Graph Spectral Embedding using the Geodesic Betweenness Centrality

Shay Deutsch
Discover
shaydeutsch@discover.com [1]

Stefano Soatto
University of California, Los Angeles
soatto@ucla.edu

Abstract

We introduce the Graph Sylvester Embedding (GSE), an unsupervised graph representation of local similarity, connectivity, and global structure. GSE uses the solution of the Sylvester equation to capture both network structure and neighborhood proximity in a single representation. Unlike embeddings based on the eigenvectors of the Laplacian, GSE incorporates two or more basis functions, for instance using the Laplacian and the affinity matrix. Such basis functions are constructed *not* from the original graph, but from one whose weights measure the centrality of an edge (the fraction of the number of shortest paths that pass through that edge) in the original graph. This allows more flexibility and control to represent complex network structure and shows significant improvements over the state of the art when used for data analysis tasks such as predicting failed edges in material science and network alignment in the human-SARS CoV-2 protein-protein interactome.

1 Introduction

Analysis of networks arising in social science, material science, biology and commerce, relies on both local and global properties of the graph that represents the network. Local properties include similarity of attributes among neighboring nodes, and global properties include the structural role of nodes or edges within the graph, as well as similarity among distant nodes. Graph representations based on the eigenvectors of the standard Graph Laplacian have limited expressivity when it comes to capturing local and structural similarity, which we wish to overcome.

We propose an *unsupervised* approach to learn embeddings that exploit both the Laplacian and the affinity matrix combined into a Sylvester equation. However, to better capture local as well as structural similarity, we consider the Laplacian and the affinity matrix *not* of the original graph, but rather of a derived *betweenness centrality graph.* Centrality is a function of the topology of the graph that is not captured in the original edge weights. We choose node embeddings based on the Spectral Kernel Descriptors of the solution of the Sylvester equation, for simplicity, although any number of alternate descriptors could be employed.

The key to our approach is three-fold: (i) The betweenness centrality of edges, defined as the fraction of shortest paths going through that edge, to capture semi-global structure in the original graph; (ii) The use of both the Laplacian and the affinity matrix of the centrality graph, to modulate the effect of local and global structure; (iii) The use of the Sylvester equation as a natural way of combining these two graph properties into a single basis.

Our approach is related to uncertainty principles of graph harmonic analysis, that explore the extent in which signals can be represented simultaneously in the vertex and spectral domains. We extend this approach to characterize the relation between the spectral and non-local graph domain spread of signals defined on the nodes.

We illustrate the use of our method in multiple unsupervised learning tasks including network alignment in protein-to-protein interaction (PPI) for the human-SARS-CoV-2 protein-protein interactome and forecasting failure of edges in material science.

2 Related Work

For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{W})$ with vertices in the set $\mathcal{V} = \{1, 2, ..., N\}$ and edge weights in $\mathcal{W} = \{(w_{ij})(i,j) \in \mathcal{V}\}$, where w_{ij} denotes the weight of an edge between nodes i and j, we measure the *centrality* of an edge by the fraction of the number of shortest paths that pass through that edge, called *edge betweenness centrality* (EBC). From the betweenness centrality of the edges of $\mathcal{G}$, we construct a modified graph with the same connectivity of $\mathcal{G}$, but edges

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

[1] Work done while at University of California, Los Angeles.

