

GRASP: Graph Alignment through Spectral Signatures

Judith Hermanns¹, Anton Tsitsulin², Marina Munkhoeva³, Alex Bronstein⁴,
Davide Mottin¹, and Panagiotis Karras¹

¹ Aarhus University

{davide, judith, panos}@cs.au.dk

² University of Bonn

tsitsulin@bit.uni-bonn.de

³ Skoltech

marina.munkhoeva@skolkovotech.ru

⁴ Technion

bron@cs.technion.ac.il

Abstract. What is the best way to match the nodes of two graphs? This *graph alignment* problem generalizes graph isomorphism and arises in applications from social network analysis to bioinformatics. Some solutions assume that auxiliary information on known matches or node or edge attributes is available, or utilize arbitrary graph features. Such methods fare poorly in the pure form of the problem, in which only graph structures are given. Other proposals translate the problem to one of aligning node embeddings, yet, by doing so, provide only a single-scale view of the graph.

In this paper, we transfer the shape-analysis concept of functional maps from the continuous to the discrete case, and treat the graph alignment problem as a special case of the problem of finding a mapping between functions on graphs. We present GRASP, a method that first establishes a correspondence between functions derived from Laplacian matrix eigenvectors, which capture multiscale structural characteristics, and then exploits this correspondence to align nodes. Our experimental study, featuring noise levels higher than anything used in previous studies, shows that GRASP outperforms state-of-the-art methods for graph alignment across noise levels and graph types.

1 Introduction

Graphs model relationships between entities in several domains, e.g., social networks, protein interaction networks, email communication or chemical molecules. The structure of such graphs captures rich information on how people are connected, how molecules function, or how proteins interact.

At the same time, the expressive nature of graphs also implies complexity, which renders some fundamental problems hard. For instance, the *graph isomorphism* problem, which is to determine whether two graphs share the same structure is neither known to be polynomially solvable nor **NP**-complete, and

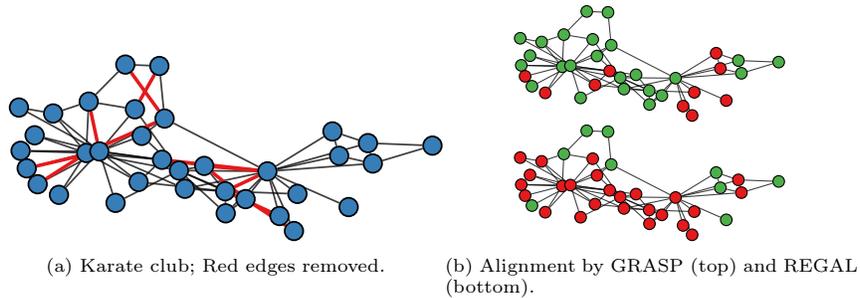


Fig. 1: With a few removed edges, REGAL [12], an alignment method based on *local* features, fails to correctly align the distorted Karate club graph to the original; GRASP identifies most of nodes (correctly aligned nodes in green).

has been used to define the **GI** complexity class [17]. Problems that generalize graph isomorphism occur frequently in the field of graph analytics. One of those is the **NP**-complete *subgraph isomorphism* problem; another is *graph alignment*, which aims to find the best (exact or inexact) matching among the nodes of a pair of graphs; a solution to this problem is sine qua non in tasks such as identifying users in different social networks [14], matching objects in images by establishing feature correspondences and comprehending protein response in the body [15].

In case additional background information is available, such as node and edge attributes in the two graphs to be aligned, or existing valid *seed* matches, then the problem is solvable via supervised methods [24,6]. However, in case only graph structures are given, then the problem of aligning two graphs by matching structures, is at least as hard as graph isomorphism even in its approximate version [1].

Existing approaches to the graph alignment problem are oriented toward using a few *heuristic graph features*, such as landmarks, in order to detect a good alignment [12], *exploiting additional information* such as node attributes [35] or bipartite networks [20], or optimizing objectives based only on *local connections* among nodes [10,27,22]. On the other hand, the spectra of *Laplacian matrices* have been successfully employed to devise a similarity measure among graphs [31]. Laplacian spectra capture important *multiscale* properties, such as local-scale ego-nets and global-scale communities. Previous approaches rooted in spectral characteristics decompose large matrices expressing all alignments among edges in two graphs [10,27,22] and formulate the solution as finding the leading eigenvector of such matrices. These approaches disregard most eigenvectors and consider only local edge variations. To our knowledge, the spectral properties of *Laplacian matrices* have *not yet* been utilized to any significant extent for an end-to-end graph alignment method.

We propose GRASP, short for **GR**aph Alignment through **SP**ectral Signatures, a principled approach towards detecting a good alignment among graphs, grounded on their spectral characteristics, i.e., eigenvalues and eigenvectors of their Laplacian matrices [7]. We transfer the methodology of matching among

shapes based on *corresponding functions* [28] to the domain of graphs: we first extract a mapping of node-evaluated functions grounded on the graph’s heat kernel, and then apply this mapping to the matching on nodes. Figure 1 shows an example alignment of the Karate club with a deteriorated version obtained by removing some edges; GRASP correctly aligns most of the nodes, while RE-GAL [12] based on local descriptors fails to do so.

In short, we propose GRASP, a graph alignment method based on spectral graph characteristics and show its effectiveness in recovering real-graph alignments, with higher accuracy and similar runtime as the state of the art.

2 Related Work

We discuss related work in two main categories: **restricted alignment**, which requires ground-truth mapping or other additional information, and **unrestricted alignment**, which requires neither supervision nor additional information. Table 1 gathers together previous works’ characteristics.

