Spectral Subgraph Localization

Ama Bembua Bainson[*] Aarhus University ama@cs.au.dk
Amit Boyarski[*] Technion – IIT amitboy1000@gmail.com
Judith Hermanns[*] Aarhus University judith@cs.au.dk
Petros Petsinis[*] Aarhus University petsinis@cs.au.dk
Niklas Aavad Aarhus University 202008561@post.au.dk
Casper Dam Larsen Aarhus University 202008583@post.au.dk
Tiarnan Swayne Aarhus University 202009717@post.au.dk
Davide Mottin Aarhus University davide@cs.au.dk
Alex M. Bronstein Technion – Israel Institute of Technology bron@cs.technion.ac.il
Panagiotis Karras U. of Copenhagen & Aarhus U. piekarras@gmail.com

Abstract

Several graph analysis problems are based on some variant of *subgraph isomorphism*: Given two graphs, G and Q, does G contain a subgraph isomorphic to Q? As this problem is $\mathbf{NP}$-complete, past work usually avoids addressing it explicitly. In this paper, we propose a method that *localizes*, i.e., finds the best-match position of, Q in G, by aligning their Laplacian spectra and enhance its stability via bagging strategies; we relegate the finding of an exact node correspondence from Q to G to a subsequent and separate *graph alignment* task. We demonstrate that our localization strategy outperforms a baseline based on the state-of-the-art method for graph alignment in terms of accuracy on real graphs and scales to hundreds of nodes as no other method does.

1 Introduction

Graph analysis tasks frequently require *localizing* a smaller target graph Q within a larger source graph G, i.e., finding an induced subgraph of G best aligned with Q. This type of problem may appear as *subgraph discovery* [1, 2], where we seek a target graph in G, as *subgraph querying* [3, 4], where we find out whether a target subgraph match exists within a collection of source graphs, or as *graph matching* [5–7], where we have to align corresponding nodes across two graphs, potentially of different sizes; the problem finds practical application in detecting sub-molecules in bigger molecules [8], localizing parts of shapes in computational geometry [9], and detecting an electronic subcircuit within a larger one [10] by sampling multiple subgraphs and comparing the spectra of their adjacency matrices to that of the query subgraph. Despite the problem's prevalence, past research has avoided tackling it directly due to its $\mathbf{NP}$-hardness.

In this paper, we propose a *spectral* solution to the problem of *subgraph localization*, built by identifying the spectrum $\boldsymbol{\lambda}_Q$ of a graph Q within that of another graph G. Figure 1 visualizes an instance of the subgraph localization problem by our formulation; we aim to find a function δ that indicates which nodes in G correspond to Q. Our solution effectively recovers both the nodes belonging to the part and the edges that connect the part to the graph. This problem is an instance of *inverse eigenvalues* problems [11], the class of problems which aim to reconstruct a

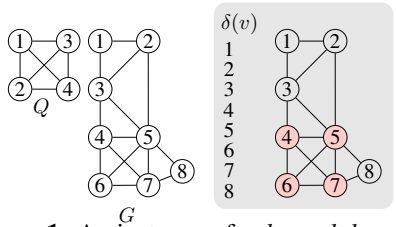


Figure 1: An instance of *subgraph localization* (left) and its solution (right).

[*]Equal contribution.

Bainson et al., Spectral Subgraph Localization. *Proceedings of the Second Learning on Graphs Conference (LoG 2023)*, PMLR 231, Virtual Event, November 27–30, 2023.

matrix from its spectrum. Further, we go beyond this one-off optimization-based solution to enhance the method's stability by applying a bagging-like strategy. Our experimental study demonstrates that our approach tackles the subgraph localization problem more effectively than state-of-the-art competitors.

In summary, our contributions are as follows:

- We propose a spectral formulation of and solution to the subgraph localization problem (Sec. 4).
- We enhance the stability of the method via customized bagging-like strategies (Sec. 5).
- We experimentally validate the effectiveness of our solution on real and synthetic graphs (Sec. 6).

2 Related Work

The *subgraph isomorphism* problem is to decide whether a source graph contains a target subgraph and return that exact subgraph. This problem is mainly solved for very small target subgraphs (≤ 10 nodes) and aims at exact matches. Several methods speed up this process by exploiting query specifics, such as patterns in multiple subgraph queries [12]. By contrast, our method aims at bigger target subgraphs. In *subgraph querying*, the goal is to identify all source graphs among a collection that contain a query target subgraph, without necessarily indicating its position [3, 4, 13]. The goal of *subgraph matching* is to match the nodes of a smaller graph to those of a subgraph in a bigger graph. Many methods for graph matching effectively solve subgraph isomorphism, though not specifically designed for this purpose [5]. Recent work [14, 15] employs deep neural models to learn *node embeddings* [16] subsequently used for matching. Yet, these methods require training on (query, target) pairs. The problem of *subgraph localization* seeks a good fit, by some measure [17] of a target subgraph within a bigger source graph. This problem has been scarcely studied. A recent application in computer vision [18] uses subgraph localization to detect temporal actions. However, this model uses edges for temporal aspects and inter-scene relations, and hence does not generalize to arbitrary graphs. An existing spectral solution [19] is limited to special families of graphs, such as cliques.

3 Subgraph localization

All aforementioned problems have in common the search for one graph within another. We study the most generic form of this problem, which corresponds to the problem named *Subgraph localization* in our previous discussion. That is, we aim to identify a subset of the nodes of a graph G corresponding to an input graph Q; we do not aim at an exact 1-to-1 correspondence among all graph elements, but to simply detect a set of best matches.

Problem 1. *The* subgraph localization *problem for a graph $G = \langle V, E\rangle$, where V is a set of n nodes and $E \subseteq V \times V$ is a set of edges, and a query graph $Q = \langle V_Q, E_Q\rangle$ with $n_Q = |V_Q|$, $n_Q < n$, calls to find a set of nodes $V_S \subset V$, inducing a set of edges $E_S \subset E$, such that $|V_S| = |V_Q|$ and there exists a bijective function $f : V_S \to V_Q$ between the nodes in V_S and those in V_Q such that for each $(i, j) \in E_S$ there exists $(f(i), f(j)) \in E_Q$ and vice versa.*

In many applications solving subgraph localization, we do not need to explicitly materialize the correspondence function f. Such a one-to-one correspondence is not explicitly sought for. Thus, we can eschew recovering an exact f and instead aim at finding an indicator function $\delta : V \to \{0, 1\}$ such that $\delta(v) = 1$, if $v \in V_Q$, and $\delta(v) = 0$ otherwise.

