CAPER: Coarsen, Align, Project, Refine A General Multilevel Framework for Network Alignment

Jing Zhu
University of Michigan, Ann Arbor
jingzhuu@umich.edu

Danai Koutra
University of Michigan, Ann Arbor
dkoutra@umich.edu

Mark Heimann
Lawrence Livermore Natl Laboratory
heimann2@llnl.gov

ABSTRACT

Network alignment, or the task of finding corresponding nodes in different networks, is an important problem formulation in many application domains. We propose CAPER, a multilevel alignment framework that Coarsens the input graphs, Aligns the coarsened graphs, Projects the alignment solution to finer levels and Refines the alignment solution. We show that CAPER can improve upon many different existing network alignment algorithms by *enforcing alignment consistency across multiple graph resolutions: nodes matched at finer levels should also be matched at coarser levels.* CAPER also accelerates the use of slower network alignment methods, at the modest cost of linear-time coarsening and refinement steps, by allowing them to be run on smaller coarsened versions of the input graphs. Experiments show that CAPER can improve upon diverse network alignment methods by an average of 33% in accuracy and/or an order of magnitude faster in runtime.

ACM Reference Format:
Jing Zhu, Danai Koutra, and Mark Heimann. 2022. CAPER: Coarsen, Align, Project, Refine A General Multilevel Framework for Network Alignment. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557563

1 INTRODUCTION

Graphs or networks are foundational representations for relational structure and their analysis is useful in innumerable scientific and industrial applications. In many diverse tasks, such as recommendation across multiple social networks, protein-protein interaction analysis, and database schema matching [7], it is necessary to discover meaningful correspondences between nodes in multiple networks. This general problem is called network alignment.

Network alignment methods in general have two main limitations. First, they may overfit to local structural similarity and fail to preserve higher-order measures of matching consistency [1, 4]. Second, especially the most accurate methods tend to rely on solving challenging optimization problems with high computational complexity, e.g. quadratic or cubic time in the number of nodes in one of the input graphs [1, 16, 20].

We argue that multilevel network analysis is a powerful technique for improving network alignment algorithms on both fronts. Accordingly, we design the first general multilevel framework to pair with any network alignment method, a four-step framework which we call CAPER: (1) Coarsening a graph into multiple levels of varying coarseness, (2) Aligning at the coarsest level, and (3) Projecting back to finer levels, and (4) Refining the solution at each level. We can accelerate the use of slow network alignment algorithms by running them on the smaller coarsened graphs, while refining the solutions at multiple levels of structural resolution encourages greater consistency between the local and global structure of matched nodes. Our contributions can be summarized as follows:

- **General-Purpose Framework**: We propose an intuitive multilevel framework (CAPER) in which any network alignment method can be used.
- **Design Choices and Empirical Success**: We propose and study specific design choices and parameter settings that work well within CAPER. We provide code and additional supplementary material at https://github.com/GemsLab/CAPER.
- **Study of Accuracy and Runtime Tradeoff**: Through complexity analysis and experiments, we show that CAPER is able to improve accuracy by 33% on average across multiple datasets and/or is 10x faster runtime than baselines, depending on the properties of the base methods employed.

2 RELATED WORK

Graph Coarsening and Multilevel Methods. Graph coarsening [12] is the process of shrinking a large graph into a similar smaller one, such that some properties or structures are preserved, e.g. spectral graph properties or cliques. It has been used to accelerate many graph mining tasks, including graph clustering [3], node embedding [2, 11] and graph neural networks [17].

Network Alignment. We focus on unsupervised approaches requiring no known matchings *a priori*. These can be categorized into two groups. (1) Classic graph alignment approaches often formulate an **optimization-based assignment** problem. FINAL [18] optimizes a topological consistency objective which may be augmented with node and edge attribute information. MAGNA [15], applied to biological networks, uses genetic algorithms to evolve network populations while maximizing proximity consistency criteria. More recently, Zhang *et al.* [20] leveraged kernel methods to solve the quadratic assignment problem, but requires cubic computational complexity. (2) Another line of work relies on **embedding-based methods.** REGAL [5] matches structural node embeddings [6, 14] that are directly comparable across networks. CONE-Align [1] uses embeddings modeling proximity within each graph [14] and aligns the graphs' embedding spaces with a subspace alignment procedure, while GWL [16] solves a Gromov-Wasserstein optimization problem to jointly find node embeddings and the graph matching. G-CREWE [13] uses graph compression to accelerate the matching

