover, Figures 6 and 7 show that the Centered Padding scheme achieves better performance in terms of matching accuracy than the Naive Padding scheme, though the improvement is not as dramatic as that previously seen.

Experiments

Our previous simulation explored the effect on multiple channels on multiplex matchability. We next consider the performance of our multiplex matched filter approach in detecting a hidden template in a multilayer social media network from Magnani and Rossi (2011). The background network contains 3 aligned channels representing user activity in FriendFeed, Twitter and Youtube (where the Youtube and Twitter channels were generated via FriendFeed which aggregates user information across these platforms). In total, there are 6, 407 unique vertices across the three channels, with the channel specific networks satisfying:

Channel	Vertices	Edges
FriendFeed	5,540	31,921
Twitter	5,702	42,327
YouTube	663	614

Given a 35 vertex multiplex template \mathbf{g} (embedded into the background) created by Pacific Northwest National Laboratories for the DARPA MAA program, we ran our M-GMMF algorithm to attempt to recover the embedded template in \mathbf{h} ; results are summarized below.

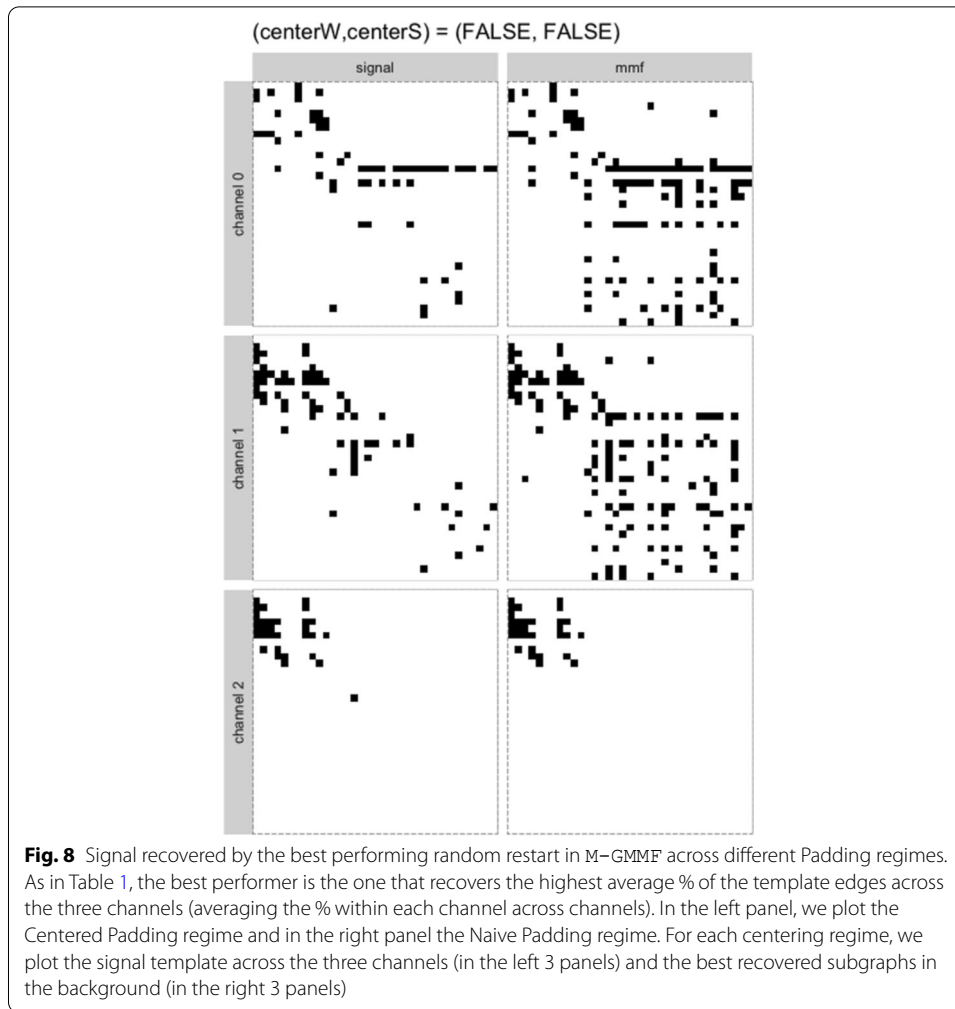
In our first experiment, we first considered running “cold-start” M-GMMF; that is, no prior information (in the form of seeds, hard or soft) is utilized in the algorithm. We consider padding the graph via the Naive Padding and Centered Padding regimes of “The multiplex graph matching distance” section, and for each padding regime, we ran M-GMMF with $N = 100$ random restarts. Numeric results are summarized in Table 1 (with the best recovered background signals also plotted in Figure 8). While the best recovered signal in the Naive Padding regime captures all but two template edges, this is at the expense of many extraneous background edges that do not appear in the template. On the other hand, the Centered Padding regime recovers most of the template edges (across the three channels) with minimal extra template edges in the recovered signal.

The M-FAQ algorithmic primitive used in our implementation of M-GMMF is most effective when it can leverage a priori available matching data in the form of *seeded* vertices. Seeds can either come in the form of *hard seeds* (a priori known 1-to-1 matches; here that would translate to template vertices whose exact match is known in the background) or *soft seeds* (where a soft seeded vertex v in \mathbf{g} has an a priori known distribution over possible matches in \mathbf{h} ; here this would translate into template vertices with

Table 1 For each padding regime, we provide the % of template edges present in the recovered background signal in the best random restart

Padding regime	% recovered in ch. 1	% recovered in ch. 2	% recovered in ch. 3
Centered	86.67	85.07	96.77
Naive	98.33	100	96.77

For example, the best recovered background signal in the Centered Padding regime recovered 86.67% of the edges in template channel 1, and 85.07% of the edges in template channel 2, and 96.77% of the edges in template channel 3. Here, the best performer is the one that recovers the highest average % across the three channels (averaging the % within each channel across channels)



a list of candidate matches in the background). While hard seeds are costly and often unavailable in practice, there are many scalable procedures in the literature for automatically generating soft seed matches. Here, we use as a soft-seeding the output of Moorman et al. (2018, 2021), a filtering approach for finding all subgraphs of the background network homomorphic to the template.