At first glance, finding such an indicator function seems easier than recovering a bijective function f. However, even in this identity-function formulation, the problem corresponds to the decision version of the subgraph isomorphism problem, which asks whether a graph G contains a subgraph isomorphic to another graph Q. Thus, the problem is still **NP**-complete. Even so, we further relax our requirements, allowing the function δ to be a binary version of a continuous real-value function $\mathbf{v} : V \to \mathbb{R}$ on values below a threshold τ, whereby $\delta(v) = 1$ only if $\mathbf{v}(v) < \tau$ and $\delta(v) = 0$ otherwise.

This relaxed problem calls to find a real function, or, equivalently, a real vector $\mathbf{v} \in \mathbb{R}^n$, with $n = |V|$, for a known permutation of nodes. To overcome the requirement for a known node permutation, we consider a permutation-invariant spectral alignment approach reminiscent of the Hamiltonian operator used in shape analysis [9, 20]. Before delving into the approach, we introduce the notation.

Background. The *adjacency matrix* of graph G with n nodes is a $n\times n$ matrix $\mathbf{A}\in\{0,1\}^{n\times n}$ where $\mathbf{A}_{ij}=1$ if $(i, j) \in E$, 0 otherwise. The *degree matrix* $\mathbf{D}$ is an $n\times n$ diagonal matrix where

each entry $d_{ii}=\sum_{j\neq i}\mathbf{A}_{ij}$ is the degree of node i. The *Laplacian matrix* is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The Laplacian matrix of undirected graphs is symmetric and positive semi-definite, hence its eigenvalues $\lambda_1, \ldots, \lambda_n$, are real and non-negative. The *spectrum* $\boldsymbol{\lambda}(\mathbf{M})$ of a matrix $\mathbf{M}$ is the ordered sequence $\lambda_1 \leq \ldots \leq \lambda_n$ of its eigenvalues. A graph's spectrum is that of its Laplacian matrix.

4 Spectral Subgraph localization

We examine how the presence of a subgraph within a graph affects the graph's spectrum. Spectral theory establishes that the spectrum of a subgraph interlaces with the spectrum of the graph. However, the problem is also non-trivially affected by nodes beside the subgraph. Still, if we could compensate for the effect of nodes other than the subgraph's nodes, the two spectra would be indistinguishable. Following this reasoning, we devise a novel objective for subgraph localization. To that end, we first propose an original connection between subgraph localization and inverse eigenvalue problems with structural constraints [11].

Inverse Eigenvalue Problem. The general *additive inverse eigenvalue problem* (AIEP) is defined as follows:

Problem 2 (AIEP, Problem 3.6 in [11])**.** *Given an $n \times n$ matrix $\mathbf{A}$, a special class of matrices $\mathcal{N}$, and a set of scalars $\{\lambda_{Q_i}\}_{i=1}^{k}$, find $\mathbf{X} \in \mathcal{N}$ such that $\{\lambda(\mathbf{A}+\mathbf{X})_i\}_{i=1}^{k} = \{\lambda_{Q_i}\}_{i=1}^{k}$.*

A vast literature on this problem (see [11] and references therein) explores questions regarding the existence of solutions and numerical approximation algorithms for various special classes of matrices $\mathcal{N}$. A common variant of Problem 2 expresses the problem as a least squares problem between the spectra:

$$\min_{\mathbf{X}\in\mathcal{N}} \|\boldsymbol{\lambda}(\mathbf{A}+\mathbf{X}) - \boldsymbol{\lambda}_Q\|^2. \tag{1}$$

We establish a connection between the subgraph localization problem (Problem 1) and the additive inverse eigenvalue problem (Problem 2). Under the above formulation, we aim to find a $\mathbf{v}$ that, added to the diagonal of the Laplacian of G, renders its first n_Q eigenvalues equal to those of the query graph. In addition to finding $\mathbf{v}$, we aim to remove from G the edges that connect the identified part to the remaining nodes. To the best of our knowledge, this is the first time such a connection has been established, and the first time an AIEP with structural Laplacian constraints is considered.

For simplicity, we commence with an intuitive scenario although our solution applies to any graph G and any query Q without prejudice. We assume that G has a number of clearly separated communities, one of which corresponds to the query graph Q. A community is defined by a cut, as nodes within the same community are more well connected than nodes across communities. Without loss of generality, assume the graph comprises two distinct communities. In this case, G's Laplacian is a block matrix with two *diagonal* blocks $\mathbf{L}_{11}\in\mathbb{R}^{n_Q\times n_Q}$ and $\mathbf{L}_{22}\in\mathbb{R}^{(n-n_Q)\times(n-n_Q)}$ and a few entries in the blocks $\mathbf{L}_{12}\in\mathbb{R}^{n_Q\times(n-n_Q)}$ and $\mathbf{L}_{21}\in\mathbb{R}^{(n-n_Q)\times n_Q}$ representing edges across the two communities. The spectra $\boldsymbol{\lambda}(\mathbf{L})$ of G and $\boldsymbol{\lambda}(\mathbf{L}_Q)$ of Q differ on the nodes in $\mathbf{L}_{22}$ and the edges in $\mathbf{L}_{12}$ and $\mathbf{L}_{21}$.