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557563

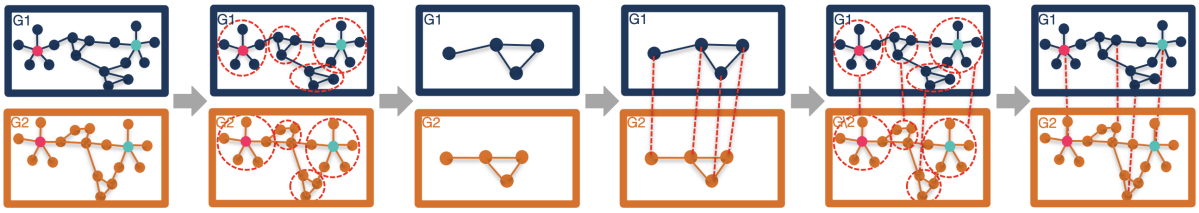


Figure 1: An illustrative example of CAPER. The pink and blue nodes in the leftmost figure have the same local structural similarity, so methods that overfit the local structural similarity may misalign them. But with the help of higher-order information (coarsened graphs in step 4), CAPER is able to eventually correctly align the pink nodes as well as the blue nodes.

Table 1: Comparing alignment meta-frameworks.

	General	Multiscale	Improves accuracy	Improves runtime
MOANA [19]	✗	✓	✗	✓
Boosting [9]	✗	✗	✓	✗
RefiNA [4]	✓	✗	✓	✗
CAPER	✓	✓	✓	✓

step of embedding-based alignment, though the embedding step is performed on the entire input graphs.

Network Alignment Meta-Frameworks. A few recent works have proposed meta-frameworks to improve unsupervised network alignment algorithms. This includes MOANA, the only other multilevel network alignment approach [19]. MOANA uses multiresolution matrix factorization to accelerate FINAL [18] (it produces negative-valued adjacency matrices that do not work with other network alignment methods) at the cost of some accuracy. RefiNA [4] makes the opposite tradeoff, enforcing greater local consistency to increase accuracy of several base methods at the cost of adding additional runtime. Another meta-framework [9] studies how design choices of recent embedding-based network alignment methods can be combined to increase accuracy via boosting. Meanwhile, our approach inherits all these benefits, as shown in Tab. 1.

3 PRELIMINARIES

Graphs. We consider two graphs G_1 and G_2 with nodesets $\mathcal{V}_1, \mathcal{V}_2$ and adjacency matrices $\mathbf{A}_1, \mathbf{A}_2$ containing edges between nodes. A graph G_i has a coarsened version $\tilde{G}_i$ with a smaller nodeset $\tilde{\mathcal{V}}_i$ of $\tilde{n} < n$ nodes. Each node in the original graph corresponds to a node in the coarsened graph, represented by an assignment matrix $\mathbf{P} \in \{0,1\}^{n \times \tilde{n}}$. For clarity, we drop the $\tilde{G}$ notation unless it is necessary to distinguish coarsened and uncoarsened versions.

Alignment. An alignment between the nodes of two graphs can be represented by a matrix $\mathbf{S}$, where s_{ij} is the (real-valued or binary) similarity between node i in G_1 and node j in G_2.

Problem Statement. Given two graphs G_1 and G_2 with meaningful node alignments, but none known *a priori*, we seek to shrink them into coarsened versions $\tilde{G}_1$ and $\tilde{G}_2$, and recover their alignment $\mathbf{S}$ from $\tilde{\mathbf{S}}$ obtained by aligning their coarser versions $\tilde{G}_1$ and $\tilde{G}_2$.

4 METHOD

Next, we detail our CAPER framework, the first general-purpose multilevel framework for unsupervised network alignment that can accommodate any base network alignment approach. It consists of four steps that are carefully designed in order to achieve higher accuracy and/or lower runtime compared to its base alignment methods: Coarsen, Align, Project, Refine (CAPER). In Fig. 1, we provide an example of how CAPER can implicitly enforce higher-order structural consistency that improves network alignment.

