AFRL-RI-RS-TR-2023-224

ALGORITHMIC PRIMITIVES FOR ALIGNING AND MERGING COMPLEX NETWORKS

UNIVERSITY OF MARYLAND

DECEMBER 2023

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2023-224 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
HANNAH UVANNI
Work Unit Manager

/ S /
SCOTT PATRICK
Deputy Chief,
Intelligence Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
DECEMBER 2023	FINAL TECHNICAL REPORT	MAY 2020	JULY 2023

4. TITLE AND SUBTITLE

ALGORITHMIC PRIMITIVES FOR ALIGNING AND MERGING COMPLEX NETWORKS

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-20-2-1001	N/A	
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		R30D

6. AUTHOR(S)

Vince Lyzinski, Daniel L. Sussman, Carey E. Priebe, Youngser Park, Ben Johnson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Maryland
3112 Lee Building, 7809 Regents Drive
College Park MD 20742-5141

8.PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

DARPA
675 North Randolph St
Arlington VA 22203-2114

10.SPONSOR/MONITOR'S ACRONYM(S)

AFRL / RI & DARPA

11.SPONSOR/MONITOR'SREPORT NUMBER(S)

AFRL-RI-RS-TR-2023-224

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. **SUPPLEMENTARY NOTES**

14. ABSTRACT

In the course of working on Phase II of the DARPA Modeling Adversarial Activity (MAA) program, we have worked to further develop our Multiplex Graph Matching Matched filter methodology, focusing on extending our core approach to the very large, richly featured networks that were the focus of MAA Phase II. We further worked to refine and extend our open source R code-base, denoted iGraphMatch—which is available for download on CRAN—for robust simulation of correlated graph pairs, efficient implementation of a suite of matching approaches, and novel matching visualization methods. We produced illustrative simulations and data analyses on MAA provided data and on externally provided real data sources.

15. SUBJECT TERMS

Graph matching, template discovery, vertex nomination, graph alignment, network analysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	U	U	SAR	24

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
HANNAH UVANNI	N/A

PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

LIST OF TABLES

1. SUMMARY

Note: This report will focus on our progress in Phase II of the MAA program. For a complete accounting of our team's accomplishments under Phase I of the program, please see our Phase I final report (available at https://apps.dtic.mil/sti/pdfs/AD1101398.pdf).

In the course of working on Phase II of the DARPA Modeling Adversarial Activity (MAA) pro- gram, we have worked to further develop our Multiplex Graph Matching Matched filter methodology, focusing on extending our core approach to the very large, richly featured networks that were the focus of MAA Phase II. We introduced and further developed the theory of alignment strength, which provides a principled measure of correlation across networks and a practical measure of the fidelity of a given alignment between two networks. We further worked to refine and extend our open source `R` code-base, denoted `iGraphMatch`—which is available for download on `CRAN`—for robust simulation of correlated graph pairs, efficient implementation of a suite of matching approaches, and novel matching visualization methods. We extended our work on vertex nomination (VN) along multiple fronts: developing new and enhanced VN algorithms, developing ensemble VN approaches, extending our work on robust VN in the presence of adversarial noise, and working to develop the closely related subgraph nomination framework. We developed novel methods and theory for simultaneously embedding multiple networks, with an aim towards robust analysis of network time-series. We produced illustrative simulations and data analyses on MAA provided data and on externally provided real data sources.

The cooperative agreement period is (including extensions) 4/21/2020 - 7/31/2023. In addition to prime PI Lyzinski, and subcontract PIs Sussman, Park and Priebe, and supplement contractor Johnson, Dr. Jesus Arroyo (Postdoc funded by MAA at UMD (University o, now in a tenure-track position in the Department of Statistics at Texas A&M) worked on key aspects of theory and methods development and helped implement our algorithms in `R`. Graduate students involved include:

At Boston University (BU):

- Wenrui Li: Graduated May 2021 (co-advised by Prof. Eric Kolaczyk, BU Dept. Math and Stats.), Postdoc at University of Pennsylvania in Department of Biostatistics, Epidemiology and Informatics
- Christy Lin: Graduated Aug 2021 (co-advised by Prakash Ishwar, BU Dept. Electrical and Computer Engineering), currently at Moderna
- Ben Draves: Graduated 2021, now working at Uber
- Kelly Kung: Graduated 2022 (co-advised by Prof. Judith Lok, BU Dept. Math and Stats.), now at Hewlett-Packard
- Esther Wang: Graduated 2022, currently at Liberty Mutual
- Zihuan Qiao: Graduated 2022, now at Meta

At University of Maryland (UMD):

- Jesse Milzman: Graduated August 2021, (co-advised by Prof. Doron Levy at UMD), at Army Research Lab.
- Konstantinos Pantazis: Graduating May 2022, Postdoctoral fellowship at JHU 2022-23
- Ayushi Saxena: Ph.D. Candidate- Graduating May 2024 (est.)
- Sheyda Peyman: Ph.D. Candidate- Graduating May 2024 (est.)
- Al-Fahad Al-Qadhi: Ph.D. Candidate- Graduating May 2025 (est.)
- Tong Qi: Ph.D. Candidate- Graduating May 2025 (est.)

- Zhirui Li: Ph.D. Candidate- Graduating May 2025 (est.)

At Johns Hopkins (JHU):
- Heather Patsolic: Graduated Spring 2020 (co-advised by Priebe and Lyzinski), now at Accenture Federated Services
- Joshua Agterberg: Graduated 2023, now at UPenn as postdoc, starting as Assistant Professor, Department of Statistics, University of Illinois in AY2024
- Lingyao Meng: Graduated 2021, now at Bloomberg LP
- Chen, Guodong – defending Fall 2023 (est.)
- Archarrya, Aranyak – defending Spring 2024 (est.)
- Chen, Tianyi – defending Fall 2024 (est.)

2. METHODS, ASSUMPTIONS, AND PROCEDURES

Key contributions to the program: The Multiplex Graph Matching Matched Filter method provides a scalable, highly-flexible approach for detecting noisy induced multiplex template in a larger richly-featured multiplex background network.