To cancel out this difference, we aim to transform $\mathbf{L}$ to a Hamiltonian [20] operator $\mathcal{H} = \mathbf{L} + \mathrm{diag}(\mathbf{v})$, where $\mathbf{v} : V \to \mathbb{R}$ is a scalar real-valued function and $\mathbf{L}$ is the Laplacian. The Hamiltonian reduces to the Laplacian if the potential is $\mathbf{0}$. According to [9, Lemma 1], if we add to the diagonal of $\mathbf{L}$ a vector $\mathbf{v}$ having non-zero values, $\mathbf{v}(v) \geq \tau$, *limited to* nodes in $\mathbf{L}_{22}$, i.e., outside V_Q, then eigenvectors corresponding to eigenvalues $\lambda_i < \tau$ of the resulting spectrum $\boldsymbol{\lambda}(\mathbf{L} + \mathrm{diag}(\mathbf{v}))$ will have non-zero values *limited to* the positions corresponding to nodes in V_Q, in effect rendering $\boldsymbol{\lambda}(\mathbf{L} + \mathrm{diag}(\mathbf{v}))$ similar to $\boldsymbol{\lambda}(\mathbf{L}_Q)$. Still, the non-zero entries between communities in $\mathbf{L}_{12}, \mathbf{L}_{21}$ affect the spectrum. To cancel that effect, we introduce a *Laplacian editing matrix* that removes the contribution of such edges to the Laplacian of the graph G:

$$\mathbf{E} = \begin{bmatrix} -\mathrm{diag}(\mathbf{L}_{12}\mathbf{1}) & \mathbf{L}_{12} \\ \mathbf{L}_{21} & -\mathrm{diag}(\mathbf{L}_{21}\mathbf{1}) \end{bmatrix}$$

where $\mathbf{L}_{12}\mathbf{1}$ ($\mathbf{L}_{21}\mathbf{1}$) corrects node degrees for the removal of edges in $\mathbf{L}_{12}$ ($\mathbf{L}_{21}$). In effect, the corrected Laplacian $\mathbf{L} - \mathbf{E}$ is equivalent to the Laplacian of a graph with two connected components, one of which isomorphic to the query graph Q. Thus, the solution $\mathbf{v}$ renders the $|V_Q|$ smallest eigenvalues of the corrected Laplacian indistinguishable from the spectrum of Q, $\boldsymbol{\lambda}_Q$, i.e., $\boldsymbol{\lambda}(\mathbf{L} -$

$\mathbf{E} + \mathrm{diag}(\mathbf{v})) = \boldsymbol{\lambda}_Q$, where, with a slight abuse of notation, $\boldsymbol{\lambda}(\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v}))$ refers to the $|V_Q|$ smallest eigenvalues of $\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v})$. Since both $\mathbf{v}$ and $\mathbf{E}$ are unknown, we optimize the objective:

$$\min_{\mathbf{v},\mathbf{E}} \ \|\boldsymbol{\lambda}(\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v})) - \boldsymbol{\lambda}_Q\|_2^2 \quad \text{s.t. } \mathbf{E} = \mathbf{E}^\top, \mathbf{E}\mathbf{1} = \mathbf{0},\ \mathrm{off}(\mathbf{L} - \mathbf{E}) \leq 0,\ \|\mathbf{v}\| = c. \tag{2}$$

This objective is not convex, yet it only depends on the spectrum, for which there exists efficient approximations [21]; it leads to a solution even if the initial value of $\mathbf{v}$ is a noisy version of the ground truth. As constraints, we postulate that $\mathbf{E}$ should be: (i) symmetric, $\mathbf{E} = \mathbf{E}^\top$; (ii) row- (and, by symmetry, also column-) centered, $\mathbf{E}\mathbf{1} = \mathbf{0}$, with every row summing to 0; and (iii) yielding only non-positive off-diagonal entries $\mathrm{off}(\mathbf{L} - \mathbf{E}) \leq 0$. In addition, we enforce that $\mathbf{v}$ be a point on the surface of a sphere of radius c, via the constraint $\|\mathbf{v}\| = c$. Proposition 4.1 provides a sufficient condition on c for the optimality of Equation (2), considering the noiseless case where G exactly contains the subgraph Q.

Proposition 4.1. *When $c > \sqrt{n - n_Q}\max(\boldsymbol{\lambda}_Q)$, the global optimum of Equation (2) is obtained at*

$$\mathbf{v} = \begin{cases} 0 & \text{if } v_i \in V_Q \\ \frac{c}{\sqrt{n - n_Q}} & \text{otherwise} \end{cases} \quad \text{with } \tilde{\mathbf{v}} = \frac{\mathbf{v} - \min(\mathbf{v})}{\max(\mathbf{v}) - \min(\mathbf{v})}, S_{ij} = |\tilde{v}_i - \tilde{v}_j| A_{ij}, \mathbf{E} = \mathrm{diag}(\mathbf{S}\mathbf{1}) - \mathbf{S} \tag{3}$$

Proof. Let $\mathbf{E}$ be constructed from Equation 3. $\mathbf{L} - \mathbf{E}$ is the Laplacian of a graph composed of two disjoint components, one of which is exactly the component indicated by Equation 3, i.e., the query subgraph Q. Then there is a permutation $\boldsymbol{\Pi}$ such that $\boldsymbol{\Pi}\mathbf{L}\boldsymbol{\Pi}^\top$ is a block diagonal matrix with the Laplacian of each component on the diagonal. Without loss of generality, we assume that the Hamiltonian operator attains this block diagonal form:

$$\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v}) = \begin{bmatrix} \mathbf{L}_Q & \\ & \mathbf{L}_{\bar{Q}} + \frac{c}{\sqrt{n - n_Q}}\mathbf{1} \end{bmatrix}. \tag{4}$$

When c satisfies the stated condition, the spectrum of the bottom-right block contains only eigenvalues larger than $\max(\boldsymbol{\lambda}_Q)$. It follows that the first n_Q eigenvalues of $\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v})$ are exactly those of $\mathbf{L}_Q$, rendering the objective of Equation 2 equal to zero. □

In effect, by Proposition 4.1, we can recover the optimal solution if $\mathbf{v}$ is appropriately normalized and c is no less than a certain value. We exploit this result in Section 4.3 to design our algorithm by numerical optimization. We first introduce a regularization term.

Regularization. The objective in Equation 2 does not prevent $\mathbf{v}$ from taking arbitrary values. However, since $\mathbf{L} - \mathbf{E}$ has two connected components, $\mathbf{v}$ plays a role similar to that of Fiedler's vector in the minimization of the normalized cut [22]. This observation leads us to the *spectral regularization* $\mathbf{v}^\top(\mathbf{L} - \mathbf{E})\mathbf{v}$ that exhorts $\mathbf{v}$ to take values in the null-space of $\mathbf{L} - \mathbf{E}$. In other words, the spectral regularizer drives $\mathbf{v}$ to be a stepwise function. We combine the spectral regularization with our objective as follows:

$$\begin{aligned} \min_{\mathbf{v},\mathbf{E}} \underbrace{\|\boldsymbol{\lambda}(\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v})) - \boldsymbol{\lambda}_Q\|_2^2}_{\text{Data term}} + \mu \underbrace{\mathbf{v}^\top(\mathbf{L} - \mathbf{E})\mathbf{v}}_{\text{Spectral regularizer}} \\ \text{s.t. } \mathbf{E} = \mathbf{E}^\top, \mathbf{E}\mathbf{1} = \mathbf{0},\ \mathrm{off}(\mathbf{L} - \mathbf{E}) \leq 0,\ \|\mathbf{v}\| = c \end{aligned} \tag{5}$$

where $\mu \geq 0$ is a regularization coefficient.