4.1 Graph Coarsening

Given an input graph G_i, we want to obtain a coarsened graph $\tilde{G}_i$ using grouping-based coarsening methods. We leverage the normalized heavy-edge matching (NHEM) heuristic [3] for graph coarsening. This approach repeatedly combines pairs of adjacent nodes into a supernode in decreasing order of degree-normalized edge weight [11], which for edge (u,v) with weight w_{uv} connecting nodes u and v with degrees d_u and d_v respectively is given by $w_{uv}/\sqrt{d_u d_v}$, until no node is left uncombined or the uncombined nodes do not have uncombined neighbors (isolated nodes). The resulting coarse graph consists of these supernodes, which share an edge if any of the nodes in one supernode shared an edge in the original graph with any of the nodes in the other supernode.

Graph coarsening turns each input graph G_i into a coarsened graph $\tilde{G}_i$. We iteratively repeat this coarsening procedure up to $\mathbf{L}$ times to produce a sequence of coarsened graphs $\tilde{G}_i^{(0)}, \ldots, \tilde{G}_i^{(L)}$, where the first level is the input graph ($\tilde{G}_i^{(0)} = G_i$), and the coarsest (smallest) graph is $\tilde{G}_i^{(L)}$. Assignments between nodes at consecutive levels $\ell - 1$ and ℓ are contained in a matrix $\mathbf{P}_i^{(\ell)}$ for $\ell \in [1, \ldots, L]$.

4.2 Alignment of Coarsened Graphs

We can apply any unsupervised network alignment method to align the nodes of the coarsest graphs $\tilde{G}_1^{(L)}$ and $\tilde{G}_2^{(L)}$ to produce a matching $\mathbf{S}^{(\mathbf{L})}$. We observe that the coarsening procedure sometimes generates slightly different numbers of nodes for the same graph even if the input graphs have the same size, so the proposed formulation must be able to handle graphs of different sizes. This can be done by adding singleton nodes to the smaller graph [1, 20].

4.3 Projection

We project the alignment solution at the coarsest level $\mathbf{S}^{(\ell)}$ to a mapping between the nodes at the next finer level using the assignment matrices: $\mathbf{S}^{(\ell-1)} = \mathbf{P}_1^{(\ell)^\top} \mathbf{S}^{(\ell)} \mathbf{P}_2^{(\ell)}$. Note that this solution is coarse, and all nodes in level $\ell - 1$ mapped to the same supernode in level ℓ will have the same match. Thus, we next show how to use the finer graph structure to refine this coarse solution.

4.4 Soft Refinement

Recent work for refining network alignment [4] operates on "hard" initial solutions, where each node is mapped to at most one other

Table 2: Dataset statistics: These four datasets represent various phenomena as shown in the description column.

Name	Nodes	Edges	Description
Arenas [8]	1,133	5,451	communication network
Hamsterster [8]	2,426	16,613	social network
Facebook [10]	4,039	88,234	social network
Magna [15]	1,004	8,323	protein-protein interaction

node. Here, we propose a new refinement operator that uses the "soft" initial alignments, which better models the various strengths of several potential matches for each node, as shown in Fig. 5. Given an initial soft (real-valued) alignment $\mathbf{S}$, we iteratively apply the update rule $\mathbf{S} = \text{NORMALIZE}(\mathbf{S} \circ \mathbf{A}_1\mathbf{S}\mathbf{A}_2 + \epsilon)$, where $\circ$ denotes Hadamard product, ϵ is a small positive minimum matching score to any pair of nodes to prevent over-reliance on the initially discovered matches (we set $\epsilon = 10^{-\lceil \log_{10} \max(n_1,n_2) \rceil}$) and NORMALIZE is a single round of row-wise then column-wise normalization, as in [4].

We iteratively apply this project-and-refine procedure between successive levels until we arrive back at the input level, giving us the mapping between nodes in the original graph.

4.5 Computational Complexity

We analyze the time complexity of CAPER as a function of the number of nodes n (to simplify notation we assume this is the same for both graphs), for sparse graphs with $O(n)$ edges. Then the complexity of our CAPER framework is $Lf_{\text{coarsen}}(n) + f_{\text{align}}(\frac{n}{2^L}) + L\Big(f_{\text{project}}(n) + f_{\text{refine}}(n)\Big)$. The coarsening time applied to each of L levels, $f_{\text{coarsen}}(n)$, is linear in the number of edges using heavy-edge matching [3], which is $O(n)$. Projection f_{project} and refinement f_{refine} consist of matrix multiplications that, by maintaining a sparse matching matrix, can also run in $O(n)$ time [4].