The wealth and variety of network datasets provided in Phase II of the MAA program required significant extensions of our core Graph Matching Matched Filter (GMMF) [1] methodology. The GMMF method was designed to detect noisy induced matches to a provided template network in a larger, complex background network. The method leveraged the classical graph matching problem to use the template as a matched filter for efficiently searching the background for candidate template matches. To extend the GMMF method to the large, richly featured, multi-channel graphs in MAA Phase II, we needed to extend the classical graph matching framework to th emulti-channel network setting, efficiently incorporate a wealth of vertex and edge features into the GMMF optimization framework, and develop efficient algorithmic primitives for feasibly running the matched filter approach on graphs with tens-of-millions of vertices. The resulting routine, dubbed the Mulitplex Graph Matching Matched Filter (MGMMF) method, is fast, flexible and achieves excellent performance on the MAA provided challenge datasets.

Our core MGMMF approach begins by incorporating the rich vertex and edge features of the transactional semantic property graphs of MAA Phase II into the multiplex graph matching problem defined in [2]. For a suitably weighted (see [2] for a variety of weighting schemes) of m-vertex multi-channel template $(\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_c)$ and n-vertex background $(\tilde{B}_1, \tilde{B}_2, \ldots, \tilde{B}_c)$the multiplex graph matching problem seeks to solve

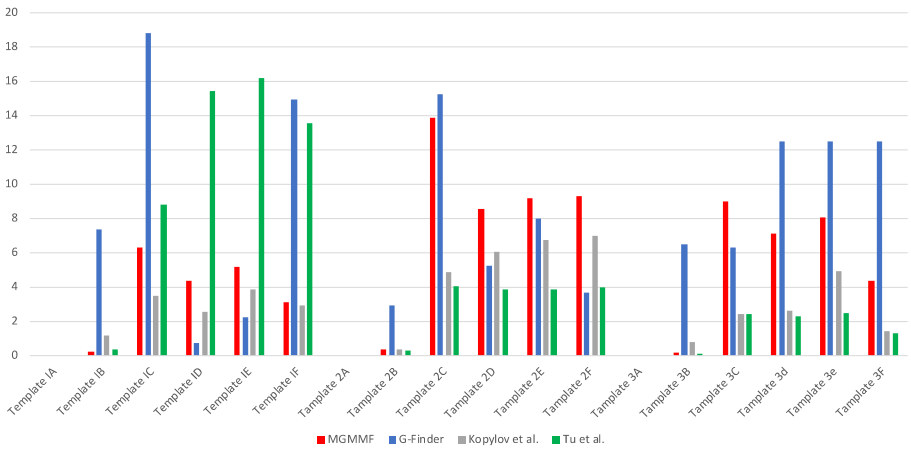

Figure 1. GED scores on AIDA v2.1.2: Graph edit distance scores (lower is better) from [3] for the best recovered signal for the filter algorithms of [4] (green) and [5] (gray); G-finder of [6] (blue; as implemented in [5]) and M-GMMF (red; with 1000 random restarts) for each of the 18 templates.

$$\operatorname*{argmin}_{P\in\Pi_n}\sum_{i=1}^{c}\|(\widetilde{A}_i\oplus\mathbf{0}_{n-m})P-P\widetilde{B}_i\|_F^2 \Leftrightarrow \operatorname*{argmin}_{P\in\Pi_n}\sum_{i=1}^{c}-\operatorname{trace}(\widetilde{A}_i\oplus\mathbf{0}_{n-m})P\widetilde{B}_iP^T), \tag{1}$$

where Π_n is the set of $n\times n$ permutation matrices (a permutation is a binary matrix with precisely one 1 in each row and column of the matrix; here $P\,\tilde{B}_iP^T$ relabels the vertices of $\tilde{B}_i$ in order to maximize the common structure between $\tilde{B}_i$and $\tilde{A}_i\ \oplus\mathbf{0}_{n-m}$). In the above framework, edge weights are easily incorporated by suitably weighting the adjacency matrices $\tilde{A}$ and $\tilde{B}$, and edge directedness is easily incorporated by considering directed adjacency matrices. Vertex features can be encoded into the GMP via a similarity matrix S into Eq. 1 yielding the following formulation

$$\operatorname*{argmax}_{P\in\Pi_n}\sum_{i=1}^{c}\operatorname{trace}(\widetilde{A}_i\oplus\mathbf{0}_{n-m})P\widetilde{B}_iP^T)+\operatorname{trace}(SP^T). \tag{2}$$

The key insight into lifting MGMMF to the feature-rich transactional knowledge graphs (TKB) was to encode edge-features into the multiple channels of the background and the template to systematically incorporate the vertex features into the similarity matrix S. We encode the different types of edges in the TKB into different channels. Within a TKB, we see the different (`E(rdf:type)`, `E(argument)`) pairs as representing fundamentally different types of edges, much as edges across different data sources do; e.g., venmo transactions are one type of edge, and they would be encoded into a separate channel from the journal-author relations from the citation network. As such, each (`E(rdf:type)`, `E(argument)`) pair provides a distinct channel in the multiplex graph, and background/template edges are divided amongst the channels via

i. A hard split based on `E(argument)`: Within each (`E(rdf:type)`, `E(argument)`) channel, only edges with matching `E(argument)` are potentially present. If the `E(argument)` property is missing in the background, we allow the edge to possibly exist in all (`E(rdf:type)`, `E(argument)`) channels.