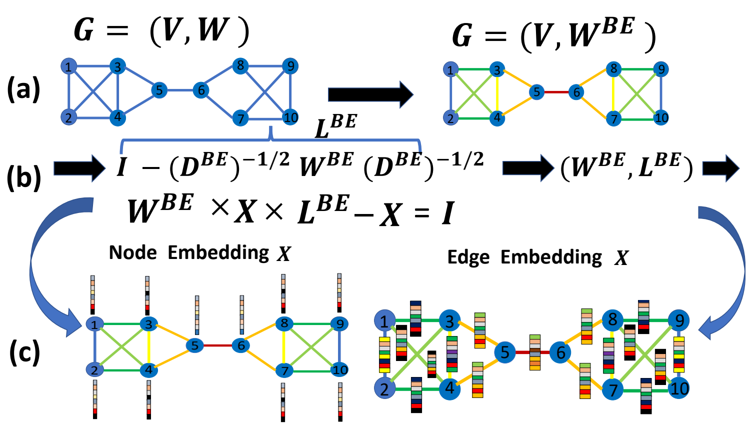


Figure 1: Construction of the Graph Sylvester Embedding (GSE): (a) the original graph is transformed into the centrality graph, by replacing $w_{i,j}$ with $w_{i,j}^{BE}$; (b) $\mathbf{W}^{BE}$ and $\mathbf{L}^{BE}$ are combined into a Sylvester equation, whose solution is used to define (c) node embeddings using the Spectral Kernel, from which edge embedding can be constructed. From the color signatures (best seen zoomed in), we can see that nodes that have different structural properties have different descriptors, whereas nodes with similar structural properties have similar descriptors even when they are not adjacent.

weighted by their EBC value. The result is called edge betweenness centrality graph (BCG), or *centrality graph* for short, and denoted by $\bar{\mathcal{G}}$, which measures the importance of edges given the adjacency matrix.

Graph centrality-based measurements, such as vertex centrality, eigenvector centrality [20], and edge betweenness centrality [30] have been widely used in network analysis and its diverse applications [7, 10]. There is a large body of work that focuses on graph embedding methods that, similar to our approach, aim to learn a function $\Phi : \mathcal{V} \to \mathbb{R}^m$ from the graph nodes into a vector space $\mathbb{R}^m$ [24, 1, 33, 43]. Such methods construct graph representations that are based on either local or structural similarities.

Graph convolutional networks [26, 11, 42] are among the most popular methods for graph-based feature learning, combining the expressiveness of neural networks with the graph structures. Typically, these methods assume that node features or attributes are available, while we focus on problems where only the graph network is available. There is also a large body of work that focuses on graph embedding methods that, similar to our approach, only assume that the affinity matrix is given as an input. Such methods construct graph representations that are based on either local or structural similarities [24, 1, 33, 43]. Most Graph convolutional networks architecture designs capture low-frequency information. Recently, [31] proposed a Gaussian process model using spectral graph wavelets to optimize the model hyperparameters of the wavelets, which was shown to achieve a low approximation error in dense areas of the graph spectrum.

Uncertainty principles in graphs have been explored by [2, 40], extending traditional uncertainty principles of signal processing to more general domains such as irregular and unweighted graphs [35]. [2] provides definitions of the graph and spectral spreads, where the graph spread is defined for an *arbitrary* vertex, and studies to what extent a signal can be localized both in the graph and spectral domains.

Other related work includes different approaches in manifold learning [34, 5, 41], manifold regularization [22, 17, 13, 16, 12, 18, 14, 15], graph diffusion kernels [9], community detection in networks [25] and kernel methods widely used in computer graphics [3, 39] for shape detection. Most methods assume that signals defined over the graph nodes exhibit some level of smoothness with respect to the connectivity of the nodes in the graph and therefore are biased to capture local similarity.

The Sylvester equation was previously employed in a variety of graph mining applications such as network alignment and subgraph matching [44, 19]. Most previous works used the *continuous Sylvester equation* with two similarity input matrices corresponding to the two graphs. We propose a different approach, where we solve the *discrete-time Sylvester equation* using the similarity matrix and its associated Laplacian.

3 Preliminaries and Definitions

Consider an undirected, weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{W})$. The degree $d(i)$ of a node is the sum of the weights of edges connected to it. The combinatorial graph Laplacian is denoted by $\mathbf{L}$, and defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, with $\mathbf{D}$ the diagonal degree matrix with entries $d_{ii} = d(i)$. The eigenvalues and eigenvectors of $\mathbf{L}$ are denoted by $\lambda_1, \ldots, \lambda_N$ and $\phi_1, \ldots, \phi_N$, respectively. The normalized Laplacian is defined as $\mathbf{L}_N = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ and its real eigenvalues are in the interval $[0, 2]$. In this work we use the normalized Laplacian $\mathbf{L}_N$, which from now on we refer to as $\mathbf{L}$ for simplicity.