2.1 Restricted Alignment

Restricted methods incorporate non-structural information; a restricted method can be *supervised* or *assisted*.

Supervised methods exploit pre-aligned pairs of seed nodes to construct a first alignment. *Percolation graph matching* (PGM) [14,33] propagates ground-truth alignments across the network. *Representation learning* approaches, such as IONE [24], PALE [26], and DeepLink [36], learn a low-dimensional embedding of the graph nodes and map the node embeddings of one graph to another. A similar method aligns multiple networks at once [6]. *Active network alignment* [25] applies active learning to elicit expert guidance on alignments. Overall, supervised methods rely on prior knowledge, which may not be available.

Assisted methods utilize auxiliary information or structural constraints. BigAlign [20] focuses on bipartite graphs; however, most graphs are not bipartite. FINAL [35] aligns nodes based on similarity of topology and attributes. GSANA [34] lets pairwise distances to seed nodes guide the matching. Another variant matches *weighed* matrices using their spectra [32]; that is inapplicable to the unweighted case. Overall, restricted methods cannot handle cases where the only given information is graph structure.

2.2 Unrestricted Alignment

Unrestricted methods require neither prior knowledge of ground-truth pairs nor other information on the input graph.

Integer-programming methods. Klau [15] presents a Lagrangian relaxation for the integer programming problem posed by network alignment; the resulting algorithm is polynomial, yet still impracticable for large networks.

Method	Unsuperv.	Spectral	Plain	Flexible	Offline Precomp.	Multiscale
Supervised	✗	✗	✓	✗	✗	✗
BigAlign	✓	✗	✓	✗	✗	✗
FINAL	✓	✗	✗	✗	✗	✗
IsoRank	✓	✗	✓	✓	✓	✗
LaplMatch	✓	✓	✓	✗	✓	✗
LREA	✓	✓	✓	✗	✗	✗
REGAL	✓	✗	✓	✓	✓	✗
GRASP	✓	✓	✓	✓	✓	✓

Table 1: Related work in terms of present (✓) and absent (✗) properties. *Supervised* methods [24,6,26,36] require aligned nodes as input. *Spectral* methods [27] use spectral properties of alignment matrices. FINAL [35] does not work on *plain* graph structures as it requires node attributes. REGAL [12] and IsoRank [22,30] are *flexible* in allowing different algorithms for alignment (e.g. bipartite matching, nearest neighbors). Among unrestricted methods (rows 4–7), LREA [27] cannot benefit from offline precomputation (results in Figure 6). GRASP explicitly captures *multiscale* properties through the heat kernel.

Embedding-based methods. REGAL [12] constructs node embeddings based on the connectivity structure and node attributes, and uses the similarity between these features for node alignment; we classify REGAL as an unrestricted method since it can work without attributes. CONE-Align [5] realigns node representations, without prejudice to the representation used.

Matrix decomposition methods. IsoRank [30] aligns multiple protein-protein interaction networks aiming to maximize the overall quality across all input networks; it constructs an eigenvalue problem for every pair of input networks and extracts a global alignment across a set of networks by a k -partite matching; it uses structural properties (PageRank), but also relies on a similarity measure between nodes which in a biology-specific case builds on the similarity of the proteins; it is improved with greedy approaches [18]. Another improvement on IsoRank, IsoRankN [22], performs spectral clustering on the induced graph of pairwise alignment scores; as it is based on spectral methods, IsoRankN is claimed to be both error-tolerant and computationally efficient. EigenAlign [10] formulates the problem as a Quadratic Assignment Problem that considers both matches and mismatches and solves it by *spectral decomposition* of matrices. Building thereupon, Low-Rank EigenAlign [27] solves a maximum weight bipartite matching problem on a *low-rank* version of a node-similarity matrix, hence requires memory linear in the size of the graphs. However, EigenAlign variants use the first eigenvector of a joint adjacency matrix between the two graphs to be aligned, rather than the eigenvectors of graph Laplacians, which provides richer information.

Belief propagation methods. NetAlign [3] solves a *sparse* graph alignment variant by message passing.

2.3 Shape Matching

Our work is inspired by shape matching methods that employ spectral properties [23,28,21]. Functional maps [28] generalize the matching of points to the matching of *corresponding functions* among shapes, by revealing a common decomposition of such functions using the eigenvectors of the Laplace-Beltrami operator; the graph equivalent of that operator is a graph’s Laplacian matrix. Extensions of this methods match non-isometric shapes by aligning their Laplace-Beltrami operators’ eigenbases [21], and match a part of a shape to another full shape in the spectral domain [23] without requiring spatially modeling the part of a shape.

2.4 Spectral Methods

Graph spectra [7] facilitate problem-solving in graph analysis, image partitioning, graph search, and machine learning [29]. NetLSD [31] uses Laplacian spectral signatures to detect graph similarity, *but not to align graphs*, in a multi-scale fashion. LaplMatch [16] derives a permutation matrix for shape matching from Laplacian eigenvectors, without considering multiscale properties. While calculating a graph’s spectrum is computationally challenging, recent work proposes an approximation via spectral moments estimated through random walks [8].

3 Background and Problem

Graph Alignment. Consider two undirected graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where V_* are node sets, $E_* \subseteq V_* \times V_*$ are edges, and⁵ $|V_1| = |V_2| = n$. A graph’s *adjacency matrix* $A \in \{0, 1\}^{n \times n}$ is a binary matrix where $A_{ij} = 1$ if there is an edge between nodes i and j and $A_{ij} = 0$ otherwise.