For each node in the template, the output of Moorman et al. (2018, 2021) produces a multiset of candidate matches in the background, where each candidate match corresponds to a template copy contained in the background as a subgraph (not necessarily as an induced subgraph). We convert the candidate matches into probabilities by simply converting the multiset to a count vector and normalizing the count vector to sum to 1. We then consider the normalized count vectors as rows of a stochastic matrix; this stochastic matrix provides M-FAQ with a soft-seeding/initialization which can be used to initialize the algorithm. Considering random restarts as perturbations (akin to Step 2.i of M-GMMF as presented in “[Multiplex graph matching matched filters](#)” section) of the soft-seeding (conditioned on retaining nonnegative entries), we ran M-GMMF using a generalization of the Centered Padding regime, which is defined as follows: For each $i \in [c]$, define the weighted adjacency matrices $\tilde{A}_i \in \mathbb{R}^{m \times m}$ and $\tilde{B}_i \in \mathbb{R}^{n \times n}$ via

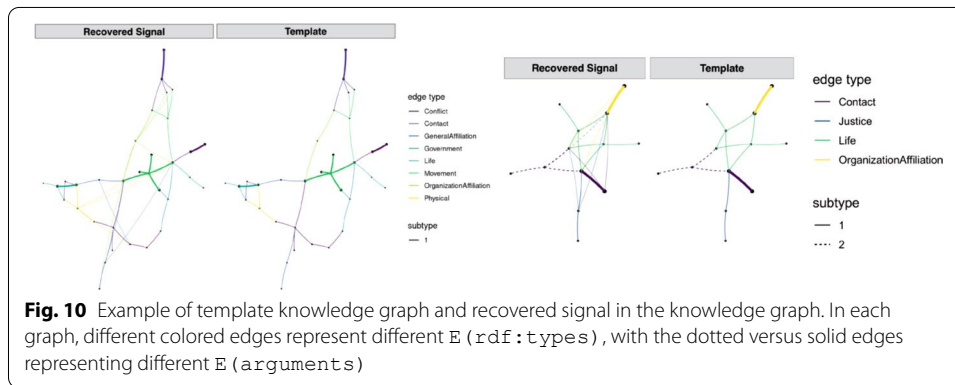
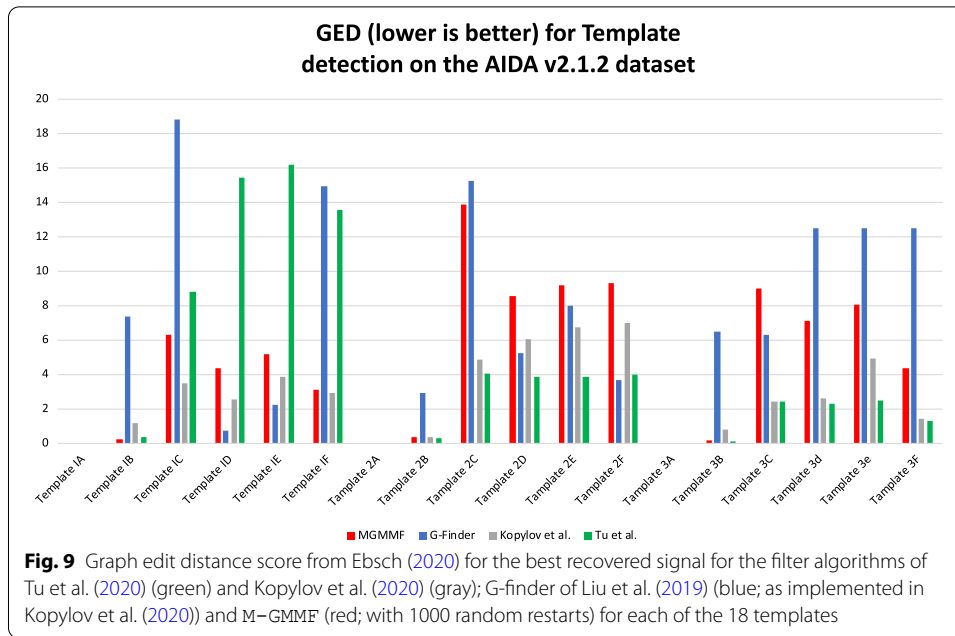
$$\begin{aligned}
\tilde{A}_i(u, v) &= \begin{cases} 1 & \text{if } u, v \in V(H_i), \text{ and } \{u, v\} \in E(H_i); \\ -w & \text{if } u, v \in V(H_i), \text{ and } \{u, v\} \notin E(H_i); \\ 0 & \text{if } u \text{ or } v \in [m] \setminus V(H_i); \end{cases} \\
\tilde{B}_i(u, v) &= \begin{cases} 1 & \text{if } u, v \in V(G_i), \text{ and } \{u, v\} \in E(G_i); \\ -1 & \text{if } u, v \in V(G_i), \text{ and } \{u, v\} \notin E(G_i); \\ 0 & \text{if } u \text{ or } v \in [n] \setminus V(G_i); \end{cases}
\end{aligned} \tag{11}$$

where we vary w from 0 to 1. Note that $w = 0$ yields Naive Padding, and $w = 1$ yields Centered Padding. Optimal performance in the present experiment was achieved with $w = 0.25$, in which case $N = 4000$ random restarts yielded an induced subgraph in the background that was exactly isomorphic to the template network.

M-GMMF on semantic property graphs

For our second example, we consider the semantic property graph released by Pacific Northwest National Laboratories as part of the MAA-AIDA Data Release V2.1.2 via the DARPA MAA program (Ebsch 2020). In this dataset, the background network is a knowledge graph constructed from a variety of documents (e.g., newspaper articles) by DARPA's AIDA program. At a high level, the graph is encoding the real-world relationships between a variety of entities (people, locations, major events, etc.) that can be automatically extracted from a variety of data sources. Practically, the graph is a richly featured network on the order of 100K nodes. Node properties/features include entity name; `rdf:type` (corresponding to a structured ontology of types defined by the AIDA ontology); `textValue`; `linkTarget`; start time; among others. Edge properties/features include edge name/id; `rdf:type`; argument (values given to edges of a given `rdf:type`); among others. Note that many nodes and edges do not have values for all properties. The provided templates, themselves richly featured knowledge graphs, are of the order of 10s of vertices (ranging in size from 33 nodes/40 edges to 11 nodes/11 edges); for each of three template types, there are 6 variants with varying error levels, including one variant (version "A") that is perfectly/isomorphically embedded into the background.