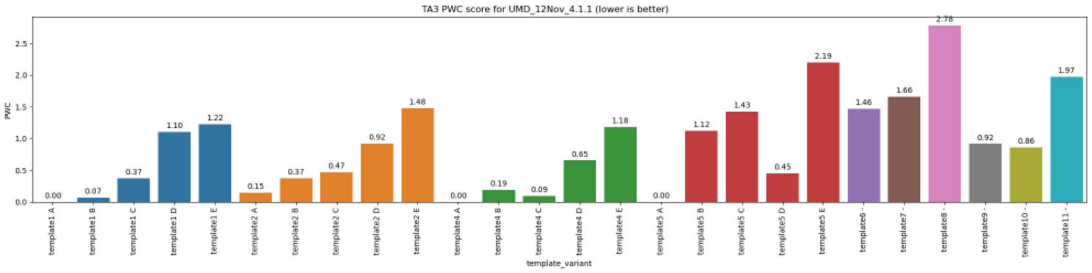

Figure 2. GED scores on NYC v4.1.1: Graph edit distance scores (lower is better) for MGMMF on the templates provided in the NYC v4.1.1 dataset. For templates 1, 2, 4, 5 there are five template variants with varying noise levels and types (with variant A isomorphically embedded); for templates 6–11, a single template (with unknown noise) was provided.

ii. A soft split (weighted according to `E(rdf:type)` similarity: Within each (`E(rdf:type)`, `E(argument)`) channel, each background edge with matching (or missing) `E(argument)` is present, and the edge is weighted according to a similarity measured between its `E(rdf:type)` and that of the channel. Scalability gains can be achieved by thresholding the similarities to improve sparsity.

We lastly incorporate space/time constraints via extra constraint channels in the multiplex network (constraining matching edges to have the desired spatiotemporal structure). Lastly, numerically weighted edges are encoded via edge weights in the template/background. To encode the vertex features into Eq.1, we simply require a way to compute vertex "feature similarities." These similarities are encoded into an $m \times n$ similarity matrix S which is efficiently incorporated in iGraphMatch. To encode both vertex and edge features, we simply require a way to compute vertex and edge feature similarities; indeed, we view this as being a key to automating our procedure and allowing for further applicability of our method in novel data domains. Performance results for an early implementation of MGMMF on the AIDA v2.1.2 TKB, see Figure 1.

Our core MGMMF approach uses constrained gradient descent in order to discover local minima (for the quadratic, richly-featured graph matching objective function in Eq. 2) in the feasible region. As the objective function is quadratic, the gradient descent step involves finding the optimal gradient search direction, which here amounts to solving a Linear Assignment Problem (LAP). This LAP has been a computational bottleneck in our code, preventing us from running our approach on the ever larger Phase II program datasets. Here the gradient $\nabla P^{(t)}$ is effectively a very wide cost matrix with m rows and n columns, for $m \ll n$. In this case, for the LAP it's straightforward to show that row i can be "matched" to column j iff

$$\nabla P^{(t)}[i, j] \leq m\text{-th smallest entry of } \nabla P^{(t)}[i, :]. \quad (3)$$

To whittle down the cost matrix, we can "delete" all columns of $\nabla P^{(t)}$ for which this condition fails for all rows, which yields a new cost matrix C with m rows and at most m^2 columns. We can then solve the LAP problem on C, and obtain the same answer as solving the LAP problem on $\nabla P^{(t)}$. Note that a LAP solver runtime scales cubically, naively solving the LAP problem on $\nabla P^{(t)}$ with

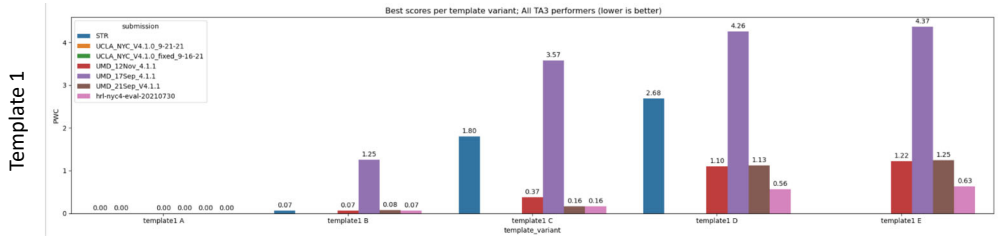

Figure 3. GED scores on NYC v4.1.1 Template 1: Graph edit distance scores (lower is better) from [3] for comparing performance across MAA Phase II performers; MGMMF is the red bar.

$O(n^3)$ complexity, this Row deletion procedure yields an LAP routine with $O(mn + m^4)$ complexity [7]. Practically, this technique (briefly mentioned in [8], and formalized in MGMMF in [7]) allows for immense speed-up of this subroutine LAP bottleneck, and allows us to efficiently implement our MGMMF approach on the largest provided program datasets without any loss in performance that could be inherent to an approximate LAP procedure. For example, the running times (on a 32 cores, Azure machine) for 1000 random restarts matching the V41s-2p-1c.json (Template) to V22-hrl-alignment.json are as follows: 10.4 hours (Original R implementation), 35 seconds (Python with fastLAP), and 10 seconds (C++ with fastLAP).

For the very large NYC V4.1.1 TKB's, preprocessing the data took ≈ 3 minutes per template /background pair, and each random restart (multiple restarts are run to efficiently search for local maxima of the non-convex Eq. 2) feasible region) took between $\approx 11 - 33$ seconds on aver- age to run. This was run on an Amazon Web Services (AWS) EC2 `c5.24xlarge` instance with 96 cores; we execute 4 Monte Carlo (MC) runs in parallel, with 24 threads per MC run and OpenMP dynamic scheduling. Performance is summarized in Figures 2-6. The dataset consisted of a very large background TKB (on the order of 10^7 vertices) with many smaller templates to match to the background. For each of 5 templates (templates 1–5), there are five variants with varying noise levels and noise types (with variant A isomorphically embedded). For each of six additional templates (templates 6–11) a single template with unknown noise structure was provided. Our overall performance, as measured by the Graph Edit Distance (GED) algorithm of [3], is plotted in Figure 2. Lower GED is better, and we see the expected result that as more error is added to the templates, our recovered signal is noisier. Comparing our performance to the other MAA program performers (see Figures 3-6; MGMMF is the red bar), we see that MGMMF is comparable to the best program algorithms, often achieving near-optimal or optimal results. While we focused here on the large NYC

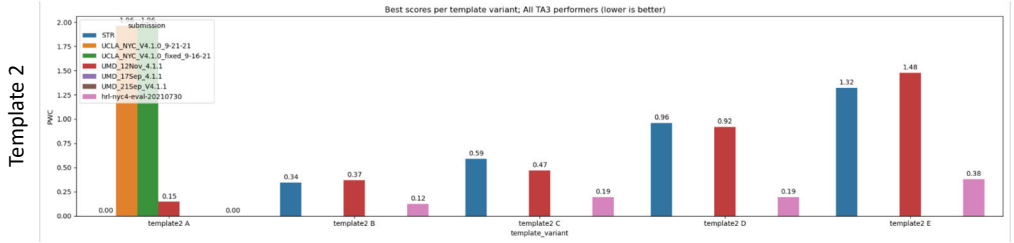

Figure 4. GED scores on NYC v4.1.1 Template 2: Graph edit distance scores (lower is better) from [3] for comparing performance across MAA Phase II performers; MGMMF is the red bar.