Definition 1. *Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a graph alignment $R : V_1 \rightarrow V_2$ is an injective function that maps nodes of G_1 to nodes of G_2 .*

The graph alignment problem is to find such a function, which, expressed as a permutation matrix P , minimizes the difference $\|PA_1P^\top - A_2\|^2$. In case of isomorphic graphs, there exists a P such that $PA_1P^\top = A_2$, i.e., aligns the two graphs exactly. We are interested in the general, unrestricted problem case, in which there are no additional constraints on node attributes or matches known in advance. The problem is hard and not known to be in **NP**.

We may express graph alignment in terms of a ground truth function $\tau : V_1 \rightarrow V_2$ that returns the correct alignment between the nodes V_1 in G_1 and the nodes V_2 in G_2 . In the case of isomorphic graphs, this ground truth function τ is a bijection that admits an inverse mapping $\tau^{-1} : V_2 \rightarrow V_1$. The composition of the indicator function $\delta_i : V_1 \rightarrow \{0, 1\}$ with τ^{-1} , $\delta_i \circ \tau^{-1} : V_2 \rightarrow \{0, 1\}$ expresses the

⁵ Solutions to the problem of aligning graphs with unequal numbers of nodes can rest on solutions to this basic problem form.

complete isomorphism among the two graphs, returning 1 if node $u \in V_2$ maps to node $i \in V_1$, 0 otherwise. By generalization, the composition $g_i = f_i \circ \tau^{-1}$ maps functions in G_2 to functions in G_1 for any family of real-valued functions $f_1, \dots, f_q, f_i : V_1 \rightarrow \mathbb{R}$ and $g_1, \dots, g_q, g_i : V_2 \rightarrow \mathbb{R}$ that associate a real value to each node in G_1 and G_2 . This transformation among functions is called a *functional representation* of the mapping τ . In effect, finding an alignment among the nodes of two graphs corresponds to finding an alignment among functions on those nodes. We use such *functional alignments* as a shortcut to *node alignments*. To get there, we extend the concept of a functional map [28] from the continuous to the discrete case.

Functional maps. The operator $T_{\mathcal{F}} : (V_1 \times \mathbb{R}) \rightarrow (V_2 \times \mathbb{R})$ maps functions f on the nodes in G_1 to functions g on the nodes in G_2 , i.e. $T_{\mathcal{F}}(f) = f \circ \tau^{-1} = g$. This operator is linear in the function space, i.e., $T_{\mathcal{F}}(c_1 f_1 + c_2 f_2) = (c_1 f_1 + c_2 f_2) \circ \tau^{-1} = c_1 f_1 \circ \tau^{-1} + c_2 f_2 \circ \tau^{-1} = c_1 T_{\mathcal{F}}(f_1) + c_2 T_{\mathcal{F}}(f_2)$. In addition, let ϕ_1, \dots, ϕ_n and ψ_1, \dots, ψ_n denote orthogonal bases for the space of functions on G_1 's nodes, $V_1 \times \mathbb{R}$, and that on G_2 's nodes, $V_2 \times \mathbb{R}$, respectively. Since those functions produce n -dimensional vectors, we can represent them as linear combinations of their basis vectors, $f = \sum_{i=1}^n a_i \phi_i$ and $g = \sum_{j=1}^n b_j \psi_j$. Then, by the linearity of $T_{\mathcal{F}}$,

$$T_{\mathcal{F}}(f) = T_{\mathcal{F}} \left(\sum_{i=1}^n a_i \phi_i \right) = \sum_{i=1}^n a_i T_{\mathcal{F}}(\phi_i) = \sum_{i=1}^n a_i \sum_{j=1}^n c_{ij} \psi_j = \sum_{j=1}^n b_j \psi_j$$

where $T_{\mathcal{F}}(\phi_i) = \sum_{j=1}^n c_{ij} \psi_j$. It follows that each coefficient b_j is the dot-product $\sum_{i=1}^n a_i c_{ij}$ between the coefficients (a_1, \dots, a_n) of functions in G_1 and the coefficients (c_{1j}, \dots, c_{nj}) of the operator $T_{\mathcal{F}}$. In conclusion, in order to align real-valued functions on the nodes of two graphs, we need to find a *mapping matrix* $C \in \mathbb{R}^{n \times n}$ of coefficients among those functions; such a mapping matrix C maps functions from G_1 to G_2 , even when the ground-truth mapping τ is unknown. In a nutshell, GRASP obtains such a mapping matrix C for a well-chosen function and applies that C to mapping the indicator function δ from G_1 to G_2 , thereby constructing a node alignment. The main question we need to answer is what orthogonal basis and functions we should use to construct our mapping matrix C . The next section answers this question and builds on that answer to devise a solution.

4 Solution

Here, we choose an orthonormal basis and a function, which are, in our judgement, appropriate for node alignment purposes, and define the complete pipeline of our solution.

4.1 Choice of basis: Normalized Laplacian

As a basis for representing functions as linear combinations of base functions, we use the eigenvectors of the graph's *normalized Laplacian*, i.e., the matrix $\mathcal{L} = I -$

$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where D a diagonal degree matrix of node degrees $D_{ii} = \sum_{j=1}^n A_{ij}$ and A is the graph adjacency matrix; its eigendecomposition is $\mathcal{L} = \Phi\Lambda\Phi^\top$, where Λ is a diagonal matrix of eigenvalues, $\{\lambda_1, \dots, \lambda_n\}$, i.e., the graph's *spectrum*, which encodes information about communities, degree distribution, and diameter, and Φ is a matrix of eigenvectors, $\Phi_{\mathcal{L}} = [\phi_1\phi_2 \dots \phi_n]$. The eigenvectors form an orthogonal basis, which we use a standard basis. We use ϕ (ψ) to indicate the eigenvectors of the Laplacian of graph G_1 (G_2).