The principle challenge in applying our M-GMMF methodology on such richly featured data is sensibly incorporating the rich, structured features into our multiplex network framework. Towards this end, we adopted the following approach. We incorporated the vertex features/properties into a penalty term in the objective function, encoding the features into a vertex-to-vertex similarity matrix S (this is possible provided that similarities are easily computed within each vertex covariate, which is the case here). Edge features were used to divide the knowledge graph into multiple overlapping channels in a multiplex network. One channel was assigned to each unique (`E(rdf:type)`, `E(argument)`) pair in the template, and we divided background edges amongst the channels via a measure of similarity between the edge's (`E(rdf:type)`, `E(argument)`) pair and that of the channel. Note that this yields many, often overlapping, channels in the multiplex graph. Spatiotemporal constraints can be coded into separate channels in the multiplex graph, one channel per spatiotemporal constraint in the template. Each constraint (e.g., action A must occur between x and y days after action B) yields an edge filter, with only edges that could potentially satisfy the constraint being added to that constraint channel. Lastly, numeric edge features are used to further



weight the edges in the background/template. The final objective function we used in our M-GMMF approach was then of the form

$$\arg \max_{P \in \Pi_n} \sum_{i=1}^c \text{tr}(\hat{A}_i \oplus \mathbf{0}_{n-m}) P \hat{B}_i P^T + \text{tr}(S P^T). \quad (12)$$

Performance of M-GMMF and other competing approaches (as scored by the GED scoring metric of Ebsch 2020) are presented in Figure 9 (Note that the filtering approach of Tu et al., as presented in Tu et al. (2020), also present a method that uses the clean “A” template for training, and achieves essentially the best score for all template versions; we did not include those scores for comparison in our figure).

Overall, the challenge was to detect noisy matches to each of 18 provided templates: 3 distinct template types, each with 6 variants encompassing different amounts of template noise; see Figure 10, for a pair of example templates and M-GMMF recovered signals here. One template of each type was constructed to have a perfect isomorphic match in

the background, while the noisy templates were designed for inexact/fuzzy/noisy matching. All approaches identified the isomorphic match in the background for all three template types version “A”, while our approach achieved its best relative results on the larger, more complex template (Template 1). For Template 1, we obtained the best (or nearly the best) score on 3/6 versions; see Table 2 for detail.

Templates 2 and 3 were essentially tree-like structures (nodes/edges is 13/15 and 11/11 respectively), and we suspect that the filtering-based approaches of Tu et al. (2020) and Kopylov et al. (2020) are more suitable to this problem type. Indeed, M-GMMF is designed for larger/more complex templates, though our performance (especially compared to the non-filtering G-Finder of Liu et al. 2019) is encouraging on these instantiations, especially on version “B” of the templates.

Discussion

In this paper, we presented a framework for finding noisy copies of a multiplex template in a large multiplex background network. Our strategy, which extends (Sussman et al. 2019) to the multiplex setting, uses a novel formulation of multiplex graph matching combined with multiple random restarts to search the background for locally optimal matches to the template. The effectiveness of the resulting algorithm, named M-GMMF, is further demonstrated both theoretically (in a novel multiplex correlated network framework) and empirically on both real and simulated data sets. To formalize the M-GMMF approach, we provide a very natural extension of the classical graph matching problem to multiplex networks that is easily amended to matching graphs of different orders (both across networks and channels). We also extend the Frank-Wolfe based FAQ algorithm of Vogelstein et al. (2014) to the multiplex setting.

There are a number of extensions and open questions that arose during the course of this work. Natural theoretic extensions include lifting Theorems 1 and 2 to non-edge independent models (note that certain localized dependencies amongst edges can easily be handled in the McDiarmin proof framework, while globally dependent errors provide a more significant challenge); formulating the analogues of Theorems 1 and 2 in the weighted, attributed graph settings; and considering the theoretic properties of various continuous relaxations of the multiplex GM problem akin to Aflalo et al. (2015), Lyzinski et al. (2015), Bento and Ioannidis (2019). A key methodological question in multiplex graph matching was touched upon in Remark 3; indeed, we expect the question of how to weight the matching across channels to be essential when applying these methods to topologically diverse and weighted networks. If the order of magnitude of edge weights vary across channel, then it is easy to see a GM algorithm aligning channels with large edge weights at the expense of the alignment accuracy in other channels. Judiciously

Table 2 Performance on Template 1

Method	T1A	T1B	T1C	T1D	T1E	T1F
M-GMMF	0	0.24	6.28	4.35	5.17	3.12
G-Finder	0	7.38	18.81	0.76	2.25	14.96
2020)	0	0.347	8.783	15.47	19.194	13.596
2020)	0	1.17	3.47	2.58	3.85	2.94

choosing (λ_i) would allow for the signal in channels with smaller edge weights to be better leveraged towards a better overall matching.

While the largest network we consider in this work has $\approx 100,000$ vertices, scaling this approach to very large networks is essential. By utilizing efficient data structures for sparse, low-rank matrices and a clever implementation of the LAP subroutine of $M\text{-FAQ}$ (step 2.ii. in the presented $M\text{-GMMF}$ algorithm), we are able to match $O(10)$ vertex templates to $20K$ -vertex background graphs in < 10 s per restart with our base $M\text{-GMMF}$ code (available in `iGraphMatch`) implemented in \mathbb{R} on a standard laptop. Further work to scale $M\text{-GMMF}$ by leveraging both efficient data structures and scalable approximate LAP solvers is currently underway.

Appendix

Herein we collect details of our auxiliary algorithms and proofs of our main results.

Proof of Theorem 1

For each $m < n$, denote the set of permutations in Π_n that permute exactly k labels of $[m]$ by $\Pi_{n,m,k}$. For each $P \in \Pi_n$ (with associated permutation σ_P), define

$$\Delta_P = \left\{ \{i, j\} \in \binom{[m]}{2} \text{ s.t. } \{i, j\} \neq \{\sigma_P(i), \sigma_P(j)\} \right\}, \quad (13)$$

and for each $i \in [c]$, define

$$\begin{aligned} \Delta_P^{(i,1)} &= \{ \{j, \ell\} \in \Delta_P \text{ s.t. } 0 \neq \widehat{C}_i(j, \ell) \neq \widehat{D}_i(\sigma_P(j), \sigma_P(\ell)) \neq 0 \}; \\ \Delta_P^{(i,2)} &= \{ \{j, \ell\} \in \Delta_P \text{ s.t. } 0 \neq \widehat{C}_i(j, \ell) \neq \widehat{D}_i(\sigma_P(j), \sigma_P(\ell)) = 0 \}. \end{aligned}$$

Let the set of the c_1 low-noise channels ($q < 1/2$) be denoted \mathcal{C}_1 , and the set of the c_3 high-noise channels ($q > 1/2$) be denoted \mathcal{C}_3 . If there exists an $n_1 > 0$ such that for all $n > n_1$, we have that for all $k \in \{1, 2, 3, \dots, m(n)\}$ and all $P \in \Pi_{n,m,k}$,

$$\sum_{i \in \mathcal{C}_1 \cup \mathcal{C}_3} \left(2|\Delta_P^{(i,1)}| + |\Delta_P^{(i,2)}| \right) (1 - 2s_i)(1 - 2q_i) \geq k \sqrt{\frac{672m^{1+\alpha}c}{\beta}}, \quad (14)$$

then letting $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{B}}$ be the padded, centered adjacency matrices of \mathbf{G} and \mathbf{H} respectively (the errorful \mathbf{g} and \mathbf{h}), for $n > n = \max(n_0, n_1)$ we have that

$$\mathbb{P} \left(\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|(\widehat{\mathbf{A}}_i \oplus \mathbf{0}_{n-m})P - P\widehat{\mathbf{B}}_i\|_F^2 \notin \mathcal{P}_{m,n} \right) \leq 2n^{-2}, \quad (15)$$

(see “Appendix 1.1.1” for proof of this bound).