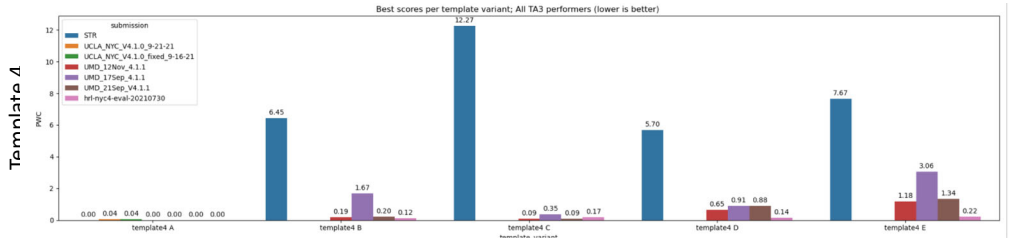

Figure 5. GED scores on NYC v4.1.1 Template 4: Graph edit distance scores (lower is better) from [3] for comparing performance across MAA Phase II performers; MGMMF is the red bar.

challenge dataset, we note that similar performance and scalability was exhibited on other Phase II data sets as well.

MGMMF is focused on finding the best fitting match to the template in the background rather than all closely matching templates. In order to diversify the matched templates returned by MG- MMF, we also developed an adaptive weighting procedure for solution diversification in [7]. In both theory and practice, we demonstrate the capability of sequentially modifying the MGMMF algorithm to bias the algorithm against returning previously recovered solutions. We demonstrate this here via the following experiment. To measure the fidelity of recovered templates when no ground truth is present, we use the graph edit distance (GED) metric outlined in [3]. We run 32 random restarts of the MGMMF algorithm for each template recovery (we plot results for template 1A, 1B, 1C, 1D here; other templates yield similar results), plotting the empirical distribution function of the GED of the recovered templates; results are plotted in Figure 7 . Different penalization values are represented with different colors in the plot. From Figure 7, we see that the solution diversification is successful at yielding recovered templates including our optimal fits (the best recovered templates in the $\varepsilon = 0$ case) and templates that are close to optimal by GED. These templates further recover significant signal not recovered in the $\varepsilon = 0$ case; see Table 1. In the table, for each vertex in the template we show how many vertices in the background have similarity greater than 0 (i.e., are potential matches)—this is shown in the #> 0 row. We see that the solution diversification is successful at recovering additional possible matches in the suboptimal recovered templates. We also see that too severe of a penalty (here the $\varepsilon = 0.01$ case) yields fewer unique nodes and worse GED fits. While choosing an optimal ε is of paramount importance, we do not have a fully principled recommendation for a best choice. We do recommend smaller penalty combined with more

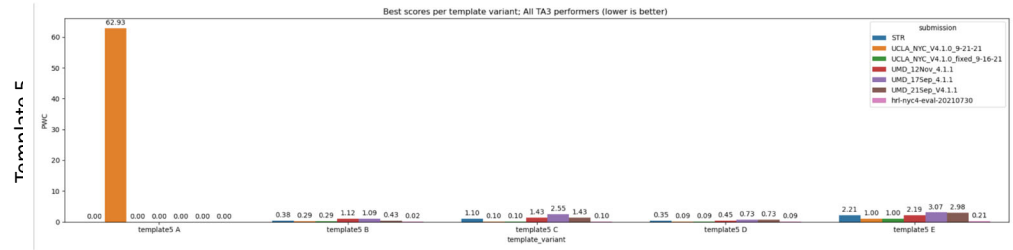

Figure 6. GED scores on NYC v4.1.1 Template 5: Graph edit distance scores (lower is better) from [3] for comparing performance across MAA Phase II performers; MGMMF is the red bar.

random restarts which achieved our best results.

3. RESULTS AND DISCUSSION

Herein, we briefly provide more detail for the methods and algorithms outlined in the introduction. While we focus here on *alignment strength*, *vertex nomination*, `iGraphMatch`, and *multiple graph embeddings methods*, we made significant progress on a number of other, MAA inspired, projects. In the area of graph matching, we developed a first procedure for principally matching across unipartite and bipartite graphs [9]; developed novel regularization methods for matching across heterogeneous topologies [10]; and made novel connections between maximum

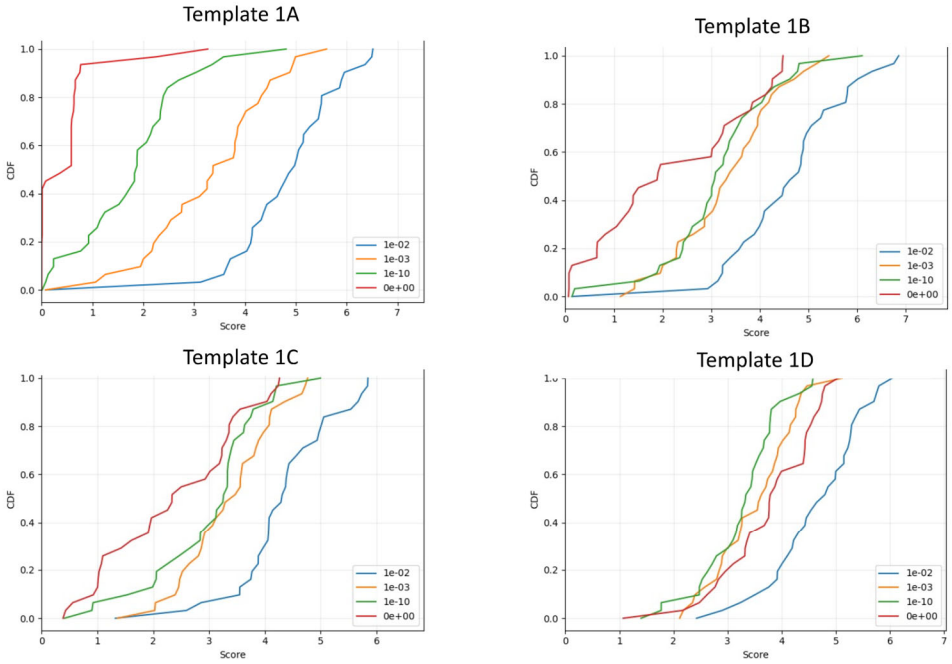

Figure 7. We run 32 random restarts of the MGMMF algorithm for each template recovery, plotting the empirical CDF of the GED of the recovered templates. Different penalization values are represented with different colors in the plot.