We consider this basis to be suitable, since the eigenvectors of the normalized Laplacian converge to the eigenfunctions of the Laplace-Berlrami operator [4], which measures the smoothness of continuous surfaces.

4.2 Choice of function: Heat Kernel

The choice of functions $f_i : V_1 \rightarrow \mathbb{R}$, $g_i : V_2 \rightarrow \mathbb{R}$, is critical for our method. A poor choice would be detrimental. A good choice should have the following properties:

Expressiveness. The function should express the graph's structure. For instance, a constant function returning the same value for all nodes would not yield a meaningful alignment.

Permutation-invariance. The function should not depend on the node index i ; the indicator function lacks this property.

Multiscale robustness. The function should robustly capture both local and global structures (e.g., edges and communities), insensitively to small perturbations.

A function fulfilling these requirements is the time-parameterized *heat kernel* [31]:

$$H_t = \Phi e^{-t\Lambda} \Phi^\top = \sum_{j=1}^n e^{-t\lambda_j} \phi_j \phi_j^\top \quad (1)$$

where $H_{t[ij]}$ measures the flow of heat from node i to node j at time t , as it diffuses from each node's neighborhood to the whole graph. We build our model functions over a sequence of time steps t using the *diagonal* of the heat kernel, which measures the heat flowing back to each node at time t .

The heat kernel expresses multiscale graph structure in a permutation-invariant manner and is robust to small changes. In the beginning of the diffusion, Equation (1) emphasises large λ , which correspond to *local* edge and ego-net properties. As time progresses, smaller eigenvalues get emphasized, reflecting *global* graph properties, such as communities.

We build our *corresponding functions* f_i, g_i , from the heat kernel at different time steps t , as linear combinations of the graph's Laplacian orthogonal eigenvectors. Let $F \in \mathbb{R}^{n \times q}$, $F = [f_1, \dots, f_q]$ be the matrix containing the diagonals of the heat kernel of G_1 , $H_t^{G_1}$, over q time⁶ steps, $f_i = \sum_{j=1}^n e^{-t_i \lambda_j} \phi_j \odot \phi_j$, where \odot denotes the element-wise vector product. Likewise, the matrix $G \in \mathbb{R}^{n \times q}$, $G =$

⁶ In our experiments we select $q = 100$ values evenly spaced on the linear scale in the range $[0.1, 50]$.

$[g_1, \dots, g_q]$ contains the diagonals of $H_t^{G_2}$, the heat kernel of G_2 . While the q columns of F and G contain the same time-dependent heat-kernel-diagonal functions on the nodes of two graphs, their n rows (i.e., nodes) are not aligned. We need to obtain such a node alignment.

4.3 Mapping matrix

We approximate each function f_i using only the first k eigenvectors, as done, by analogy, on shapes analysis [4], and thereby calculate the corresponding function matrices F and G . F and G can be thought as coefficient matrices used to produce linear combinations, $F^\top \Phi$ and $G^\top \Psi$, of the Laplacian eigenvectors of G_1 and G_2 , respectively. With a slight abuse of notation, we denote with Φ and Ψ the first k eigenvectors, hence $F^\top \Phi$ and $G^\top \Psi$ are in $\mathbb{R}^{q \times k}$. In the projection of the functions on the first k eigenvectors, we would like the corresponding functions to be equal up to a coefficient matrix $C \in \mathbb{R}^{k \times k}$. In the case of isomorphic graphs, it holds that $F^\top \Phi = G^\top \Psi C$

$$\begin{bmatrix} \text{diag}(g_1^\top \Psi) \\ \vdots \\ \text{diag}(g_q^\top \Psi) \end{bmatrix} \begin{bmatrix} c_{11} \\ \vdots \\ c_{kk} \end{bmatrix} = \begin{bmatrix} \Phi^\top f_1 \\ \vdots \\ \Phi^\top f_q \end{bmatrix} \quad (2)$$

Matrix C is diagonal in the case of isomorphic graphs and deviates from a diagonal form as graphs diverge from isomorphism; for simplicity, we assume a diagonal C , and obtain the diagonal entries that minimize the L_2 -norm difference $\|\cdot\|_2^2$ between the left and right side of Eq. (2) using the ordinary least squares method, as in [21]. In Section 4.5 we delve into the case of non-isomorphic graphs.

4.4 Node-to-node correspondence

We consider the delta function $\delta_i(\cdot)$ as corresponding function; these functions yield an $n \times n$ identity matrix. We express such a function as a vector of coefficients, since the vector of δ_i is the i th row of the heat kernel at $t = 0$:

$$\delta_i = H_{i,t=0}^{G_1} = \sum_{j=1}^n \phi_{ij} \phi_j$$

The computation for delta functions on G_2 follows equivalently using Ψ in place of Φ . We may match the coefficient vectors of these corresponding indicator functions, as, ideally, for two matching nodes $v_i \in V_1$ and $v'_j \in V_2$, the coefficients of δ_i and δ_j for Φ and Ψ should be identical. In particular, the coefficients expressing δ_i as a linear combination of the first k eigenvectors are $\phi_{i1}, \dots, \phi_{ik}$. We set Φ^\top and $C\Psi^\top$ in $\mathbb{R}^{k \times n}$ as coefficient matrices of the delta functions, aligned by C . Rows correspond to the first k Laplacian eigenvectors, while columns stand for graph nodes, rather than for time steps of heat diffusion. We need to match coefficient vectors, i.e., columns of Φ^\top and $C\Psi^\top$, to each other. This problem amounts to a *linear assignment problem*; we apply an