3.1 Centrality Graph

The edge betweenness centrality (EBC) is defined as

$$w_{i,j}^{BE} = \sum_{s \neq t} \frac{\sigma_{st}(w_{ij})}{\sigma_{st}} \tag{1}$$

where σ_{st} is the number of the shortest distance paths from node s to node t and $\sigma_{st}(w_{ij})$ is the number of those paths that includes the edge w_{ij}. Since we do not use other forms of centrality in this work, we refer to EBC simply as centrality. Accordingly, the (edge betweenness) *centrality graph* BCG is defined as $\bar{\mathcal{G}} = (\mathcal{V}, \mathcal{W}^{\mathcal{BE}})$ which shares the same connectivity but modifies the edges of $\mathcal{G}$ to be $w_{i,j}^{BE}$. Formally, the similarity matrix representing the connectivity of $\bar{\mathcal{G}}$ is

$$(\mathbf{W})_{i,j}{}^{BE} = \begin{cases} \sum_{s \neq t} \frac{\sigma_{st}(w_{ij})}{\sigma_{st}} & \text{if } i \sim j \text{ in } \mathcal{G} \\ 0 & \text{else.} \end{cases} \tag{2}$$

Note that while $\mathcal{G}$ and $\bar{\mathcal{G}}$ share the same connectivity, their spectral representation given by the eigensystem of the graph Laplacian $\mathbf{L}^{BE}$, the eigenvalues and their associated eigenvectors $\lambda_1^{\mathbf{L}^{BE}}, \dots, \lambda_N^{\mathbf{L}^{BE}}$ and $\phi_1^{\mathbf{L}^{BE}}, \dots, \phi_N^{\mathbf{L}^{BE}}$, are rather different. Importantly, the eigenvectors $\phi_1^{BE}, \dots, \phi_N^{BE}$ provide a different realization of the graph structure in comparison to the eigenvectors of the input weights of the graph, that is captured by a different diffusion process around each vertex. We denote with $\Phi^{\mathbf{W}^{BE}}, \Phi^{\mathbf{L}^{BE}}$ the matrices corresponding to the eigenvector decomposition of $\mathbf{W}^{BE}, \mathbf{L}^{BE}$ respectively.

3.2 Structural and local similarity trade-off Via Total Vertex and Spectral Spreads

The eigensystems of the Laplacian $\mathbf{L}^{BE}$ and the affinity matrix $\mathbf{W}^{BE}$ provide two alternative basis functions that capture different properties of the graph structure (see Fig.2 for an illustration). One way to study how basis functions characterize a network's structure and local properties is by using the concepts of vertex spread and spectral spread:

Definition 1 *(Total vertex domain spread)*
The global spread of a non zero signal $\mathbf{x} \in l^2(\mathcal{G})$ with respect to a matrix $\mathbf{W}$ (corresponding to an arbitrary affinity matrix) is defined as

$$\mathbf{g}_\mathbf{W}(\mathbf{x}) = \frac{1}{||\mathbf{x}||^2} \sum_{i \sim j} w_{ij} x(i) x(j) = \frac{1}{||\mathbf{x}||^2} \mathbf{x}^T \mathbf{W} \mathbf{x} \tag{3}$$

where $i \sim j$ corresponds to vertices connected in the similarity graph $\mathbf{W}$.

Definition 2 *(Spectral spread) [2]*
The spectral spread of a non zero signal $\mathbf{x} \in l^2(\mathcal{G})$ with respect to a similarity affinity matrix $\mathbf{W}$ is defined as

$$\mathbf{g}_\mathbf{L}(\mathbf{x}) = \frac{1}{||\mathbf{x}||^2} \sum_{l=1}^{N} \lambda_l |\hat{x}(l)|^2 = \frac{1}{||\mathbf{x}||^2} \mathbf{x}^T \mathbf{L} \mathbf{x} \tag{4}$$

where

$$\hat{x}(l) = \sum_i x(i) \phi_l^*(i) \tag{5}$$

is the graph Fourier transform of the signal $\mathbf{x} \in l^2(\mathcal{G})$ with respect to the eigenvalue λ_l.