Meanwhile, with NHEM shrinking the graph by approximately a factor of 2 at each level [3], note that we are able to run the base alignment step on a smaller graph, incurring a runtime of $f_{\text{align}}(\frac{n}{2^L})$ as opposed to $f_{\text{align}}(n)$ by applying the base alignment algorithm to the full input graphs. Thus, CAPER can offer computational speedup particularly for slow base alignment methods, where f_{align} may be asymptotically large (such as $O(n^3)$),and the savings may outweigh the overhead of coarsening, projection, and refinement.

5 EXPERIMENTS

We first describe our experimental setup and the datasets and baseline methods used in our empirical analysis, and then show quantitative improvements from CAPER and a closer ablation study.

Data. We use simulated and real alignment scenarios on graphs representing various real-world phenomena (Tab. 2). Following prior works [5, 9, 18], we simulate a network alignment scenario with known ground truth: a graph with adjacency matrix $\mathbf{A}$ is aligned to a noisy permuted copy $\mathbf{A}^* = \overline{\mathbf{S}}\mathbf{A}\overline{\mathbf{S}}^\top$ and $\overline{\mathbf{S}}$, for which we generate a random permutation matrix $\overline{\mathbf{S}}$; we then randomly remove edges from $\mathbf{A}^*$ with probability $p \in [0.05, 0.10, 0.15, 0.20, 0.25]$. The MAGNA [15] networks are protein-protein interaction (PPI) networks that are aligned to versions of themselves with various percentages of low-confidence PPIs (edges) added; thus, all edges in this graph represent real-world phenomena and we do not need to synthesize an alignment scenario.

Table 3: Number of nodes in the coarsened graph after 2–4 levels of coarsening.

Name	2	3	4
Hamsterster	1,288	702	418
Facebook	2,078	1,078	572

Baselines. We use **(1)** FINAL [18] and **(2)** REGAL [5], which are popular unsupervised network alignment methods that have usable public codebases and represent different classes of techniques (optimization and node embeddings), demonstrating the wide applicability of our framework. We also use a more recent approach, **(3)** GWL [16], which combines optimization and node embeddings, and achieves good accuracy but has slow runtime due to its $O(n^3)$ computational complexity. Moreover, we consider the post hoc refinement method RefiNA applied to each of the network alignment methods: **(4)** FINAL-RefiNA, **(5)** REGAL-RefiNA and **(6)** GWL-RefiNA. Additionally, we use **(7)** MOANA [19] as a baseline, the only other multilevel network alignment method.

For FINAL's prior alignment information, we take the top $k = \lfloor \log_2 n \rfloor$ most similar nodes by degree for each node [1, 5]. We set other parameters for REGAL [5] and GWL [16] using the defaults recommended by the authors.

CAPER variants. We test variants of CAPER using each base alignment method: CAPER(FINAL), CAPER(REGAL), and CAPER(GWL). We use 3 coarsening levels and 100 refinement iterations, as in [4], to balance accuracy and computational efficiency (we found that more refinement may increase performance if that is desired and increased runtime is acceptable).

Evaluation. We measure **alignment accuracy**, or the proportion of correctly aligned nodes, and **runtime**.

5.1 Alignment Accuracy

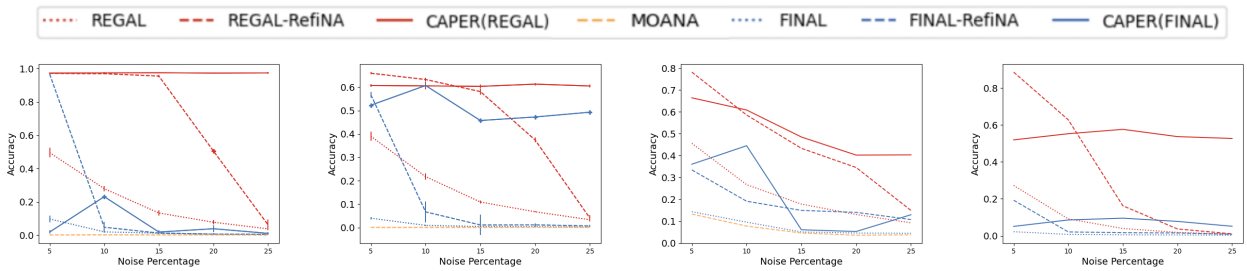
Setup. In Fig. 2, we report the average accuracy and standard deviation (+ sign: standard deviation > 0.05) over five trials for each setting, except for Magna where we do not simulate alignments.