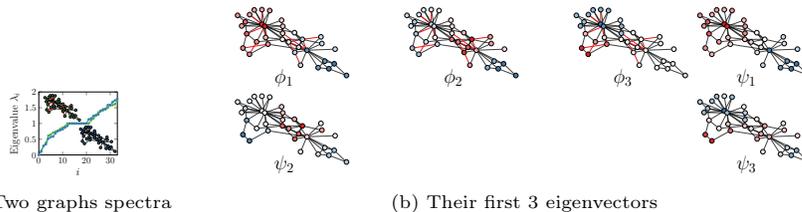


Fig. 2: We remove **red** edges from the **green** graph to obtain the **blue** graph. Eigenvalues interlace (a); eigenvectors ϕ_1, ϕ_2, ϕ_3 for **green** and ψ_1, ψ_2, ψ_3 for **blue** highlight common structures.

off-the-shelf algorithm therefore, such as **nearest neighbor search** or **Jonker-Volgenant (JV)** [13], to obtain a one-to-one matching between the columns of Φ^\top and $C\Psi^\top$, and hence an alignment of nodes. GRASP is flexible in that we may choose any matching method.

4.5 Base Alignment

We have hitherto assumed that the graphs to be aligned are isomorphic, hence their eigenvectors correspond to each other with possible sign changes and an orthogonal diagonal mapping matrix C exists. Still, if the graphs are not isomorphic, then their eigenvectors diverge and the diagonal matrix C , which we enforce, cannot capture their relationship well. Figure 2 highlights this issue: at a high level the eigenvectors underline common structures, but they differ at the node level. In this case, we need to *align* the two eigenvector bases before we consider aligning corresponding vectors and, eventually, nodes. We express this *base alignment* [21] in terms of an alignment matrix M .

Diagonalization. We align the eigenvectors Ψ by a rotation matrix M so as to transform Ψ into Φ : $\hat{\Psi} = \Psi M$. Since $\mathcal{L}\Psi = \Psi\Lambda$, finding Ψ is equivalent to the solution of the quadratic minimization problem $\min_{\Psi} \text{off}(\Psi^\top \mathcal{L}_2 \Psi)$ s.t. $\Psi^\top \Psi = I$ which penalizes elements $\text{off}(\cdot)$ outside of the diagonal, in order to preserve orthogonality of the basis.

Since the eigenvectors are orthonormal, $\Psi^\top \Psi = I$ and for G_2 's graph Laplacian eigenvectors Λ_2 , $\Psi^\top \mathcal{L}_2 \Psi = \Psi^\top \Psi \Lambda_2 = \Lambda_2$, and $M^\top \Psi^\top \mathcal{L}_2 \Psi M = M^\top \Lambda_2 M$. Putting the above together, our diagonalizing term is:

$$\min_M \text{off}(M^\top \Lambda_2 M) \text{ s.t. } M^\top M = I$$

As we are minimizing over orthogonal matrices we can equivalently express the objective above as a minimization over orthogonal matrices of size $n \times n$, $S(n, n)$:

$$\min_{M \in S(n, n)} \text{off}(M^\top \Lambda_2 M)$$

Coupling. In addition, the correspondence $\tau : G_1 \rightarrow G_2$ so that $\phi_i \approx \tau \circ \psi$ translates to

$$\min_{\Phi} \|F^\top \Phi - G^\top \Psi M\|_F^2$$

where F and G contain each graphs's corresponding functions. We combine the minimization terms for diagonalization and coupling, to get the following minimization problem, with regularization factor μ ⁷:

$$\min_{M \in S(n,n)} \text{off}(M^\top \Lambda_2 M) + \mu \|F^\top \Phi - G^\top \Psi M\|_F^2 \quad (3)$$

Given that the eigenvectors of isomorphic graphs match with sign changes, we initialize M as a diagonal matrix with $M_{ii} = 1$ if $\|F^\top \phi_i - G^\top \psi_i\| \leq \|F^\top \phi_i + G^\top \psi_i\|$, $M_{ii} = -1$ otherwise. Eq. (3) leads to a manifold optimization problem, which we solve by trust-region methods [2].

Scalability. We avoid computing all eigenvectors $n \times n$, exploiting the fact that we only need the first k eigenvectors for calculating C (see Section 4.3). So we only align the first k eigenvectors of Ψ to the first k eigenvectors of Φ , i.e $\bar{\Phi} = \hat{\Psi} = \bar{\Psi} M$ with $\bar{\Phi} = [\phi_1, \dots, \phi_k]$ and $\bar{\Psi} = [\psi_1, \dots, \psi_k]$. Let $\bar{\Lambda}_2 = \text{diag}(\lambda_1, \dots, \lambda_k)$, the problem in Eq. (3) becomes

$$\min_{M \in S(k,k)} \text{off}(M^\top \bar{\Lambda}_2 M) + \mu \|F^\top \bar{\Phi} - G^\top \bar{\Psi} M\|_F^2 \quad (4)$$

After obtaining M , we use the eigenvectors in $\bar{\Phi}$ and the aligned eigenvectors $\hat{\Psi} = \bar{\Psi} M$ in the next step for the final alignment of nodes. Our approach effectively trades off graph alignment with a proxy problem of manifold optimization, which we solve with reasonable accuracy and scalability.

4.6 Our algorithm: GRASP

Putting it all together, GRASP consists of five steps; the pseudocode is given in the supplementary material.