While we expect Eq. (14) to hold in more general settings, below we show it holds with high probability in the present Erdős-Rényi setting with our given theorem assumptions. Here, $\Delta_P^{(i,2)}$ is empty, and a simple application of McDiarmid’s inequality yields that for a fixed $P \in \Pi_{n,m,k}$, we have

$$|\Delta_P^{(i,1)}| \in (|\Delta_P|p(1-p), 3|\Delta_P|p(1-p)),$$

with probability at least

$$1 - 2\exp\left\{\frac{-2|\Delta_P|p^2(1-p)^2}{8}\right\}. \quad (16)$$

Indeed, if each $h_i \sim \text{ER}(n, p)$, then

$$\Delta_P^{(i,1)} = \sum_{\{j,\ell\} \in \Delta_P} \mathbb{1}\{\widehat{C}_i(j, \ell) \neq \widehat{D}_i(\sigma_p(j), \sigma_p(\ell))\},$$

so that $\mathbb{E}(\Delta_P^{(i,1)}) = 2p(1-p)|\Delta_P|$. Also, $\Delta_P^{(i,1)}$ is then a function of at most $2|\Delta_P|$ independent Bernoulli random variables, and changing the value of any one these can change the value of $\Delta_P^{(i,1)}$ by at most 2. McDiarmid's inequality then yields the desired result

$$\mathbb{P}(|\Delta_P^{(i,1)} - \mathbb{E}(\Delta_P^{(i,1)})| \geq t) \leq 2\exp\left\{-\frac{2t^2}{8|\Delta_P|}\right\}, \quad (17)$$

by setting $t = p(1-p)|\Delta_P|$.

Note that if $m > 6$, then as $P \in \Pi_{n,m,k}$, $mk/3 \leq |\Delta_P| \leq mk$. Therefore, if $m > 6$ then for n sufficiently large, we have that with probability at least (letting ξ in Assumption (5) be appropriately defined)

$$1 - 2\exp\left\{\frac{-2mkp^2(1-p)^2}{24}\right\} \geq 1 - 2\exp\{-8k \log(n)\}. \quad (18)$$

the event

$$\{|\Delta_P^{(i,1)}| \in [mkp(1-p)/3, 3mkp(1-p)]\}$$

occurs. Now, we have that

$$\begin{aligned} \sum_{i \in \mathcal{C}_1 \cup \mathcal{C}_3} \left(2|\Delta_P^{(i,1)}| + |\Delta_P^{(i,2)}|\right)(1-2s_i)(1-2q_i) &> \frac{4}{3}mkp(1-p)(1/2-s)(1/2-q)(c_1-c_3) \\ &\geq \frac{4}{3}mk(1-p)\gamma\sqrt{m^{\alpha-1}c} \\ &= \frac{4}{3}k(1-p)\gamma\sqrt{m^{1+\alpha}c} \end{aligned}$$

where the second inequality holds by assumption. We see then that Eq. (14) holds with high probability with an appropriate choice of γ (independent of k).

To finish the proof, we proceed as follows. For each $P \in \Pi_n$ define

$$X_P := \frac{1}{4} \left[\sum_{i=1}^c \|(\widehat{A}_i \oplus \mathbf{0}_{n-m})P - P\widehat{B}_i\|_F^2 - \sum_{i=1}^c \|(\widehat{A}_i \oplus \mathbf{0}_{n-m}) - \widehat{B}_i\|_F^2 \right] \quad (19)$$

Let the equivalence relation " \sim " on $\Pi_{n,m,k}$ be defined via $P \sim Q$ if there exists a $U \in \mathcal{P}_{n,m}$ such that $PU = Q$. Note that if $P \sim Q$ then

$$\sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m})P - P\hat{B}_i\|_F^2 = \sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m})Q - Q\hat{B}_i\|_F^2,$$

and so $X_P = X_Q$. Let $\Pi_{n,m,k}^*$ be a fixed (but arbitrarily chosen) set composed of one member of each equivalence class according to “ \sim ”, and note that $|\Pi_{n,m,k}^*|$ is at most $m^{2k} n^{2k}$. From Eq. (18), for $n > n_2$ we have that for each $P \in \Pi_{n,m,k}^*$,

$$\begin{aligned} & \mathbb{P}\left(\bigcup_{i=1}^c \left\{|\Delta_P^{(i,1)}| \notin [mkp(1-p)/3, 3mkp(1-p)]\right\}\right) \\ & \leq 2\exp\{-8k \log n + \log c\} \leq 2\exp\{-7k \log n\}. \end{aligned} \quad (20)$$

Denote the event bound in Eq. (20) via $\mathcal{E}_{n,P}$. For $n > \max(n_2, n_0)$, we then have

$$\begin{aligned} & \mathbb{P}\left(\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m})P - P\hat{B}_i\|_F^2 \notin \mathcal{P}_{m,n}\right) = \mathbb{P}(\exists P \notin \mathcal{P}_{m,n} \text{ s.t. } X_P \leq 0) \\ & \leq \sum_{k=1}^m \sum_{P \in \Pi_{n,m,k}^*} \mathbb{P}(X_P \leq 0) \\ & \leq \sum_{k=1}^m \sum_{P \in \Pi_{n,m,k}^*} \mathbb{P}(|X_P - \mathbb{E}(X_P)| \geq \mathbb{E}(X_P) | \varepsilon_{n,P}^c) + \mathbb{P}(\varepsilon_{n,P}) \\ & \leq \underbrace{\sum_{k=1}^m \sum_{P \in \Pi_{n,m,k}^*} \mathbb{P}(|X_P - \mathbb{E}(X_P)| \geq \mathbb{E}(X_P) | \varepsilon_{n,P}^c)}_{\leq 2n^{-2} \text{ by the proof in App. 1.1.1}} + 2 \sum_{k=1}^m \exp(-7k \log n + 2k \log n + 2k \log m) \\ & \leq 4n^{-2}, \end{aligned}$$

as desired.