Results. While the existing multilevel alignment method, MOANA, has accuracy below its single-level counterpart FINAL as expected, our multilevel framework, CAPER, significantly outperforms different base alignment methods as well as their single-level refined variants using RefiNA. Moreover, we can see that **CAPER is more robust to noise** due to the multilevel consistency that it encourages; this is especially notable for CAPER(REGAL) whose performance is very stable even when the noise level increases.

5.2 Alignment Runtime

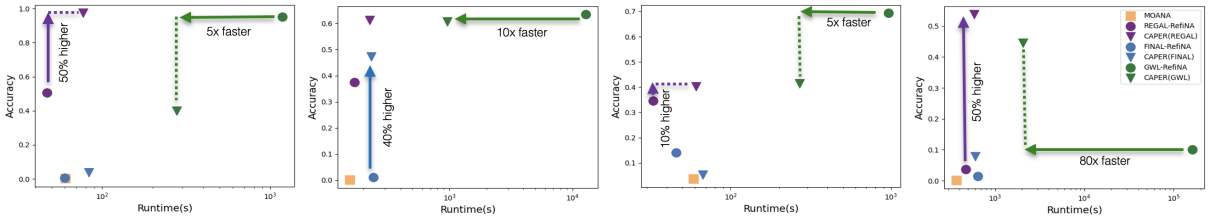
Setup & Evaluation. Due to GWL's slow runtime, we only run it for one trial on the largest Facebook dataset. Others are averaged over five trials in Fig. 3.

Results. For faster base methods such as FINAL and especially REGAL, our improvements are mainly in accuracy (up to 50% higher accuracy); the computational savings of performing the alignment on smaller graphs does not outweigh the overhead of coarsening and refinement when the base alignment method is fast. However, when the base alignment method is slow, as is the case for GWL, our framework results in considerable computational savings (5-80×



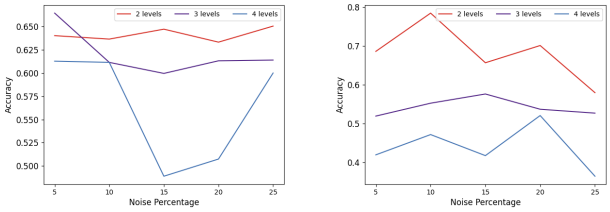
(a) Arenas (b) Hamsterster (c) Magna (d) Facebook

Figure 2: Accuracy (solid lines) vs. different noise levels. CAPER outperforms baselines, particularly as noise increases.



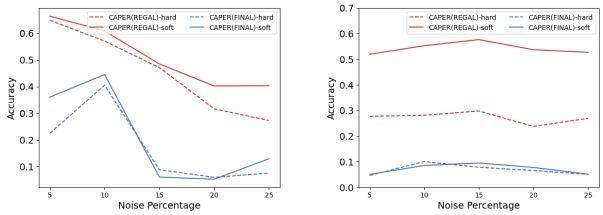
(a) Arenas (b) Hamsterster (c) Magna (d) Facebook

Figure 3: Accuracy vs. runtime for CAPER and RefiNA for 20% noise. CAPER yields better accuracy for FINAL and REGAL by enforcing higher-order consistency. For GWL, CAPER runs up to 80x faster because the alignment is run on smaller graphs.



(a) Hamsterster (b) Facebook

Figure 4: Sensitivity to number of coarsening levels for CAPER(REGAL). In general, 2 levels leads to highest accuracy. We use 3 levels for the best accuracy/runtime tradeoff.



(a) Magna (b) Facebook

Figure 5: Sensitivity to soft/hard refinement for CAPER(REGAL) and CAPER(FINAL). Soft refinement works significantly better, especially for accurate base methods.

faster), while largely preserving accuracy. Compared with MOANA, CAPER achieves better accuracy.

5.3 Sensitivity Analysis

Number of levels. Figure 4 compares the performance of CAPER(REGAL) with different numbers of coarsening levels on the Hamsterster and Facebook datasets. The number of coarsening levels leads to a tradeoff between accuracy and runtime: more coarsening leads to smaller coarsened graphs (Tab. 3) and faster runtime at a cost of some accuracy. For our main experiments, we used 3 levels of coarsening for all datasets to balance this tradeoff, and could use 2 levels to achieve even higher accuracy.