Steps 1: Compute eigenvectors. In the first step, calculate the Laplacians $\mathcal{L}_1, \mathcal{L}_2$ of the two graphs G_1 and G_2 . Then compute the eigenvectors Φ, Ψ and eigenvalues Λ_1, Λ_2 by the eigendecomposition $\mathcal{L}_1 = \Phi \Lambda_1 \Phi^\top$ and $\mathcal{L}_2 = \Psi \Lambda_2 \Psi^\top$.

Step 2: Compute corresponding functions. In the second step, calculate the matrices of corresponding functions $F = [f_1, \dots, f_q]$ and $G = [g_1, \dots, g_q]$ as diagonals of the heat kernel at time steps $[t_1, \dots, t_q]$ with $f_i = \sum_{j=1}^n e^{-t_i \lambda_j} \phi_j \odot \phi_j$ and g_i equivalently using Ψ .

Step 3: Base alignment. After the corresponding functions are calculated, obtain the base alignment matrix M by minimizing Eq. 3. Then align the first k columns of Ψ , denoted by $\bar{\Psi}$ to the corresponding first k columns $\bar{\Phi}$ of Φ as $\hat{\Psi} = \bar{\Psi} M$.

⁷ $\mu = 0.132$ in our experiments

Step 4: Calculate mapping matrix. Under the assumption that C is a diagonal matrix, calculate its diagonal elements c_{11}, \dots, c_{kk} by solving the least squares problem:

$$\min_{[c_{11}, \dots, c_{kk}]^\top} \left\| \begin{bmatrix} \text{diag}(g_1^\top \hat{\Psi}) \\ \vdots \\ \text{diag}(g_q^\top \hat{\Psi}) \end{bmatrix} \begin{bmatrix} c_{11} \\ \vdots \\ c_{kk} \end{bmatrix} - \begin{bmatrix} \bar{\Phi}^\top f_1 \\ \vdots \\ \bar{\Phi}^\top f_q \end{bmatrix} \right\|_2^2 \quad (5)$$

We then set $C = \text{diag}(c_{11}, \dots, c_{kk})$.

Step 5: Node alignment. To get the final node alignment, we apply a linear assignment algorithm on the rows of $\bar{\Phi}$ and $C^\top \hat{\Psi}$, which hold the indicator function coefficients.

Complexity analysis The computation of the first k Laplacian eigenvectors takes $\mathcal{O}(k \max\{|E_1|, |E_2|\})$ by fast methods for diagonally dominant matrices [19]. Base alignment needs $\mathcal{O}(k^3)$ to solve the orthogonality constraint through trust-region methods. The least-squares method runs in $\mathcal{O}(qk)$. The matching step by JV runs in $\mathcal{O}(n^3)$. Overall, the $\mathcal{O}(n^3)$ time factor is dominant.

Connection to Differential Geometry Our work rests on the theory on Riemannian manifolds [11] and builds on the analogy between a graph’s Laplacian and the continuous Laplace-Beltrami operator [31].

5 Experiments

We experiment on three different real-world networks; Table 1 in the Supplementary Material lists their properties. As in [12], we randomly permute the node order and inject noise by randomly deleting edges with probability p . We generate a graph for $p = 0.01$, which we align to 5 noisy graphs, one for each p value, ranging from 0.05 to 0.25; we measure alignment accuracy as the average ratio of correctly aligned nodes; note that none of the noisy graphs in a pair is a subset of the other.

Baselines. We compare against the following established state-of-the-art baselines for *unrestricted* graph alignment.

- **REGAL** [12]: A method based on embeddings utilizing local structural features. REGAL allows one-to-many matchings. For the sake of fairness, we let REGAL provide one-to-one matchings using the JV linear assignment algorithm, as GRASP does; we confirmed that, doing so, it fares better than using nearest neighbors.
- **Low Rank EigenAlign (LREA)** [27]: A spectral method that yields one-to-one matchings via the minimization of edge mismatches.

We eschew a comparison with IsoRank [30,22] and other methods for the alignment of biological networks [9], since REGAL and LREA significantly outperform those methods.

Parameter tuning. After experimenting with different settings, we settled for using $k = 20$ eigenvalues and $q = 100$ corresponding functions, which yield best accuracy. We report the tuning experiments in the supplementary material.

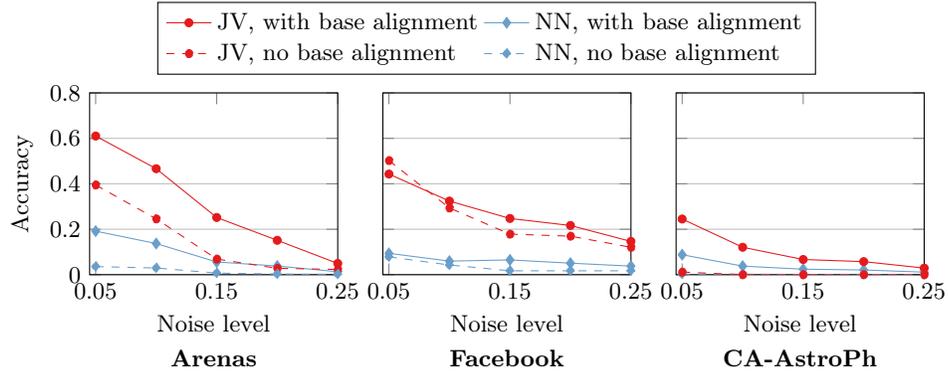


Fig. 3: Accuracy of nearest neighbor and JV matching algorithms.