Proof of Eq. (15)

Recall, for each $P \in \Pi_n$, we defined

$$\begin{aligned} X_P &:= \frac{1}{4} \left[\sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m}) - P\hat{B}_i P^T\|_F^2 - \sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m}) - \hat{B}_i\|_F^2 \right] \\ &= \sum_{i=1}^c \frac{1}{2} \left(\text{tr}((\hat{A}_i \oplus \mathbf{0}_{n-m})\hat{B}_i) - \text{tr}((\hat{A}_i \oplus \mathbf{0}_{n-m})P\hat{B}_i P^T) \right) \\ &= \sum_{i=1}^c \sum_{\{j,\ell\} \in \Delta_P} \hat{A}_i(j, \ell) \left[\hat{B}_i(j, \ell) - \hat{B}_i(\sigma_P(j), \sigma_P(\ell)) \right] \end{aligned}$$

Assuming that $P \in \Pi_{n,m,k}$, then $|\Delta_P| \leq mk$. Note that X_P is a function of (at most) $3c|\Delta_P|$ independent Bernoulli random variables, and changing any one of these Bernoulli random variables can change the value of X_P by at most 8. McDiarmid’s inequality (McDiarmid 1989) then implies that for any $t \geq 0$,

$$\mathbb{P}(|X_P - \mathbb{E}(X_P)| \geq t) \leq 2\exp\left\{-\frac{2t^2}{192cmk}\right\}. \quad (21)$$

Note that if $\hat{C}_i(j, k), \hat{D}_i(j, k), \hat{D}_i(\sigma_P(j), \sigma_P(k)) \in \{1, -1\}$ then

$$\begin{aligned}\mathbb{E} \hat{A}_i(j, \ell) \hat{B}_i(j, \ell) &= \hat{C}_i(j, \ell) \hat{D}_i(j, \ell) (1 - 2s_i)(1 - 2q_i) = \hat{C}_i(j, \ell)^2 (1 - 2s_i)(1 - 2q_i) \\ \mathbb{E} \hat{A}_i(j, \ell) \hat{B}_i(\sigma_p(j), \sigma_p(\ell)) &= \hat{C}_i(j, \ell) \hat{D}_i(\sigma_p(j), \sigma_p(\ell)) (1 - 2s_i)(1 - 2q_i).\end{aligned}$$

Define

$$\begin{aligned}\Delta_p^{(i,0)} &= \{ \{j, \ell\} \in \Delta_p \text{ s.t. } \hat{C}_i(j, \ell) \neq 0 \}; \\ \Delta_p^{(i,1)} &= \{ \{j, \ell\} \in \Delta_p^{(i,0)} \text{ s.t. } 0 \neq \hat{C}_i(j, \ell) \neq \hat{D}_i(\sigma_p(j), \sigma_p(\ell)) \neq 0 \}; \\ \Delta_p^{(i,2)} &= \{ \{j, \ell\} \in \Delta_p^{(i,0)} \text{ s.t. } 0 \neq \hat{C}_i(j, \ell) \neq \hat{D}_i(\sigma_p(j), \sigma_p(\ell)) = 0 \}; \\ \Delta_p^{(i,3)} &= \{ \{j, \ell\} \in \Delta_p^{(i,0)} \text{ s.t. } 0 \neq \hat{C}_i(j, \ell) = \hat{D}_i(\sigma_p(j), \sigma_p(\ell)) \};\end{aligned}$$

so that $|\Delta_p^{(i,0)}| = |\Delta_p^{(i,1)}| + |\Delta_p^{(i,2)}| + |\Delta_p^{(i,3)}|$. We then have

$$\begin{aligned}\mathbb{E}(X_p) &= \sum_{i=1}^c \left[|\Delta_p^{(i,0)}| (1 - 2s_i)(1 - 2q_i) - |\Delta_p^{(i,3)}| (1 - 2s_i)(1 - 2q_i) \right. \\ &\quad \left. + |\Delta_p^{(i,1)}| (1 - 2s_i)(1 - 2q_i) \right] \\ &= \sum_{i=1}^c \left(2|\Delta_p^{(i,1)}| + |\Delta_p^{(i,2)}| \right) (1 - 2s_i)(1 - 2q_i).\end{aligned}\tag{22}$$

Note that if $P, Q \in \Pi_{n,m,k}$, then $X_P = X_Q$ if $\sigma_p(j) = \sigma_q(j)$ for all $j \in [m]$; i.e., if there exists a $U \in \mathcal{P}_{m,n}$ such that $PU = Q$. Note that this defines an equivalence relation on $P, Q \in \Pi_{n,m,k}$ which we will denote by “ \sim ,” and let $\Pi_{n,m,k}^*$ be a fixed (but arbitrarily chosen) set composed of one member of each equivalence class according to “ \sim .” Note that $|\Pi_{n,m,k}^*|$ is at most $m^{2k}n^{2k}$. Letting $t = \mathbb{E}(X_p)$ in Eq. (21), we have that if $n > n = \max(n_0, n_1)$

$$\begin{aligned}\mathbb{P}(\exists P \notin \mathcal{P}_{m,n} \text{ s.t. } X_P \leq 0) &\leq \sum_{k=1}^m \sum_{P \in \Pi_{n,m,k}^*} \mathbb{P}(X_P \leq 0) \\ &\leq \sum_{k=1}^m \sum_{P \in \Pi_{n,m,k}^*} \mathbb{P}(|X_P - \mathbb{E}(X_P)| \geq \mathbb{E}(X_P)) \\ &\leq \sum_{k=1}^m \sum_{P \in \Pi_{n,m,k}^*} 2\exp\left\{-\frac{1344m^{1+\alpha}ck^2}{192\beta cmk}\right\} \\ &\leq \sum_{k=1}^m \sum_{P \in \Pi_{n,m,k}^*} \exp\left\{-\frac{7m^\alpha k}{\beta}\right\} \\ &\leq \sum_{k=1}^m 2\exp\{-7k \log n + 2k \log n + 2k \log m\} \\ &\leq 2\exp\{-2 \log n\}\end{aligned}$$

as desired.

Proof of Theorem 2

For each $P \in \Pi(n)$, define

$$\begin{aligned}\Delta_P^{(1)} &:= \{ \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = 1; H(\sigma_P(j), \sigma_P(\ell)) = 0 \}; \\ \Delta_P^{(2)} &:= \{ \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = 0; H(\sigma_P(j), \sigma_P(\ell)) = 1 \},\end{aligned}$$

where Δ_P is defined as in Eq. (13). Suppose that there exists an $n_1 > 0$ such that for all $n > n_1$, we have that for all $k \in [m = m(n)]$ and all $P \in \Pi_{n,m,k}$ (where γ is an appropriately chosen constant)

$$|\Delta_P^{(1)}| \sum_i 2(1 - 2s_i)(1 - r_i - t_i) + |\Delta_P^{(2)}| \sum_i 2(1 - 2q_i)(1 - r_i - t_i) \geq k \sqrt{\frac{\gamma m^{1+\alpha} c}{\beta}}, \quad (23)$$

then

$$\mathbb{P} \left(\arg \min_{P \in \Pi_n} \sum_{i=1}^c \|(\hat{A}_i \oplus \mathbf{0}_{n-m})P - P\hat{B}_i\|_F^2 \notin \mathcal{P}_{m,n} \right) = 2n^{-2}, \quad (24)$$

where the proof of the bound in Eq. (24) uses “Appendix 1.3” and then the proof follows mutatis mutandis from the proof in “Appendix 1.1.1”.