Table 1. Template 1A potential matches for different penalizations

node	1	2	34	5	6	7	8	9	10	11	12	13	14	15
$\varepsilon = 0$	1	8	13	20	1	1	1	3	1	14	2	1	1	5
$\varepsilon = 0.01$	1	23	215	32	1	1	1	8	1	24	5	4	1	24
$\varepsilon = 10^{-3}$	4	21	2029	32	5	5	4	27	1	26	5	27	3	31
$\varepsilon = 10^{-10}$	4	32	3032	31	5	5	4	32	5	30	5	32	4	32
#> 0	4	32	3232	32	5	5	4	32	5	32	5	32	4	32
node	16	17	1819	20	21	22	23	24	25	26	27	28	29	30
$\varepsilon = 0$	1	1	11	2	3	1	3	19	1	1	1	3	4	1
$\varepsilon = 0.01$	1	1	11	3	16	1	14	32	1	2	2	5	18	1
$\varepsilon = 10^{-3}$	1	4	43	3	27	5	26	31	12	5	4	5	27	4
$\varepsilon = 10^{-10}$	4	20	43	3	32	5	32	32	29	5	4	5	32	4
#> 0	4	32	43	3	32	5	32	32	32	5	4	5	32	4
node	31	32	3334	35	36	37								
$\varepsilon = 0$	3	4	12	4	1	1								
$\varepsilon = 0.01$	3	25	12	5	1	1								
$\varepsilon = 10^{-3}$	13	31	415	5	5	6								
$\varepsilon = 10^{-10}$	28	32	431	5	5	21								
#> 0	32	32	432	5	5	32								

likelihood estimation and graph matching and matchability [11]. In the area of network estimation and causal inference, we studied the effect of network noise on branching factors in network epidemiological models [12]; and we studied causal inference under network interference in noisily observed network settings [13]. In the area of network embeddings, we proposed a theoretical framework for analyzing random walk based node-embedding algorithms for graphs based on learning node embeddings by optimizing objective functions involving skip-bigram statistics computed from random walks on a graph [14]. This framework subsumes several existing algorithms as special cases and introduces extensions and techniques that simplify theoretical analysis.

3.1 Alignment Strength

Key contributions to the program: Alignment strength practical measure of the correlation across graphs and allows us to discover whether two graphs are "correctly" matched after the fact.

In a sequence of papers [15, 16, 17], we introduced and developed the notion of the alignment strength between a pair of graphs. For two graphs G and H (with true alignment P) the alignment strength is defined via

Table 2. Performance on COVID data

k	average P@k	chance P@k
5	0.4	0.07
10	0.65	0.14
100	7.2	1.40
200	12.6	2.80

$$\mathrm{str}(G,H,P) := 1 - \frac{\|G - PHP^T\|_1^2}{\frac{1}{n!}\sum_{Q\in\Pi(n)} \|G - QHQ^T\|_1^2}, \tag{3}$$

where $\Pi(n)$ is the set of $n \times n$ permutation matrices. Alignment strength provides a principled measure of graph correlation, suitably balancing the dual contributions to total correlation of the common graph structure within each network and the edge correlation across networks. Early work focused on defining alignment strength as a suitable measure of correlation and looking at the impact of alignment strength on graph matchability and on algorithmic matching runtime/difficulty [15]. Subsequent work focused on developing efficient procedures for estimating the alignment strength across networks [16], and on the potential use of alignment strength to discover whether two graphs are "correctly" matched after the fact [17]. In this later work, we focus on the question of what happens when the true matching is not "optimal" according to the objective function. Can we detect this wrong alignment and the phantom correlation it induces? Our work suggests that alignment strength provides a principled and practical means to detect the level of phantom correlation and the threshold for correlation after which the phantom correlation substantiates to indicate that the graph matching recovered the true signal.

3.2 Vertex Nomination

Key contributions to the program: Vertex nomination algorithms provide principled lightly-supervised methods for retrieving matched vertices across multiple networks. Our Phase II work has focused on improved and scalable algorithmic development, further developing robust VN procedures, and extending the methodology of VN to subgraph retrieval problems.

Vertex nomination (VN) [18, 19, 20, 21] comprises a suite of algorithms that aim to efficiently query (often large) network data sets given limited training. Similar in spirit to popular network-based information retrieval (IR) procedures such as `PageRank` [22] and personalized recommender systems on graphs [23], in vertex nomination the goal is as follows: Given vertices of interest in a graph G_1, rank the vertices in a second graph G_2 based on how likely they are judged to be interesting; with (ideally) interesting vertices in G_2 concentrating at the top of the rank list. Our Phase II MAA efforts have focused on the following MAA-inspired directions:

- Developing highly scalable VN procedures that better leverage the information contained in existing seeded vertices [24] (seeded vertices, those with a priori known correspondence across

networks played a key role in many of our early VN procedures); as seeds are often costly to obtain, we worked to develop better VN algorithmic primitives [25] designed to allow us to "un-seed" our previous VN approaches.

- Developed new procedures for nominating vertices in the presence of adversarial network contamination [26], that dramatically improved upon our previous efforts in this direction [27].