Corollary. *Proposition 4.1 applies also with the spectral regularization term in Equation (5).*

Proof. Let $\mathbf{E}$ be constructed from Equations (3). $\mathbf{L} - \mathbf{E}$ is the Laplacian of a graph composed of two disjoint components, one of which is exactly indicated by Equation (3), i.e., the query subgraph Q. Then $\mathbf{v}$ in Equation (3) belongs to the null-space of $\mathbf{L} - \mathbf{E}$, rendering the regularization term 0, hence Equation (3) also provides the global minimum of Equation (5). □

4.1 Connection to partial shape localization

Equation (2) bears a connection with the Hamiltonian used for partial shape localization [9]. In particular, [9, Lemma 1] states that the eigenvectors of the Hamiltonian $\Delta_{\mathcal{X}} + \mathbf{v}$ of the Laplace-Beltrami operator $\Delta_{\mathcal{X}}$ on the Riemannian manifold $\mathcal{X}$, for large threshold τ, localizes the query region $\mathcal{R}$. To translate this operation to the cases of graphs, we substitute the Laplace-Beltrami operator with the discrete positive semi-definite Laplacian matrix $(\mathbf{L} - \mathbf{E})$ and add a potential function $\mathrm{diag}(\mathbf{v})$, inducing the Hamiltonian operator $(\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v}))$. Our results extend [9, Lemma 1] to Equation (5), which we constrain through regularization and the constraint in Proposition 4.1.

4.2 Localizing disconnected subgraphs

A special case of subgraph localization is that of a graph with a number of connected components, one of which corresponds to the query graph Q. In this case the editing matrix $\mathbf{E} = \mathbf{0}$, leading to the simpler objective:

$$\min_{\mathbf{v}} \|\boldsymbol{\lambda}(\mathbf{L} + \mathrm{diag}(\mathbf{v})) - \boldsymbol{\lambda}_Q\|_2^2 + \mu \mathbf{v}^\top \mathbf{L}\mathbf{v} \quad \text{s.t.} \quad \|\mathbf{v}\| = c. \tag{6}$$

4.3 Numerical optimization

We exploit Proposition 4.1 to craft a numerical procedure that minimizes the objective in Equation (2), collaterally optimizing for $\mathbf{E}$ and $\mathbf{v}$. In the first iteration $q = 0$, we initialize $\mathbf{E}_q = \mathbf{0}$. In iteration $q+1$ we minimize $f(\mathbf{v}, \mathbf{E}_q) = \|\boldsymbol{\lambda}(\mathbf{L} - \mathbf{E}_q + \mathrm{diag}(\mathbf{v})) - \boldsymbol{\lambda}_Q\|_2^2 + \mu \mathbf{v}^\top (\mathbf{L} - \mathbf{E}_q)\, \mathbf{v}$ for $\mathbf{v}$ given $\mathbf{E}_q$:

$$\mathbf{v}_{q+1} = \underset{\mathbf{v}:\|\mathbf{v}\|=c}{\arg\min}\, f(\mathbf{v}, \mathbf{E}_q), \tag{7}$$

via *projected gradient descent*, until convergence; an iteration $k+1$ of projected gradient descent performs the step:

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \alpha \nabla_{\mathbf{v}} f(\mathbf{v}, \mathbf{E}_q) \qquad \mathbf{v}_{k+1} = c\frac{\mathbf{x}_k}{\|\mathbf{x}_k\|}, \tag{8}$$

where $\alpha > 0$ regulates the learning rate. The gradient $\nabla_{\mathbf{v}}$ for Equation 8 requires a *differentiable eigendecomposition*, which is achievable by extant methods [23].

We subsequently update $\mathbf{E}$ according to:

$$\tilde{\mathbf{v}} = \frac{\mathbf{v}_q - \min(\mathbf{v}_q)}{\max(\mathbf{v}_q) - \min(\mathbf{v}_q)} \qquad S_{ij} = |\tilde{v}_i - \tilde{v}_j| A_{ij} \qquad \mathbf{E}_{q+1} = \mathrm{diag}(\mathbf{S1}) - \mathbf{S}. \tag{9}$$

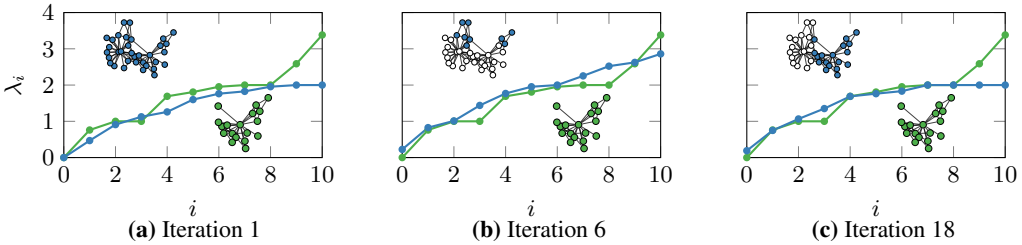
We obtain a threshold τ of the indicator function $\delta(\mathbf{v})$ for the nodes comprising the subgraph by splitting the elements of $\mathbf{v}$ into two clusters minimizing sum-of-squares error from the mean (i.e., optimizing the k-means objective in one dimension) and compute the matrix $\mathbf{E}$ from this thresholded $\mathbf{v}$ by Equations 9.