In the present ER setting, as in Eq. (16), we then have that for $j = 1, 2$,

$$|\Delta_P^{(j)}| \in \left(\frac{1}{2} |\Delta_P| p(1-p), \frac{3}{2} |\Delta_P| p(1-p) \right),$$

with probability at least

$$1 - 2 \exp \left\{ \frac{-2 |\Delta_P| p^2 (1-p)^2}{32} \right\} \quad (25)$$

Note that if $m > 6$, then $mk/3 \leq |\Delta_P| \leq mk$, so that with probability at least $1 - 4e^{-8 \log n}$ (this lower bounds Eq. (25) for judiciously chosen ξ in Eq. (7),

$$|\Delta_P^{(j)}| \in (1/6, 3/2) \cdot m k p(1-p).$$

for both $j = 1, 2$. To show Eq. (23) holds here with sufficiently high probability, note that the above bounds yield

$$\begin{aligned} & \left| \Delta_P^{(1)} \right| \sum_i 2(1 - 2s_i)(1 - r_i - t_i) + \left| \Delta_P^{(2)} \right| \sum_i 2(1 - 2q_i)(1 - r_i - t_i) \\ & \geq \frac{2}{3} m k p(1-p) \sum_i (1 - s_i - q_i)(1 - r_i - t_i) \\ & = \frac{2}{3} m k p(1-p) (c_3 + c_4 - c_1 - c_2) e_1 e_2 \\ & \geq \frac{2\gamma}{3} k(1-p) \sqrt{m^{1+\alpha} c} \end{aligned}$$

where the last inequality follows from Eq. (8). Proving Eq. (9) then follows mutatis mutandis to the proof of the analogue in Theorem 1 (which is spelled out in detail in “Appendix 1.1”), and details are omitted.

Proof details for Eq. (24)

For $P \in \Pi_n$, define

$$\begin{aligned}\Delta_P^{(1)} &:= \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = 1; H(\sigma_P(j), \sigma_P(\ell)) = 0\}; \\ \Delta_P^{(2)} &:= \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = 0; H(\sigma_P(j), \sigma_P(\ell)) = 1\}; \\ \Delta_P^{(3)} &:= \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = H(\sigma_P(j), \sigma_P(\ell)) = 1\}; \\ \Delta_P^{(4)} &:= \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = H(\sigma_P(j), \sigma_P(\ell)) = 0\}; \\ e_P &:= \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = 1\}; \\ n_P &:= \{j, \ell\} \in \Delta_P \text{ s.t. } G(j, \ell) = 0\}.\end{aligned}$$

For $P \in \Pi_{n,m,k}$, we then have that X_P defined in Eq. (19) satisfies

$$\begin{aligned}\mathbb{E}(X_P) &= (|e_P| - |\Delta_P^{(3)}|) \sum_i (1 - 2s_i)(1 - 2r_i) + (|n_P| - |\Delta_P^{(4)}|) \sum_i (1 - 2q_i)(1 - 2t_i) \\ &\quad + |\Delta_P^{(1)}| \sum_i (1 - 2s_i)(1 - 2t_i) + |\Delta_P^{(2)}| \sum_i (1 - 2q_i)(1 - 2r_i) \\ &= |\Delta_P^{(1)}| \sum_i 2(1 - 2s_i)(1 - r_i - t_i) + |\Delta_P^{(2)}| \sum_i 2(1 - 2q_i)(1 - r_i - t_i).\end{aligned}$$

Showing that this yields the desired bound combines Eq. (23) with the (nearly identical) proof technique of “Appendix 1.1.1”; details are omitted.

Abbreviations

GM(P): Graph Matching (Problem); nMGMP: Naive Multiplex Graph Matching Problem; cMGMP: Centered Multiplex Graph Matching Problem; M-GMMF: Multiplex Graph Matching Matched Filters; M-FAQ: Multiplex Fast Approximate Quadratic graph matching algorithm; ME: (Single Channel Source) Multiplex Error; MS: (Single Channel Errors) Multiplex Source; GED: Graph Edit Distance; LAP: Linear Assignment Program; DARPA: Defense Advanced Research Projects Agency; MAA: Modeling Adversarial Activity DARPA program; AIDA: Active Interpretation of Disparate Alternatives DARPA program.

Acknowledgements

Not applicable.

Authors contributions

Daniel L. Sussman, Carey E. Priebe and Vince Lyzinski, and Konstantinos Pantazis conceived of the method and developed the theory. Konstantinos Pantazis, Youngser Park, Daniel L. Sussman, Vince Lyzinski and Zhirui Li worked on writing and implementing the algorithm and performing relevant experiments. Konstantinos Pantazis, Daniel L. Sussman and Vince Lyzinski wrote and curated the manuscript. All authors read and approved the final manuscript.

Funding

Dr. Sussman's contribution to this work was partially supported by a grant from MIT Lincoln Labs and the Department of Defense. This material is based on research sponsored by the Air Force Research Laboratory and DARPA under agreement numbers FA8750-18-2-0035 and FA8750-20-2-1001. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and DARPA or the U.S. Government.

Availability of data and materials

The background graphs for the 3-channel social network in “Experiments” section is available at <http://multilayer.it.uu.se/datasets.html>. The data from the DARPA MAA analysis Section is not publicly available, and the obtained results for the outside algorithms appear in the cited papers in the literature. The code implementing the M-GMMF and M-FAQ procedures can be downloaded as part of our R package, `iGraphMatch`, which is available on CRAN or can be downloaded at <https://github.com/dpmcsuss/iGraphMatch>.

Declarations**Competing interests**

Not applicable.

Author details

¹Department of Mathematics, University of Maryland, College Park, College Park, USA. ²Department of Mathematics and Statistics, Boston University, Boston, USA. ³Center for Imaging Sciences, Johns Hopkins University, Baltimore, USA. ⁴Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, USA.