- Providing a novel formulation of the learning-to-rank problem in VN, including a practical integer linear programming solution to the problem of combining multiple VN rank-lists into a single high-fidelity ranking [28, 29]. As part of our MAA efforts, we considered working with the tripartite Knowledge Graph downloaded from https://blender.cs.illinois.edu/covid19/, and we sought to use VN to discover potentially useful COVID chemicals/drugs in this KG. After data pre-processing (removing isolated nodes, etc.) we had a pair of networks on 13292 vertices. Each vertex represents a chemical/drug in the KG and the two networks separately capture the chemical-gene (G_1) and chemical-disease interactions (G_2). For our VN training data, we scraped https://clinicaltrials.gov/ct2/results?cond=COVID-19 to compile a list of drugs/interventions being utilized in COVID clinical trials. Of this list, 86 drugs appear in our graphs G_1 and G_2, providing a training set of size 86 for use in VN. Of these 86 vertices of interest , 20 appear in DARPA Biological Technologies Office (BTO) list. Of the full set of 13292 vertices, 186 appear in DARPA BTO list. Precision at k values for our learning-to-rank method are presented below in Table 2. Chance here would be $\approx$1.4% correct retrieval. These results show that our approach is significantly better than chance, and even in this setting when the training data (our 86 drug list) and the test data (the 186 elements in the BTO list) differed significantly, i.e., with "bad" training data, learning—to—rank VN able to discover significant signal.

- Developed the framework for the Subgraph Nomination inference task; similar in spirit to vertex nomination, in subgraph nomination example subgraphs of interest are used to query a network for similarly interesting subgraphs. This is a core MAA TA3 task, especially if ranked lists of possible matches to the template is desired. In addition to developing the necessary mathematical framework for subgraph nomination, we also explore the utility of user-in-the-loop supervision in the inference cycle, demonstrating the effectiveness of even light supervision across both synthetic and real data examples [30].

- Hearkening back to the origins of the VN task in which network features were instrumental [18, 31, 20], we extended our VN formalism to the setting of richly feature networks [32], paying special heed to the key question of when utilizing both content and context will produce significantly better VN performance than using only content or context alone.

All of these advancements and developments have been aimed at both achieving a better understanding of the theoretical core at the heart of vertex nomination (and its variants), as well as implementing the actionable insights gleaned from our theoretical analysis into more robust, more scalable VN algorithms.

3.3 Joint Graph Embedding

Key contributions to the program: We have developed a suite of algorithms for jointly embedding collections of graphs, as well as novel theoretical methodologies to analyze the graphs in embedded space. These efforts are essential for principally extending classical change-point detection and anomaly detection methods to network data [33].

For MAA, joint embedding approaches have been useful for approaching the matching problem from a spectral graph inference perspective, for understanding how different network layers— as defined by differing edge-types, modality, etc—relate to each other, and for improving our optimization-based matching criterion. One of our key efforts [34] has been analyzing the prop- erties of the "Omnibus Spectral Embedding" methodology of [35] for jointly embedding multiple graphs. Joint graph embedding methods, by necessity, induce correlation between networks in the embedded space, and this paper is the first in the literature to precisely characterize the level of the correlation in a particular method and the first to analyze spectral embeddings for correlated net- works. We further propose a generalization of the Omnibus method that allows for the inducement of richer correlation structures amongst the networks, enabling higher-fidelity application of the omnibus methodology in real network time-series data.

We also analyze the properties of the Omnibus method when the different layers have different connectivity structures [36]. This is one of the first attempts to understand how joint embeddings—which attempt to recover node embeddings for each vertex in each layer and emphasize shared connectivity structure across layers—trade off between bias and variance in order to accurately capture varying network structure. Our results demonstrate that many inference tasks such as clustering, testing, and latent position estimation can benefit from joint embeddings even when the networks are very different.

4. iGRAPHMATCH

iGraphMatch [37] is an R package developed for finding corresponding vertices between two graphs, also known as graph matching. iGraphMatch is available on CRAN at https://cran. r-project.org/web/packages/iGraphMatch/index.html, with a fully worked out vignette with examples available at https://cran.r-project.org/web/packages/iGraphMatch/vignettes/ iGraphMatch.html. The package has been instrumental in broadening the scope of our Graph Matching user-base, with 11K downloads (as of 10/23/23 from https://cranlogs.r-pkg.org/ badges/grand-total/iGraphMatch). The package implements three categories of prevalent graph matching algorithms including relaxation-based, percolation-based, and spectral-based. For all cases, we have implemented them to work for a highly general set of graph properties, including weighted or unweighted graphs, directed or undirected graphs, multilayer networks, and graphs with differing number of vertices. We provide versatile options to incorporate prior information in the form of hard seeds, "soft" seeds, and similarity scores. In addition, iGraphMatch provides functions to summarize the graph matching results in terms of several evaluation measures and visualize the matching performance. Finally, the package enables users to sample correlated random graph pairs from classic random graph models to generate data for simulations.

Usage The graph matching methods share the same basic syntax:

```
gm(A, B, seeds = NULL, similarity = NULL, method = "indefinite",
***algorithm parameters***)
```

The first two arguments are graph-like objects, `seeds` is used to indicate hard seeds, and `similarity` is used to specify a cross-graph node similarity. The `method` argument specifies a graph matching algorithm to use, and one can choose from "indefinite" (default), "convex", "PATH", "percolation", "IsoRank", "Umeyama", or a user-defined graph matching function which enables users to test out their own algorithms while exploiting the generalizations and analysis tools of the package. The remaining parameters may vary depending on the specific method. An example call to 'gm' is provided below.