The SSL algorithm. We eventually present our *Spectral Subgraph Localization* (SSL) algorithm (Algorithm 1 in the supplementary material) for Problem 1. SSL takes as input the adjacency matrix $\mathbf{A}$ of the full graph G and the spectrum of a query subgraph, and returns the vector $\mathbf{v}$ and the threshold τ of the indicator function δ; it additionally requires some hyperparameters, such as the number of outer iterations $\texttt{maxiter}_{\texttt{out}}$, the number of inner iterations $\texttt{maxiter}_{\texttt{in}}$, the learning rate α, and the regularization coefficient μ. We empirically found that the number of iterations and the learning rate do not significantly affect results across datasets if chosen within some range; we report those ranges and recommended values in Table 1 in the supplementary material. On the other hand, the regularization coefficient μ in Equation 5 requires tuning for each dataset. We thus first normalize the value of μ by c^2 to remove the dependency on $\mathbf{v}$'s magnitude and then perform grid search on a range of values for μ to select an appropriate value.

The optimization process alternates the projected gradient optimization in Equation 8 and the update of $\mathbf{E}$ using Equations 9 until it converges or reaches the maximum number of iterations $\texttt{maxiter}_{\texttt{out}}$. Figure 2 illustrates the solution's progressive convergence through iterations.

4.4 Complexity Analysis

We derive the worst-case time complexity of the algorithm in the number of nodes n in the graph G. The eigendecomposition in Equation 8 takes $\mathcal{O}(n^3)$ per iteration; the computation in Equation 9



(a) Iteration 1 **(b)** Iteration 6 **(c)** Iteration 18

Figure 2: Alignment of the spectrum $\boldsymbol{\lambda}_Q$ of Q and the part of the spectrum $\boldsymbol{\lambda}(\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v}))$ of G corresponding to Q at 1, 6, and 18 iterations; the two spectra progressively approach each other, and thereby Q gets localized in G.

takes $\mathcal{O}(n^2)$ for the matrix-vector multiplication; 1-D k-means in Line 9 takes $\mathcal{O}(n)$ with the best algorithm [24]. In effect, the total time is $\mathcal{O}(\texttt{maxiter}_{\texttt{out}} \cdot (\texttt{maxiter}_{\texttt{in}} * n^3 + n^2 + n))$, where the $\mathcal{O}(n^3)$ term dominates. However, as $\mathbf{L} - \mathbf{E} + \mathrm{diag}(\mathbf{v})$ is a graph's Laplacian, its spectrum can be efficiently approximated through sampling [21].

5 Improving SSL stability

The method we have developed so far strives to optimize the objective in Equation eq5. Yet, due to the non-convex nature of that objective, it may converge to a local minimum (see Figure 8 in the Supplementary material). To prevent this disposition, we propose a *sampling-and-aggregation* methodology that resembles *bagging*. Bagging improves the stability of a predictor by learning the model's parameters on k samples of the data and then returning an aggregate prediction. Similarly, we introduce effective sampling and aggregation schemes that improve the SSL results on all datasets while incurring a constant time cost.

5.1 Edge sampling

An ideal solution based on edge sampling would exclude edges between the query graph Q and the rest of the graph, thus achieve low *conductance* [25] between them and easily localize the query. However, we do not know location of such edges in advance; that is, after all, the unknown of our problem. Therefore, we resort to sampling $t < n$ edges *uniformly* from the input graph G, to obtain a subgraph $G_i = (V_i, E_i)$, where $E_i \subset E$. We repeat the sampling process k times and obtain graphs $G_1, \ldots G_k$ graphs.

5.2 Results aggregation

We run SSL on each sampled graph $G_1, \ldots G_k$ with the same query graph Q, to obtain indicator functions $\delta_1, \ldots, \delta_k$. We regard each indicator function as a voter and devise the following three aggregation strategies to compute the final solution V_S.

Threshold: In the threshold strategy, we count the occurrences $d(v) = \sum_{i=1}^{k} \delta_i(v)$ of each node in the solutions SSL yields from each sampled graph $G_1, \ldots G_k$. We include in the final solution V_S each node v that appears in at least a fraction $\theta \in [0, 1]$ of the solutions, i.e., $\frac{d(v)}{k} \geq \theta$.

Neighborhood: The threshold strategy does not enforce any constraint on the solution V_S. Yet, in most cases the query graph Q is connected or contains only a few connected components. In such cases, the solution induced by V_S should also be connected. We utilize a greedy strategy that incrementally includes neighboring nodes in the solution it constructs to enforce connectivity. Starting from an initial highly occurring node v, we include in the solution the neighbor u of v having the most occurrences, i.e., $\arg\max_{u \in N_i(v)} d(u)$, where $N_i(v)$ is the set of neighbors of v in the sampled graph G_i, and continue selecting neighbors of the running solution until we reach $|V_Q|$ nodes.

Spectral: We select the best solution computed by the threshold method in terms of spectral difference in Equation (2). We vary the threshold θ to obtain further solutions, while enforcing the size of the solution to be $|V_Q|$.

6 Experiments

Here we empirically evaluate our method, SSL, on a number of datasets and against several hypotheses. Our evaluation aims to answer the following questions: **(Q1)** Do the regularization term and the constraint $\|\mathbf{v}\| = c$ in Equation (5) help the localization? **(Q2)** How does the conductance of the part corresponding to Q affect the quality of localization and how does SSL fare against state-of-the-art methods for graph alignment? **(Q3)** What kind of graphs are challenging for SSL and why?

6.1 Experiment Design

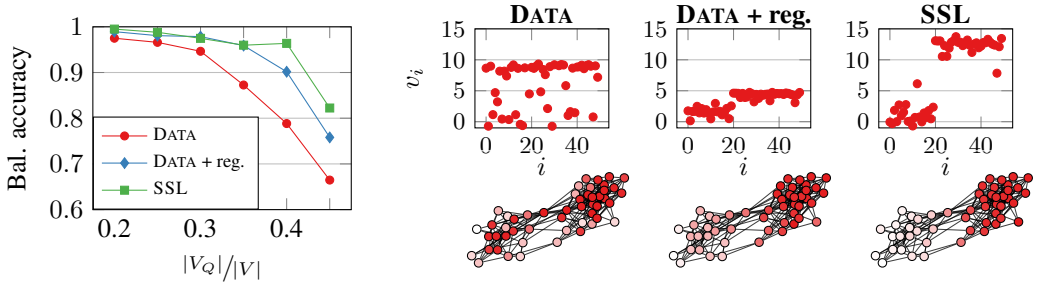
The code and data are available at https://github.com/AU-DIS/SSL.