Received: 3 December 2021 Accepted: 17 April 2022

Published online: 13 May 2022

References

- Aflalo Y, Bronstein A, Kimmel R (2015) On convex relaxation of graph isomorphism. *Proc Natl Acad Sci* 112(10):2942–2947
- Alon N, Dao P, Hajirasouliha I, Hormozdiari F, Sahinalp SC (2008) Biomolecular network motif counting and discovery by color coding. *Bioinformatics* 24(13):i241–i249
- Alon N, Yuster R, Zwick U (1995) Color-coding. *J ACM (JACM)* 42(4):844–856
- Arroyo J, Sussman DL, Priebe CE, Lyzinski V (2018) Maximum likelihood estimation and graph matching in errorfully observed networks. *arXiv preprint* [arXiv:1812.10519](https://arxiv.org/abs/1812.10519)
- Barak B, Chou C, Lei Z, Schramm T, Sheng Y (2018) (nearly) efficient algorithms for the graph matching problem on correlated random graphs. *arXiv preprint* [arXiv:1805.02349](https://arxiv.org/abs/1805.02349)
- Bento J, Ioannidis S (2019) A family of tractable graph metrics. *Appl Netw Sci* 4(1):1–27
- Boccaletti S, Bianconi G, Criado R, Del Genio CI, Gómez-Gardenes J, Romance M, Sendina-Nadal I, Wang Z, Zanin M (2014) The structure and dynamics of multilayer networks. *Phys Rep* 544(1):1–122
- Chan SC, Cheney J (2020) Flexible graph matching and graph edit distance using answer set programming. In: *International symposium on practical aspects of declarative languages*, pp 20–36. Springer
- Chaudhuri S, Chatterjee S, Katz N, Nelson M, Goldbaum M (1989) Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Trans Med Imaging* 8(3):263–269
- Chen L, Vogelstein JT, Lyzinski V, Priebe CE (2016) A joint graph inference case study: the *C. elegans* chemical and electrical connectomes. *Worm* 5(2)
- Conte D, Foggia P, Sansone C, Vento M (2004) Thirty years of graph matching in pattern recognition. *Int J Pattern Recognit Artif Intell* 18(03):265–298
- Cordella LP, Foggia P, Sansone C, Vento M (2004) A (sub) graph isomorphism algorithm for matching large graphs. *IEEE Trans Pattern Anal Mach Intell* 26(10):1367–1372
- Cullina D, Kiyavash N (2016) Improved achievability and converse bounds for erdos-rényi graph matching. In: *ACM SIGMETRICS performance evaluation review*, 44(1), pp 63–72. ACM
- Cullina D, Kiyavash N (2017) Exact alignment recovery for correlated Erdos-Renyi graphs. *arXiv preprint* [arXiv:1711.06783](https://arxiv.org/abs/1711.06783)
- Cullina D, Kiyavash N, Mittal P, Poor HV (2018) Partial recovery of Erdos-Renyi graph alignment via *k*-core alignment. *arXiv preprint* [arXiv:1809.03553](https://arxiv.org/abs/1809.03553)
- DePiero F, Krout D (2003) An algorithm using length-*r* paths to approximate subgraph isomorphism. *Pattern Recognit Lett* 24(1–3):33–46
- Ding J, Ma Z, Wu Y, Xu J (2018) Efficient random graph matching via degree profiles. *arXiv preprint* [arXiv:1811.07821](https://arxiv.org/abs/1811.07821)
- Du B, Zhang S, Cao N, Tong H (2017) First: Fast interactive attributed subgraph matching. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1447–1456
- Du X, Yan J, Zhang R, Zha H (2020) Cross-network skip-gram embedding for joint network alignment and link prediction. *IEEE Trans Knowl Data Eng*
- Ebsch CL, Cottam JA, Heller NC, Deshmukh RD, Chin G (2020) Using graph edit distance for noisy subgraph matching of semantic property graphs. In: *2020 IEEE international conference on big data (big data)*, pp 2520–2525. IEEE
- Egozi A, Keller Y, Guterman H (2012) A probabilistic approach to spectral graph matching. *IEEE Trans Pattern Anal Mach Intell* 35(1):18–27
- Emmert-Streib F, Dehmer M, Shi Y (2016) Fifty years of graph matching, network alignment and network comparison. *Inf Sci* 346–347:180–197
- Fan Z, Mao C, Wu Y, Xu J (2020) Spectral graph matching and regularized quadratic relaxations: algorithm and theory. In: *International conference on machine learning*, pp 2985–2995. PMLR
- Fang F, Sussman DL, Lyzinski V (2018) Tractable graph matching via soft seeding. *arXiv preprint* [arXiv:1807.09299](https://arxiv.org/abs/1807.09299)
- Feizi S, Quon G, Medard M, Kellis M, Jadbabaie A (2015) Spectral alignment of networks
- Fishkind DE, Lyzinski V, Pao H, Chen L, Priebe CE (2015) Vertex nomination schemes for membership prediction. *Ann Appl Stat* 9(3):1510–1532
- Fishkind DE, Adali S, Patsolic HG, Meng L, Singh D, Lyzinski V, Priebe CE (2019) Seeded graph matching. *Pattern Recognit* 87:203–215
- Foggia P, Percannella G, Vento M (2014) Graph matching and learning in pattern recognition in the last 10 years. *Int J Pattern Recognit Artif Intell* 28(01):1450001
- Fomin FV, Lokshantov D, Raman V, Saurabh S, Rao BVR (2012) Faster algorithms for finding and counting subgraphs. *J Comput Syst Sci* 78(3):698–706
- Frank M, Wolfe P (1956) An algorithm for quadratic programming. *Nav Res Logist Q* 3(1–2):95–110
- Goga O, Loiseau P, Sommer R, Teixeira R, Gummadi KP (2015) On the reliability of profile matching across large online social networks. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1799–1808. ACM
- Heimann M, Shen H, Safavi T, Koutra D (2018) Regal: representation learning-based graph alignment. In: *Proceedings of the 27th ACM international conference on information and knowledge management*, pp 117–126
- Jonker R, Volgenant A (1987) A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* 38(4):325–340