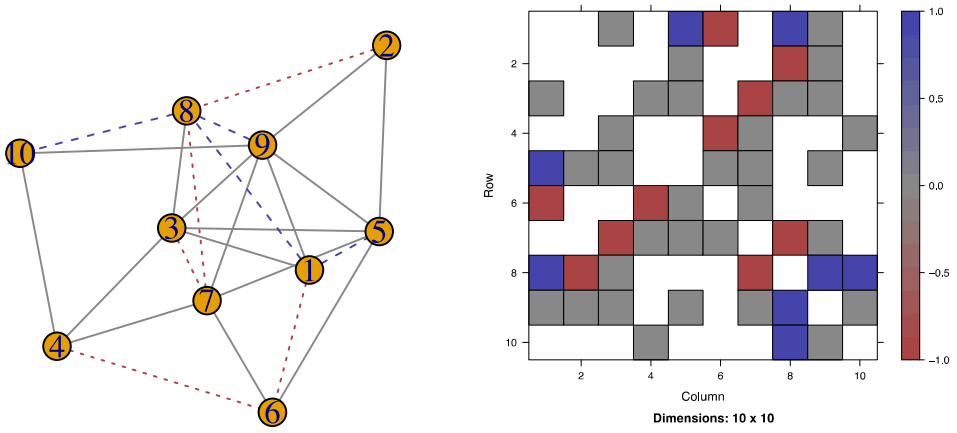

Figure 8. Example match in iGraphMatch

```
> set.seed(123)
> cgnp_pair <- sample_correlated_gnp_pair(n = 10, corr =  0.5, p =  0.5)
> g1 <- cgnp_pair$graph1
> g2 <- cgnp_pair$graph2
> seeds <- 1:10 <= 4

> m_rds <- gm(g1, g2, seeds, method = "indefinite", start = "rds", max_iter
  = 20)
```

The graph match object can be inspected and used in a variety of ways. For example, permuting the second graph easily accomplished with.

```
> m_rds %*% g2
```

The summary output gives detailed information about common edges, missing edges and extra edge across multilayer graphs.

```
> summary(m_rds, g1, g2, true_label = 1:10)

Call: gm(A = g1, B = g2, seeds = seeds, method = "indefinite", start =
    "rds", max_iter = 20)

# Matches: 6
# True Matches:  6, # Seeds:  4, # Vertices:  10, 10
```

```
  common_edges 16.000000
 missing_edges  4.000000
   extra_edges  5.000000
         fnorm  4.242641
```

Visualizing matches is straightforward by incorporating the matching in a plot command.

```
> plot(g1,g2,m_rds)
> plot(g1[],g2[],m_rds)
```

Example: Enron Graph The `iGraphMatch` package includes the Enron email network data in the form of a pair of `igraph` objects derived from the original data where each graph represents one week of emails between 184 email addresses. The two networks are unweighted and directed with edge densities around 0.01 in each graph and the empirical correlation between two graphs is 0.85. We can visualize the aligned Enron networks using the `plot` function.

Note that 37 and 32 out of the total 184 nodes are isolated from the other nodes in two graphs respectively. We'll start with just matching the largest connected components of size 145 and 151.

```
...
R> A <- Enron[[1]][][c(vinsct, v1), c(vinsct, v1)]
R> B <- Enron[[2]][][c(vinsct, v2), c(vinsct, v2)]
```

The sizes of largest connect components of two graphs are 146 and 151, which are different. Without any prior information, `seeds` and `similarity` arguments take default values which are `NULL`.

```
R> match_FW <- gm(A = A, B = B, method = "indefinite")
R> summary(match_FW, A, B)
Call: gm(A = A, B = B, method = "indefinite")
# Matches: 146, # Seeds:  0, # Vertices:  146, 151
  common_edges 353
 missing_edges 134
   extra_edges 120
         fnorm  16
```

We can then use the `best matches` function to measure and rank the vertex-wise matching performance.

```
R> bm <- best_matches(A = A, B = B, match = match_FW_center,
+               measure = "row_perm_stat")
```

Taking the *ns* best matches and using them as seeds in a second run of the matching algorithm allows us to adaptively add seeded vertices. We can also use this as a ranking for which vertices are most likely to appear in both graphs. For example, for the match above, the top 89 best matches are all nodes present in both graphs.

Future development In the short term we are looking to introduce a suite of additional matching methods that have recently been proposed in the literature and (as outlined in th MGMMF section)

incorporate the fast LAP algorithm for matching small graphs to large graphs. We also plan to incorporate novel methods for evaluating match quality and ranking matches.

5. CONCLUSIONS

Graph matching and graph template discovery are important emerging fields in the applied and theoretical statistics communities. Moreover, as graph-valued data becomes more common across the sciences, graph inference methods are becoming increasingly important to the broader scientific community. The DARPA MAA project enabled a deeper understanding of the possibilities and limitations of graph matching and template discovery in complex data environments, and the insights and methods developed therein should have a broad impact on myriad important academic/industrial/government programs that rely on these core analytics in their processing pipelines.

6. REFERENCES

[1] D. Sussman, Y. Park, C. E. Priebe, and V. Lyzinski, "Matched filters for noisy induced subgraph detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 2887–2900, 2020.

[2] K. Pantazis, D. L. Sussman, Y. Park, Z. Li, C. E. Priebe, and V. Lyzinski, "Multiplex graph matching matched filters," *Applied Network Science*, vol. 7, no. 1, pp. 1–35, 2022.

[3] C. L. Ebsch, J. A. Cottam, N. C. Heller, R. D. Deshmukh, and G. Chin, "Using graph edit distance for noisy subgraph matching of semantic property graphs," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2520–2525, IEEE, 2020.

[4] T. K. Tu, J. D. Moorman, D. Yang, Q. Chen, and A. L. Bertozzi, "Inexact attributed subgraph matching," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2575–2582, IEEE, 2020.

[5] A. Kopylov, J. Xu, K. Ni, S. Roach, and T.-C. Lu, "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, 2020.

[6] L. Liu, B. Du, and H. Tong, "G-finder: Approximate attributed subgraph matching," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 513–522, IEEE, 2019.

[7] Z. Li, B. Johnson, D. L. Sussman, C. E. Priebe, and V. Lyzinski, "Gotta match'em all: Solution diversification in graph matching matched filters," *arXiv preprint arXiv:2308.13451*, 2023.

[8] J. Bijsterbosch and A. Volgenant, "Solving the rectangular assignment problem and applications," *Annals of Operations Research*, vol. 181, no. 1, pp. 443–462, 2010.

[9] J. Arroyo, C. E. Priebe, and V. Lyzinski, "Graph matching between bipartite and unipartite

networks: to collapse, or not to collapse, that is the question," *IEEE Transactions on Network Science and Engineering*, 2021.