Justifying algorithmic choices We evaluate the impact of (i) the choice of algorithm for node-to-node assignment (Section 4.4) and (ii) base alignment (Section 4.5). Figure 3 shows that both the JV linear assignment algorithm and base alignment bring a substantial advantage over their rudimentary counterparts, consistently across datasets. In the following experiments, unless otherwise stated, we settle on the variant of GRASP equipped with base alignment.

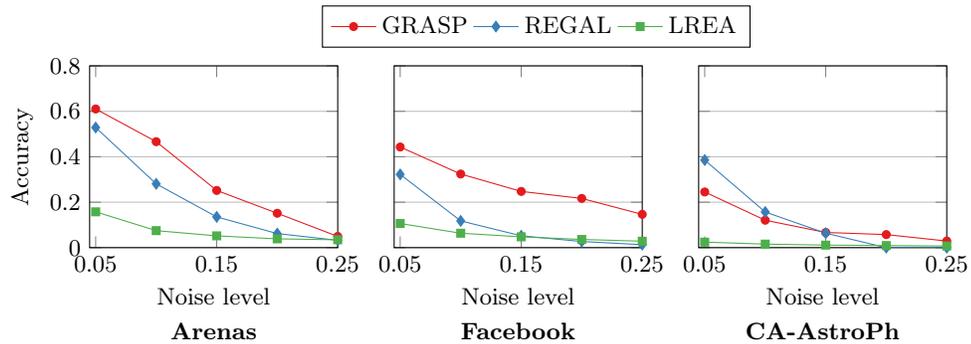


Fig. 4: Accuracy compared to REGAL and LREA

Comparison to previous methods Figure 4 shows that GRASP outperforms others by a large margin, achieving 62% accuracy in Arenas and 43% in Facebook with 5% noise, and fares at comparably well as REGAL on the CA-AstroPH graph.

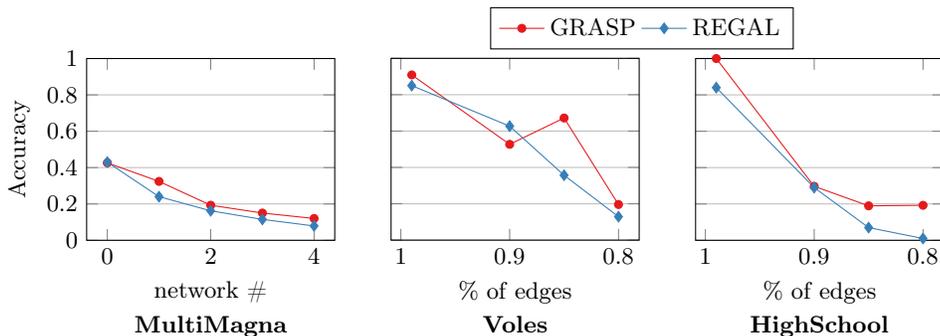


Fig. 5: Accuracy compared to REGAL on three real datasets.

Real world networks We also try matching among real world networks. MultiMagna is a collection of graphs consisting of a base yeast network and five variations thereof. We match these five variations to the original; Figure 5 presents our results. HighSchool and Voles are two evolving proximity networks. We match their latest version to versions at time steps with 80%, 85%, 90%, and 99% of all edges. Figure 5 shows the results. The advantage of GRASP observed with synthetic noise transfers to real-world alignment problems.

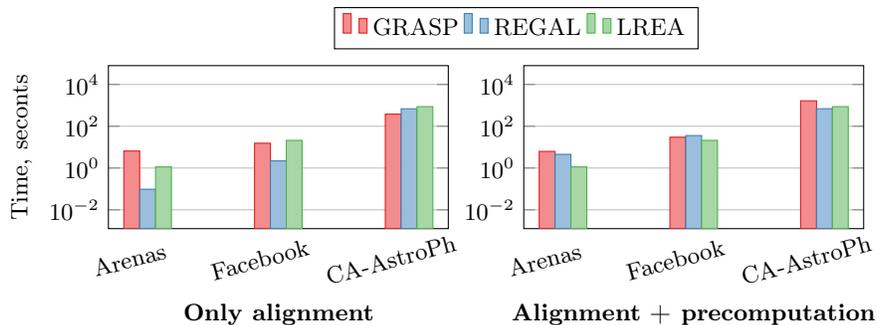


Fig. 6: Time with offline and online precomputation.

Efficiency Steps 1–3 of GRASP (Section 4.6) can be performed offline, while REGAL also allows for precomputation of representations. Figure 6 (left) shows the time to compute the alignments after precomputation. GRASP outperforms REGAL and LREA in the largest CA-AstroPh data. Figure 6 (right) shows the time with online precomputation; REGAL does not exhibit any substantial advantage even in the smaller Arenas and Facebook graphs, while GRASP attains more accurate results with a negligible increase in time. We obtained similar results on real-world network matching tasks.

6 Conclusion

We proposed a graph alignment method using their Laplacian eigenvectors. We establish a functional correspondence among pre-aligned eigenvectors, capturing multiscale graph properties and extract a linear assignment among matrix columns, attaining superior alignment quality over the state of the art. To our knowledge, this is the first work to apply a functional alignment primitive to graph alignment. In the future, we plan to extend our method to partial correspondences and examine to what extent our representations can be employed within the framework of [5].