- Kazemi E, Hassani SH, Grossglauser M (2015a) Growing a graph matching from a handful of seeds. *Proc VLDB Endow* 8(10):1010–1021
- Kazemi E, Yartseva L, Grossglauser M (2015b) When can two unlabeled networks be aligned under partial overlap? In: 2015 53rd Annual allerton conference on communication, control, and computing (Allerton), pp 33–42. IEEE
- Kiar G, Bridgeford EW, Roncal WRG, Chandrashekar V, Mhembere D, Ryman S, Zuo X, Margulies DS, Craddock RC, Priebe CE, Jung R, Calhoun VD, Caffo B, Burns R, Milham MP, Vogelstein JT (2018) A high-throughput pipeline identifies robust connectomes but troublesome variability. *bioRxiv*, p 188706
- Kivelä M, Arenas A, Barthélemy M, Gleeson JP, Moreno Y, Porter MA (2014) Multilayer networks. *J Complex Netw* 2(3):203–271
- Kivelä M, Porter MA (2017) Isomorphisms in multilayer networks. *IEEE Trans Netw Sci Eng*
- Klau GW (2009) A new graph-based method for pairwise global network alignment. *BMC Bioinf* 10(1):S59
- Kopylov A, Xu J (2019) Filtering strategies for inexact subgraph matching on noisy multiplex networks. In: 2019 IEEE international conference on big data (big data), pp 4906–4912. IEEE
- Kopylov A, Xu J, Ni K, Roach S, Lu T-C (2020) Semantic guided filtering strategy for best-effort subgraph matching in knowledge graphs. In: 2020 IEEE international conference on big data (big data), pp 2539–2545. IEEE
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25
- Larrosa J, Valiente G (2002) Constraint satisfaction algorithms for graph pattern matching. *Math Struct Comput Sci* 12(4):403–422
- Lee J, Han W, Kasperovics R, Lee J (2012) An in-depth comparison of subgraph isomorphism algorithms in graph databases. *Proc VLDB Endow* 6(2):133–144
- Liu L, Du B, Tong H (2019) G-finder: approximate attributed subgraph matching. In: 2019 IEEE international conference on big data (big data), pp 513–522. IEEE
- Lladós J, Martí E, Villanueva JJ (2001) Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Trans Pattern Anal Mach Intell* 23(10):1137–1143
- Lyzinski V, Fishkind D, Fiori M, Vogelstein JT, Priebe CE, Sapiro G (2015) Graph matching: relax at your own risk. *IEEE Trans Pattern Anal Mach Intell* 38(1):60–73
- Lyzinski V, Sussman DL (2020) Matchability of heterogeneous networks pairs. *Inf Inference A J IMA* 9(4):749–783
- Magnani M, Rossi L (2011) The ml-model for multi-layer social networks. In: 2011 International conference on advances in social networks analysis and mining, pp 5–12. IEEE
- McDiarmid C (1989) On the method of bounded differences. *Surv Combin* 141(1):148–188
- Moorman JD, Chen Q, Tu TK, Boyd ZM, Bertozzi AL (2018) Filtering methods for subgraph matching on multiplex networks. In: 2018 IEEE international conference on big data (big data), pp 3980–3985. IEEE
- Moorman JD, Tu TK, Chen Q, He X, Bertozzi AL (2021) Subgraph matching on multiplex networks. *IEEE Trans Netw Sci Eng* 8(2):1367–1384
- Mucha PJ, Richardson T, Macon K, Porter MA, Onnela J (2010) Community structure in time-dependent, multiscale, and multiplex networks. *Science* 328(5980):876–878
- Ng MK, Li X, Ye Y (2011) Multirank: co-ranking for objects and relations in multi-relational data. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1217–1225. ACM
- Onaran E, Garg S, Erkip E (2016) Optimal de-anonymization in random graphs with community structure. In: 2016 50th Asilomar conference on signals, systems and computers, pp 709–713. IEEE
- O'Shea K, Nash R (2015) An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*
- Patsolic H, Adali S, Vogelstein JT, Park Y, Priebe CE, Li G, and Lyzinski V (2014) Seeded graph matching via joint optimization of fidelity and commensurability. *arXiv preprint arXiv:1401.3813*
- Pedarsani P, Grossglauser M (2011) On the privacy of anonymized networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1235–1243. ACM
- Pinheiro MA, Kybic J, Fua P (2016) Geometric graph matching using Monte Carlo tree search. *IEEE Trans Pattern Anal Mach Intell* 39(11):2171–2185
- Priebe CE, Sussman DL, Tang M, Vogelstein JT (2015) Statistical inference on errorfully observed graphs. *J Comput Graph Stat* 24(4):930–953
- Singh R, Xu J, Berger B (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proc Natl Acad Sci* 105(35):12763–12768
- Sussman DL, Park Y, Priebe CE, Lyzinski V (2019) Matched filters for noisy induced subgraph detection. *IEEE Trans Pattern Anal Mach Intell* 42(11):2887–2900
- Takes FW, Kusters WA, Witte B (2017) Detecting motifs in multiplex corporate networks. In: International workshop on complex networks and their applications, pp 502–515. Springer
- Tu TK, Moorman JD, Yang D, Chen Q, Bertozzi AL (2020) Inexact attributed subgraph matching. In: 2020 IEEE international conference on big data (big data), pp 2575–2582. IEEE
- Ullmann JR (1976) An algorithm for subgraph isomorphism. *J ACM (JACM)* 23(1):31–42
- Umeyama S (1988) An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Anal Mach Intell* 10(5):695–703
- Vogelstein JT, Conroy JM, Lyzinski V, Podrazik LJ, Kratzer SG, Harley ET, Fishkind DE, Vogelstein RT, Priebe CE (2014) Fast approximate quadratic programming for graph matching. *PLoS ONE* 10(4):1
- White JG, Southgate E, Thomson JN, Brenner S (1986) The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philos Trans R Soc Lond B Biol Sci* 314(1165):1–340
- Yan J, Yin X, Lin W, Deng C, Zha H, Yang X (2016) A short survey of recent advances in graph matching. In: Proceedings of the 2016 ACM on international conference on multimedia retrieval, pp 167–174. ACM
- Yang B, Liu J (2016) Mining multiplex structural patterns from complex networks. In: *Wisdom Web of Things*, pp 275–301. Springer
- Zampelli S, Deville Y, Solnon C (2010) Solving subgraph isomorphism problems with constraint programming. *Constraints* 15(3):327–353

- Zanfir A, Sminchisescu C (2018) Deep learning of graph matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2684–2693
- Zaslavskiy M, Bach F, Vert JP (2009) A path following algorithm for the graph matching problem. *IEEE Trans on Pattern Anal Mach Intell* 31(12):2227–2242
- Zhang S, Tong H (2016) Final: Fast attributed network alignment. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1345–1354
- Zhang Z, Xiang Y, Wu L, Xue B, Nehorai A (2019) Kergm: Kernelized graph matching. *Adv Neural Inf Proces Syst* 32
- Zhou F, De la Torre F (2012) Factorized graph matching. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 127–134. IEEE
- Zuo X-N, Anderson JS, Bellec P, Birn RM, Biswal BB, Blautzik J, Breitner J, Buckner RL, Calhoun VD, Castellanos FX et al (2014) An open science resource for establishing reliability and reproducibility in functional connectomics. *Sci Data* 1(1):1–13

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