[10] V. Lyzinski and D. L. Sussman, "Matchability of heterogeneous networks pairs," *Information and Inference: A Journal of the IMA*, 01 2020. iaz031.

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

[12] W. Li, D. L. Sussman, and E. D. Kolaczyk, "Estimation of the branching factor in noisy networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 1, pp. 565-577, 2022.

[13] W. Li, D. L. Sussman, and E. D. Kolaczyk, "Causal inference under network interference with noise," *arXiv preprint arXiv:2105.04518*, 2021.

[14] C. Lin, D. Sussman, and P. Ishwar, "Ergodic limits, relaxations, and geometric properties of random walk node embeddings," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 1, pp. 346–359, 2022.

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

[16] D. E. Fishkind, A. Athreya, L. Meng, V. Lyzinski, and C. E. Priebe, "On a complete and sufficient statistic for the correlated bernoulli random graph model," *Electronic Journal of Statistics*, vol. 15, no. 1, pp. 2336 – 2359, 2021.

[17] D. E. Fishkind, F. Parker, H. Sawczuk, L. Meng, E. Bridgeford, A. Athreya, C. E. Priebe, and V. Lyzinski, "The phantom alignment strength conjecture: practical use of graph matching alignment strength to indicate a meaningful graph match," *Applied Network Science*, vol. 6, no. 62, 2021.

[18] D. Marchette, C. E. Priebe, and G. Coppersmith, "Vertex nomination via attributed random dot product graphs," in *Proceedings of the 57th ISI World Statistics Congress*, vol. 6, 2011.

[19] G. Coppersmith, "Vertex nomination," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 2, pp. 144–153, 2014.

[20] S. Suwan, D. S. Lee, and C. E. Priebe, "Bayesian vertex nomination using content and context," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 6, pp. 400–416, 2015.

[21] D. E. Fishkind, V. Lyzinski, H. Pao, L. Chen, and C. E. Priebe, "Vertex nomination schemes for membership prediction," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1510–1532, 2015.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

[23] Z. Huang, W. Chung, and H. Chen, "A graph model for e-commerce recommender systems," *Journal of the American Society for information science and technology*, vol. 55, no. 3, pp. 259–274, 2004.

[24] J. Yoder, L. Chen, H. Pao, E. Bridgeford, K. Levin, D. E. Fishkind, C. E. Priebe, and V. Lyzinski, "Vertex nomination: The canonical sampling and the extended spectral nomination schemes," *Computational Statistics & Data Analysis*, vol. 145, p. 106916, 2020.

[25] R. Zheng, V. Lyzinski, C. E. Priebe, and M. Tang, "Vertex nomination between graphs via spectral embedding and quadratic programming," *Journal of Computational and Graphical Statistics*, vol. 31, no. 4, pp. 1254–1268, 2022.

[26] S. Peyman, M. Tang, and V. Lyzinski, "Adversarial contamination of networks in the setting of vertex nomination: a new trimming method," *arXiv preprint arXiv:2208.09710*, 2022.

[27] J. Agterberg, Y. Park, J. Larson, C. White, C. E. Priebe, and V. Lyzinski, "Vertex nomination, consistent estimation, and adversarial modification," *Electronic Journal of Statistics*, vol. 14, pp. 3230–3267, 2020.

[28] H. S. Helm, A. Basu, A. Athreya, Y. Park, J. T. Vogelstein, M. Winding, M. Zlatic, A. Cardona, P. Bourke, J. Larson, W. C., and P. C. E., "Learning to rank via combining representations," *arXiv preprint arXiv:2005.10700*, 2020.

[29] H. S. Helm, M. Abdin, B. D. Pedigo, S. Mahajan, V. Lyzinski, Y. Park, A. Basu, C. M. White, W. Yang, and C. E. Priebe, "Leveraging semantically similar queries for ranking via combining representations," *arXiv preprint arXiv:2106.12621*, 2021.

[30] A. M. Al-Qadhi, C. E. Priebe, H. S. Helm, and V. Lyzinski, "Subgraph nomination: Query by example subgraph retrieval in networks," *Statistics and Computing*, vol. 33, no. 2, p. 52, 2023.

[31] G. A. Coppersmith and C. E. Priebe, "Vertex nomination via content and context," *arXiv preprint arXiv:1201.4118*, 2012.

[32] K. Levin, C. E. Priebe, and V. Lyzinski, "On the role of features in vertex nomination: Content and context together are better (sometimes)," *arXiv preprint arXiv:2005.02151*, 2020.

[33] G. Chen, J. Arroyo, A. Athreya, J. Cape, J. T. Vogelstein, Y. Park, C. White, J. Larson, W. Yang, and C. E. Priebe, "Multiple network embedding for anomaly detection in time series of graphs," *arXiv preprint arXiv:2008.10055*, 2020.

[34] K. Pantazis, A. Athreya, J. Arroyo, W. N. Frost, E. S. Hill, and V. Lyzinski, "The importance of being correlated: Implications of dependence in joint spectral inference across multiple networks," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 6297–6373, 2022.

[35] K. Levin, A. Athreya, M. Tang, V. Lyzinski, Y. Park, and C. E. Priebe, "A central limit theorem for an omnibus embedding of random dot product graphs and implications for multiscale network inference," *arXiv preprint arXiv:1705.09355*, 2019.

[36] B. Draves and D. L. Sussman, "Bias-variance tradeoffs in joint spectral embeddings," *arXiv*

preprint arXiv:2005.02511, 2020.

[37] Z. Qiao and D. Sussman, "igraphmatch: an r package for the analysis of graph matching," *arXiv preprint arXiv:2112.09212*, 2021.

7. List of Acronyms

Acronym	Description
AWS	Amazon Web Services
BTO	Biological Technologies Office
BU	Boston University
GED	Graph Edit Distance
GMMF	Graph Matching Matched Filters
JHU	Johns Hopkins University
LAP	Linear Assignment Problem
MAA	Modeling Adversarial Activity DARPA program
MGMMF	Multiplex Graph Matching Matched Filters
MC	Monte Carlo
TKB	Transactional knowledge graphs
UMD	University of Maryland
VN	Vertex Nomination