Hard vs. soft refinement. In Fig. 5, we see improvement from our refinement of more expressive "soft" alignments (§ 4.4), most noticeably for the base method REGAL. For FINAL, because its initial solution is less accurate on these datasets, we used hard refinement when operating directly on its solution (at the coarsest level) and soft refinement at subsequent levels. This also explains the smaller gap in performance.

6 CONCLUSION

We describe the first general-purpose multilevel framework for unsupervised network alignment. It works with various base network alignment algorithms, making them more accurate and robust by incorporating multiscale graph information, and accelerating the runtime by allowing them to operate on smaller input graphs. However, not all coarsening methods work well. Some recent spectral coarsening methods [2] will give clusters with zero nodes and thus our multi-level alignment framework could fail. One possible future direction is to characterize the effect of various coarsening methods on multilevel network alignment.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, Lawrence Livermore National Security, LLC. and was supported by the LLNL-LDRD Program under Project No. 21-ERD-012. Jing Zhu was an intern at Lawrence Livermore National Laboratory while working on this project. It was also partially supported by the NSF under Grant No. IIS 1845491, and Amazon and Facebook faculty awards.

REFERENCES

[1] Xiyuan Chen, Mark Heimann, Fatemeh Vahedian, and Danai Koutra. Cone-align: Consistent network alignment with proximity-preserving node embedding. In *CIKM*, 2020.

[2] Chenhui Deng, Zhiqiang Zhao, Yongyu Wang, Zhiru Zhang, and Zhuo Feng. Graphzoom: A multi-level spectral approach for accurate and scalable graph embedding. In *ICLR*, 2020.

[3] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.

[4] Mark Heimann, Xiyuan Chen, Fatemeh Vahedian, and Danai Koutra. Refining network alignment to improve matched neighborhood consistency. In *SDM*, 2021.

[5] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *CIKM*, 2018.

[6] Junchen Jin, Mark Heimann, Di Jin, and Danai Koutra. Toward understanding and evaluating structural node embeddings. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(3):1–32, 2021.

[7] Ehsan Kazemi. Network alignment: Theory, algorithms, and applications. Technical report, EPFL, 2016.

[8] Jérôme Kunegis. Konect: the koblenz network collection. In *WWW*, 2013.

[9] Alexander Frederiksen Kyster, Simon Daugaard Nielsen, Judith Hermanns, Davide Mottin, and Panagiotis Karras. Boosting graph alignment algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3166–3170, 2021.

[10] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[11] Jiongqian Liang, Saket Gurukar, and Srinivasan Parthasarathy. Mile: A multi-level framework for scalable graph embedding. In *ICWSM*, 2021.

[12] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. *ACM computing surveys (CSUR)*, 51(3):1–34, 2018.

[13] Kyle K Qin, Flora D Salim, Yongli Ren, Wei Shao, Mark Heimann, and Danai Koutra. G-crewe: Graph compression with embedding for network alignment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1255–1264, 2020.

[14] Ryan A Rossi, Di Jin, Sungchul Kim, Nesreen K Ahmed, Danai Koutra, and John Boaz Lee. On proximity and structural role-based embeddings in networks: Misconceptions, techniques, and applications. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(5):1–37, 2020.

[15] Vikram Saraph and Tijana Milenković. Magna: maximizing accuracy in global network alignment. *Bioinformatics*, 30(20):2931–2940, 2014.

[16] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-wasserstein learning for graph matching and node embedding. In *ICML*, pages 6932–6941, 2019.

[17] Yujun Yan, Jiong Zhu, Marlena Duda, Eric Solarz, Chandra Sripada, and Danai Koutra. GroupINN: Grouping-based interpretable neural network for classification of limited, noisy brain data. In *KDD*, pages 772–782, 2019.

[18] Si Zhang and Hanghang Tong. Final: Fast attributed network alignment. In *KDD*, 2016.

[19] Si Zhang, Hanghang Tong, Ross Maciejewski, and Tina Eliassi-Rad. Multilevel network alignment. In *The World Wide Web Conference*, pages 2344–2354, 2019.

[20] Zhen Zhang, Yijian Xiang, Lingfei Wu, Bing Xue, and Arye Nehorai. KerGM: Kernelized Graph Matching. In *NeurIPS19*, pages 3330–3341, 2019.