Hyperparameters. Unless stated otherwise, we choose $\texttt{maxiter}_{\texttt{out}} = 3$, $\texttt{maxiter}_{\texttt{in}} = 500$, $a_{\texttt{tol}} = 10^{-5}$ and $\alpha = 0.02$. Regarding the regularization coefficient, we select $\mu = 0.2$ through grid search. This choice achieves good accuracy across datasets and conductance levels. In the sampling-and-aggregation methods of Section 5, we sample $k = 30$ graphs with $t = 0.8|E|$ edges; in the threshold method, we set $\theta = 0.2$. We initialize $\mathbf{v}_i = \frac{c}{\sqrt{n}}$ uniformly for all i as we experience stable results regardless the initialization strategy.

Datasets. We evaluate SSL on the three real-world graphs from [26] and two synthetic graphs generated by the Erdős-Renyi (ER) and Barabasi-Albert (BA) models. The data characteristics are described in the supplementary material. Additionally, we generate graphs with community structure using the stochastic block model (SBM) [27].

Choosing Q. Given a number k, we generate a query workload of size $V_Q = k$ from a real-world graph G to evaluate our subgraph localization method as follows. **(1)** Randomly select a node u, add it to V_Q and place all its neighbors into a set N. **(2)** Randomly select a node u' from N, add it to V_Q, place in N all its neighbors not in V_Q. **(3)** Repeat the previous step until $|V_Q| = k$. **(4)** Set Q as the subgraph induced by V_Q in G. For graphs generated by SBM, we set Q as the smallest community.

Quality measure. To evaluate performance in a manner independent of subgraph size, we employ the popular classification performance measure of **Balanced Accuracy (BA)** [28]; given the query graph V_Q and the subgraph V_S returned by a localization algorithm, balanced accuracy is the arithmetic mean of sensitivity (or recall) and specificity: $\text{BA}(\mathbf{v}) = \frac{1}{2}\left(\frac{|V_Q \cap V_S|}{|V_Q|} + \frac{|\neg V_Q \cap \neg V_S|}{|\neg V_Q|}\right)$. Nevertheless, BA measures the extent to which the returned subgraph exactly matches the node identities of the query subgraph, without counting parts that are structurally isomorphic, yet do not share the same identity. In reverse, the spectral difference in our optimization objective treats such isomorphic parts as fully effective solutions, without considering node identities. To evaluate our methods in terms of their native objective, we also report the **Spectral difference** $\|\boldsymbol{\lambda}(\mathbf{L} - \mathbf{E} + \text{diag}(\mathbf{v})) - \boldsymbol{\lambda}_Q\|_2^2$ between the query Q and the solution graph G_S.



(a) Balanced Accuracy

(b) $\mathbf{v}$ values in ascending order and corresponding graph nodes.

Figure 3: Variants of the objective function; SSL's objective achieves the highest accuracy.

6.2 Ablation Study

We commence our study by examining how the terms in SSL's objective function (Equation 5) affect the result. The objective function consists of: (1) the **data term**, that drives the alignment between the spectrum of the part and that of the query, (2) a **spectral regularization** term that exhorts $\mathbf{v}$ to be in the null space of $\mathbf{L} - \mathbf{E}$ and (3) the **sphere constraint** that enforces a constant norm on the potential $\mathbf{v}$. To study the contribution of each term on the results, we compare SSL against two

variants thereof, one that optimizes only the data term in Equation (2), and one that optimizes a linear combination of the data term and the spectral regularization, without a sphere constraint.

We experiment on graphs with $|V| = 200$ nodes sampled form the stochastic block model, letting the size of the query subgraph increase from 20% of the graph to 45%. Figure 3a reports on the results of this ablation study in terms of average balanced accuracy over 5 sampled graphs for each subgraph size. Unsurprisingly, the optimization of the data term yields the worst results, although the method performs well on small subgraphs. Still, the addition of the spectral regularizer and sphere constraint enhances the results up to 20% accuracy. For small subgraphs the sphere constraint brings only marginal gains compared to the spectral regularization. On the other hand, on large query subgraphs, the sphere constraint boosts the accuracy by an additional 8%.

To further corroborate these results, Figure 3b shows an example of how the terms impact the potential $\mathbf{v}$, on a 40-node graph sampled from the SBM with two communities with 20 nodes each; the query graph is one of the two communities. Ideally, we would like to obtain a $\mathbf{v}$ clearly separating values between the part corresponding to the query graph and the rest. In that case, we say that $\mathbf{v}$ forms a step function. The optimization of the *data term* (left chart) alone leads to no clear separation between the two parts. Introducing the *spectral regularization* (middle chart) yields a result closer to a step function, though some nodes are incorrectly assigned to the part. Finally, the full objective in Equation 5 produces to a clearly separated potential vector $\mathbf{v}$. Visualizing the mapping of this potential to the graph G, we clearly recognize the part G_S as the light-colored nodes.