References

1. Abdulrahim, M.A.: Parallel Algorithms for Labeled Graph Matching. Ph.D. thesis, Colorado School of Mines, USA (1998)
2. Absil, P.A., Baker, C.G., Gallivan, K.A.: Trust-region methods on riemannian manifolds. *FoCM* **7**(3), 303–330 (2007)
3. Bayati, M., Gleich, D.F., Saberi, A., Wang, Y.: Message-passing algorithms for sparse network alignment. *TKDD* **7**(1), 3:1–3:31 (2013). <https://doi.org/10.1145/2435209.2435212>, <https://doi.org/10.1145/2435209.2435212>
4. Belkin, M., Niyogi, P.: Convergence of laplacian eigenmaps. In: *NeurIPS*. pp. 129–136 (2006)
5. Chen, X., Heimann, M., Vahedian, F., Koutra, D.: Cone-align: Consistent network alignment with proximity-preserving node embedding. In: *CIKM*. pp. 1985–1988 (2020)
6. Chu, X., Fan, X., Yao, D., Zhu, Z., Huang, J., Bi, J.: Cross-network embedding for multi-network alignment. In: *TheWebConf*. pp. 273–284 (2019)
7. Chung, F.R., Graham, F.C.: Spectral graph theory. No. 92, American Mathematical Soc. (1997)
8. Cohen-Steiner, D., Kong, W., Sohler, C., Valiant, G.: Approximating the spectrum of a graph. In: *KDD*. pp. 1263–1271 (2018)
9. El-Kebir, M., Heringa, J., Klau, G.W.: Natalie 2.0: Sparse global network alignment as a special case of quadratic assignment. *Algorithms* **8**(4), 1035–1051 (2015)
10. Feizi, S., Quon, G., Mendoza, M., Medard, M., Kellis, M., Jadbabaie, A.: Spectral alignment of graphs. *IEEE Trans. Netw. Sci.* (2019)
11. Gallot, S., Hulin, D., Lafontaine, J.: *Riemannian geometry*. Springer (1990)
12. Heimann, M., Shen, H., Safavi, T., Koutra, D.: REGAL: representation learning-based graph alignment. In: *CIKM*. pp. 117–126 (2018)
13. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* **38**(4), 325–340 (1987)
14. Kazemi, E., Hassani, S.H., Grossglauser, M.: Growing a graph matching from a handful of seeds. *PVLDB* **8**(10), 1010–1021 (2015)
15. Klau, G.W.: A new graph-based method for pairwise global network alignment. *BMC Bioinf.* **10**(S-1) (2009). <https://doi.org/10.1186/1471-2105-10-S1-S59>, <https://doi.org/10.1186/1471-2105-10-S1-S59>
16. Knossow, D., Sharma, A., Mateus, D., Horaud, R.: Inexact matching of large and sparse graphs using laplacian eigenvectors. In: *GdR Int. workshop*. pp. 144–153 (2009)

17. Kobler, J., Schöning, U., Torán, J.: The graph isomorphism problem: its structural complexity. Springer Science & Business Media (2012)
18. Kollias, G., Sathe, M., Mohammadi, S., Grama, A.: A fast approach to global alignment of protein-protein interaction networks. *BMC research notes* **6**(1), 35 (2013)
19. Koutis, I., Levin, A., Peng, R.: Faster spectral sparsification and numerical algorithms for sdd matrices. *TALG* **12**(2), 1–16 (2015)
20. Koutra, D., Tong, H., Lubensky, D.: BIG-ALIGN: fast bipartite graph alignment. In: *ICDM*. pp. 389–398 (2013)
21. Kovnatsky, A., Bronstein, M.M., Bronstein, A.M., Glashoff, K., Kimmel, R.: Coupled quasi-harmonic bases. In: *Comput Graph Forum*. vol. 32, pp. 439–448 (2013)
22. Liao, C.S., Lu, K., Baym, M., Singh, R., Berger, B.: IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics* **25**(12), i253–i258 (2009)
23. Litany, O., Rodolà, E., Bronstein, A.M., Bronstein, M.M.: Fully spectral partial shape matching. *Comput. Graph. Forum* **36**(2), 247–258 (2017)
24. Liu, L., Cheung, W.K., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: *IJCAI*. pp. 1774–1780 (2016)
25. Malmi, E., Gionis, A., Terzi, E.: Active network alignment: a matching-based approach. In: *CIKM*. pp. 1687–1696 (2017)
26. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: *IJCAI* (2016)
27. Nassar, H., Veldt, N., Mohammadi, S., Grama, A., Gleich, D.F.: Low rank spectral network alignment. In: *WWW*. pp. 619–628 (2018)
28. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.J.: Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.* **31**(4), 30:1–30:11 (2012). <https://doi.org/10.1145/2185520.2185526>, <https://doi.org/10.1145/2185520.2185526>
29. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
30. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. *PNAS* **105**(35), 12763–12768 (2008). <https://doi.org/10.1073/pnas.0806627105>
31. Tsitsulin, A., Mottin, D., Karras, P., Bronstein, A.M., Müller, E.: Netlsd: Hearing the shape of a graph. In: *KDD*. pp. 2347–2356 (2018)
32. Umeyama, S.: An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(5), 695–703 (1988)
33. Yartseva, L., Grossglauser, M.: On the performance of percolation graph matching. In: *ACM COSN*. pp. 119–130 (2013)
34. Yasar, A., Çatalyürek, Ü.V.: An iterative global structure-assisted labeled network aligner. In: *KDD*. pp. 2614–2623 (2018)
35. Zhang, S., Tong, H.: FINAL: fast attributed network alignment. In: *KDD*. pp. 1345–1354 (2016)
36. Zhou, F., Liu, L., Zhang, K., Trajcevski, G., Wu, J., Zhong, T.: Deeplink: A deep learning approach for user identity linkage. In: *INFOCOM*. pp. 1313–1321 (2018)