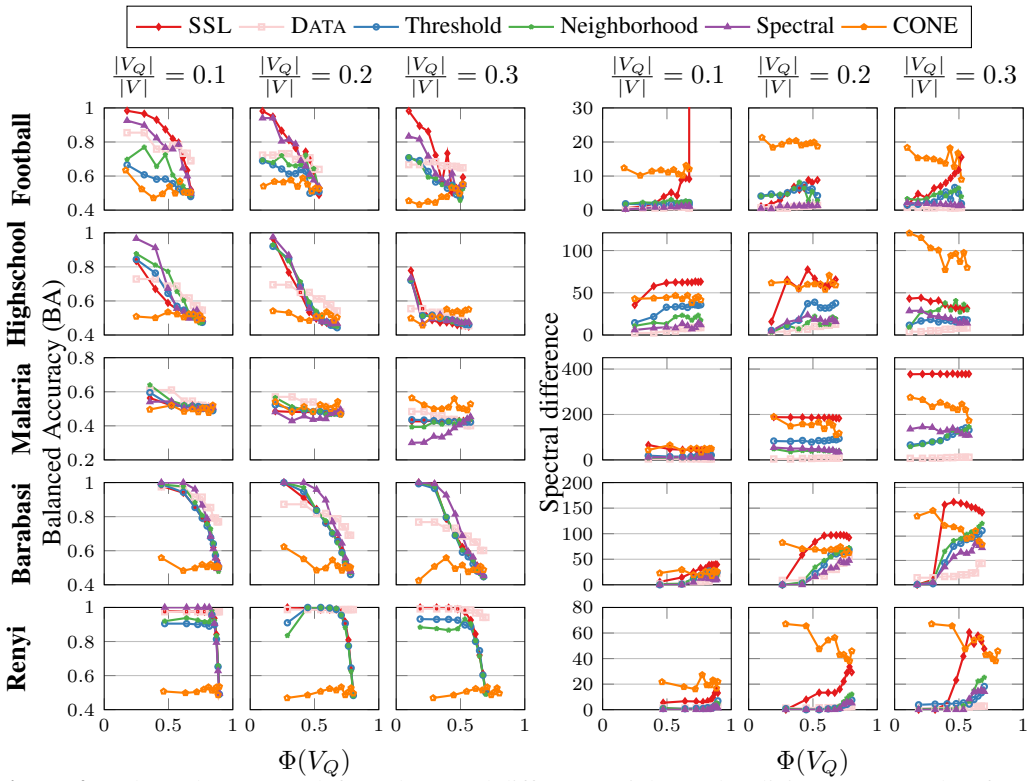


Figure 4: Balanced accuracy (left) and spectral difference (right) on localizing query graphs of 10%, 20%, 30% of the parent graph vs. conductance from query graph to rest of the parent graph.

6.3 Competing methods

Here we assess our method and its enhanced variants against previous work. To the best of our knowledge, no extant unsupervised method is capable of answering localization queries in graphs with more than 15 nodes [29]. Therefore, we compare SSL to the nearest feasible competitor, namely the state-of-the-art method for unsupervised graph alignment, CONE [30]. To set up CONE so that it detects subgraphs, we inject in the query nodes with degree 0, so that the size of the query Q corresponds to that of the graph G, i.e., $|V_Q = V|$. We extract the ensuing localization vector as the matches of query nodes in G with the default hyper-parameter settings.

Figure 4 presents the average BA of randomly generated connected subgraphs of our test data sets, 10 for each query size, as a function of *conductance*, $\Phi(V_Q) = \frac{\sum_{i \in V_Q, j \notin V_Q} A_{ij}}{\min\left(\sum_{i \in V_Q, j \in V} A_{ij}, \sum_{i \notin V_Q, j \in V} A_{ij}\right)}$, i.e., the ratio between the size of the cut among query Q and graph G and the minimum number of edges among the two resulting partitions; to vary conductance, we start with a disconnected query graph and progressively add connecting edges, and repeat this process 5 times for each of the 10 target conductance value, averaging results. A graph's *minimum conductance* is associated [31] with its *algebraic connectivity*, i.e., second smallest eigenvalue λ_2 [32]. A larger conductance denotes more edges between the query subgraph and the rest of the graph, thus a harder subgraph localization instance. We use query subgraphs corresponding to 10%, 20% and 30% of the full graph size.

The results in Figure 4 also show that SSL effectively localizes the query in real graphs. Accuracy gradually increases as conductance approaches $\Phi(V_Q) = 0$, and finally settles close to 1 when the query becomes loosely connected. On the other hand, optimizing the spectral difference alone leads to good spectral difference but modest accuracy results. The bagging-like variants perform well in most datasets. Remarkably, the Spectral strategy outperforms SSL in accuracy and exhibits comparable spectral distance to DATA, which explicitly minimizes it. The performance of SSL most often exceeds that of CONE. Even though performance drops as conductance grows, a real application would aim at detecting subgraphs that exhibit distinguishable structures, such as communities. Such subgraphs significantly deviate from random and complete subgraphs, most prominently in having *lower conductance*. As conductance grows, the query subgraph progressively immerses into other nodes, hence our spectral method cannot easily discern them. Yet, the fact that CONE fares poorly in spectral distance indicates that it neglects alternative solutions isomorphic to the query graph.

Further, the results in Figure 4 show that SSL consistently outperforms CONE on synthetic graphs. As with real graphs, we observe a gradual accuracy increase as the graph becomes progressively disconnected. Notably, on ER graphs, SSL succeeds even at high conductance values (> 0.6).

6.4 Comparison of Spectral Characteristics

Impact of the graph's spectrum. To better understand the performance of SSL on different graphs, we look at it under the lens of the graph's spectrum. Figure 5 shows the spectra of the real and synthetic graphs in our experiments, normalized in the range $\left[0, \frac{\lambda_n - \lambda_2}{\lambda_n}\right]$. First, we observe that the spectra of synthetic graphs exhibit a gradual increase and a small difference between λ_2 and the maximum eigenvalue λ_n. By the *Generalized Cheeger's inequality* [33] the k^{th}-order conductance, $\min_{V_1, V_2, \ldots, V_k} \max\{\Phi(V_i) : i = 1, 2, \ldots, k\}$, is related to the k^{th} eigenvalue. Thus, under gradual eigenvalue growth, the presence or absence of one edge does not affect the spectrum significantly, hence the projected gradient descent in SSL gracefully retrieves a good solution. On the other hand, the spectra of our real graphs exhibit an abrupt divergence between λ_2 and higher eigenvalues, indicating that a single edge may significantly affect the spectrum, rendering the task of projected gradient descent more challenging. In effect, SSL performs better as the gap between λ_2 and the rest of the eigenvalues decreases.

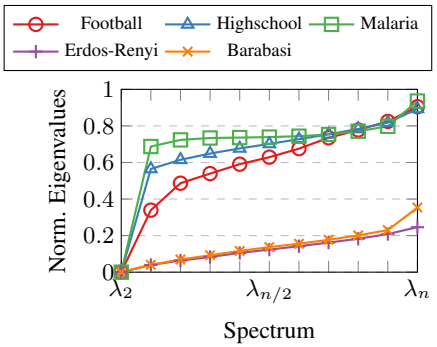


Figure 5: Datasets' graph spectra.

7 Conclusion

We studied the challenging problem of *subgraph localization*, which calls to find a set of nodes in a larger graph whose induced subgraph best matches a given query subgraph. We devised a spectral solution that adds a penalty to the larger graph's Laplacian matrix so as to obtain a spectrum matching that of the query graph. This novel approach requires solving a non-convex, non-smooth problem for which we devised a numerical method. We also proposed a suite of effective bagging-style strategies to boost performance. Our results demonstrate that our spectral method localizes query subgraphs more effectively than a baseline based on the state-of-the-art method for graph alignment. To our knowledge, this is the first endeavor in effective subgraph localization that handles graphs in the order of magnitude of hundreds of nodes.

References

[1] Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *ICDM*, pages 313–320. IEEE, 2001. 1

[2] Monica Bianchini, Giovanna Maria Dimitri, Marco Maggini, and Franco Scarselli. Deep neural networks for structured data. In *Computational Intelligence for Pattern Recognition*, pages 29–51. Springer, 2018. 1

[3] Foteini Katsarou, Nikos Ntarmos, and Peter Triantafillou. Performance and scalability of indexed subgraph query processing methods. *Proc. VLDB Endow.*, 8(12):1566–1577, 2015. 1, 2

[4] Shixuan Sun and Qiong Luo. Scaling up subgraph query processing with efficient subgraph matching. In *ICDE*, pages 220–231. IEEE, 2019. 1, 2

[5] Si Zhang and Hanghang Tong. Final: Fast attributed network alignment. In *KDD*, pages 1345–1354, 2016. 1, 2

[6] Judith Hermanns, Anton Tsitsulin, Marina Munkhoeva, Alexander M. Bronstein, Davide Mottin, and Panagiotis Karras. GRASP: graph alignment through spectral signatures. In *APWeb-WAIM*, pages 44–52, 2021.

[7] Judith Hermanns, Konstantinos Skitsas, Anton Tsitsulin, Marina Munkhoeva, Alexander Kyster, Simon Nielsen, Alexander M Bronstein, Davide Mottin, and Panagiotis Karras. Grasp: Scalable graph alignment by spectral corresponding functions. *TKDD*, 17(4):1–26, 2023. 1

[8] Rafael Najmanovich, Natalja Kurbatova, and Janet Thornton. Detection of 3d atomic similarities and their use in the discrimination of small molecule protein-binding sites. *Bioinformatics*, 24 (16):i105–i111, 2008. 1

[9] Arianna Rampini, Irene Tallini, Maks Ovsjanikov, Alex M Bronstein, and Emanuele Rodolà. Correspondence-free region localization for partial shape similarity via hamiltonian spectrum alignment. In *3DV*, pages 37–46. IEEE, 2019. 1, 2, 3, 5

[10] Marc Fyrbiak, Sebastian Wallat, Sascha Reinhard, Nicolai Bissantz, and Christof Paar. Graph similarity and its applications to hardware security. *IEEE Transactions on Computers*, 69(4): 505–519, 2019. 1

[11] Moody Chu and Gene Golub. *Inverse eigenvalue problems: theory, algorithms, and applications*. OUP Oxford, 2005. 1, 3

[12] Chi Thang Duong, Trung Dung Hoang, Hongzhi Yin, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer. Efficient streaming subgraph isomorphism with graph neural networks. *Proceedings of the VLDB Endowment*, 14(5):730–742, 2021. 2

[13] Shixuan Sun, Xibo Sun, Yulin Che, Qiong Luo, and Bingsheng He. Rapidmatch: a holistic approach to subgraph query processing. *Proc. VLDB Endow.*, 14(2):176–188, 2020. 2

[14] Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, Jure Leskovec, et al. Neural subgraph matching. *arXiv preprint arXiv:2007.03092*, 2020. 2

[15] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *ICML*, pages 3835–3845. PMLR, 2019. 2

[16] Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Ivan Oseledets, and Emmanuel Müller. FREDE: anytime graph embeddings. *PVLDB*, 14(6):1102–1110, 2021. 2

[17] Konstantinos Skitsas, Karol Orlowski, Judith Hermanns, Davide Mottin, and Panagiotis Karras. Comprehensive evaluation of algorithms for unrestricted graph alignment. In *EDBT*, pages 260–272, 2023. 2

[18] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *IEEE/CVF CVPR*, pages 10156–10165, 2020. 2

[19] Utkan Onur Candogan and Venkat Chandrasekaran. Finding planted subgraphs with few eigenvalues using the schur–horn relaxation. *SIAM Journal on Optimization*, 28(1):735–759, 2018. 2

[20] Yoni Choukroun, Alon Shtern, Alex Bronstein, and Ron Kimmel. Hamiltonian operator for spectral shape analysis. *IEEE TVCG*, 26(2):1320–1331, 2018. 2, 3

[21] David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. Approximating the spectrum of a graph. In *KDD*, pages 1263–1271, 2018. 4, 6

[22] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 4

[23] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Backpropagation-friendly eigendecomposition. *NeurIPS*, 32, 2019. 5

[24] Allan Grønlund, Kasper Green Larsen, Alexander Mathiasen, Jesper Sindahl Nielsen, Stefan Schneider, and Mingzhou Song. Fast exact k-means, k-medians and Bregman divergence clustering in 1d. *arXiv preprint arXiv:1701.07204*, 2017. 6

[25] Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 1998. 6

[26] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL `https://networkrepository.com`. 7

[27] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983. 7

[28] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *International Conference on Pattern Recognition*, pages 3121–3124, 2010. 7

[29] Indradyumna Roy, Venkata Sai Velugoti, Soumen Chakrabarti, and Abir De. Interpretable neural subgraph matching for graph retrieval. In *AAAI*, 2022. 8

[30] Xiyuan Chen, Mark Heimann, Fatemeh Vahedian, and Danai Koutra. Cone-align: Consistent network alignment with proximity-preserving node embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1985–1988, 2020. 8

[31] Jeff Cheeger. *A Lower Bound for the Smallest Eigenvalue of the Laplacian*, pages 195–200. Princeton University Press, Princeton, 2015. 9

[32] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2): 298–305, 1973. 9

[33] Victor E Lee, Ning Ruan, Ruoming Jin, and Charu Aggarwal. A survey of algorithms for dense subgraph discovery. In *Managing and Mining Graph Data*, pages 303–336. Springer, 2010. 9

[34] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002. 13

[35] Daniel B Larremore, Aaron Clauset, and Caroline O Buckee. A network approach to analyzing highly recombinant malaria parasite genes. *PLoS computational biology*, 9(10):e1003268, 2013. 13

[36] Julie Fournet and Alain Barrat. Contact patterns among high school students. *PloS one*, 2014. 13

[37] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. 13

[38] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960. 13

[39] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003. 13

[40] Jérôme Kunegis. Konect: the Koblenz network collection. In *WWW*, pages 1343–1350. ACM, 2013. 14