The trade-off can be realized by characterizing the domain enclosing all possible pairs of vertex and spectral spreads of the BCG.

Definition 3 *Feasibility domain of $\bar{\mathcal{G}}$*

$$\mathbb{D}_{(s_\mathbf{L}, s_\mathbf{W})} = \{(s_\mathbf{L}, s_\mathbf{W}) |\, g_\mathbf{L}(\mathbf{x}) = s_\mathbf{L}, g_\mathbf{W}(\mathbf{x}) = s_\mathbf{W}\} \tag{6}$$

where $\mathbf{x} \in l^2(\mathcal{G})$.

Specifically, searching for the lower boundary of the feasibility domain of the vertex and spectral spreads of $\bar{\mathcal{G}}$ is shown to yield a generalized eigenvalue problem, whose corresponding eigenvectors produce a representation which trades off local and structural node similarity. We can write the generalized eigenvalue problem (18) using the matrix pencil $\tilde{\mathbf{L}}(\beta)$

$$\tilde{\mathbf{L}}(\beta)\mathbf{x} = \gamma \mathbf{x} \tag{7}$$

where

$$\tilde{\mathbf{L}}(\beta) = \mathbf{W}^{BE} - \beta \mathbf{L}^{BE} \tag{8}$$

and the eigenvector $\mathbf{x}$ solving (18) is also a minimizer for the lower boundary of the curve enclosing the feasibility domain of the spreads (see the appendix for additional details). The scalar $\beta \in \mathbb{R}$ controls the trade-off between the total vertex and spectral spreads. The generalized eigenvalues problem formulated above resembles similar formulations in the literature using the graph Laplacian ([2] and [5]). The formulation suggested above is made in the context of the generalized eigenvalues problem that emerges from the trade-off between local and structural similarity definitions, which was proved in the Appendix for completeness.

An illustration of how local and global structures are captured by the Laplacian and the similarity graph in the noise-free case of the Barbell graph is shown in Fig.2. The eigenvector associated with the smallest non-zero eigenvalues of the graph Laplacian $\mathbf{L}^{BE}$ (blue color, Fig.2(b)) captures the global structure while the eigenvector of $\mathbf{W}^{BE}$ (Fig.2(c)) provides complementary information which emphasizes structural similarity. The proposed embedding (detailed in the next section) is obtained via the solution of the Sylvester embedding modulates the effect of both.

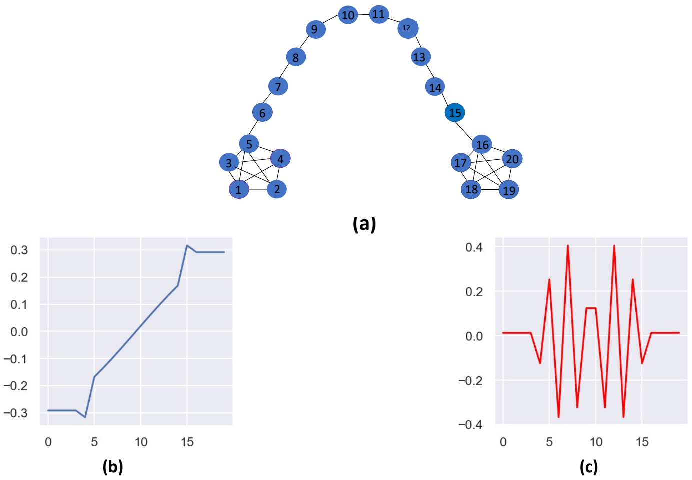


Figure 2: Illustration of the trade-off between local and global structures node similarity in the barbell graph (a) using different basis functions. Figures (b) and (c) illustrate the entries of the eigenvectors $\phi_1^{\mathbf{L}^{BE}}$ and $\phi_1^{\mathbf{W}^{BE}}$ of $\mathbf{L}^{BE}$ and $\mathbf{W}^{BE}$, respectively. The solution $\mathbf{X}$ to the Sylvester equation (with input matrices $\mathbf{L}^{BE}$ and $\mathbf{W}^{BE}$) modulates the effect of local similarity, connectivity, and global structure.

4 Proposed Graph Sylvester Embedding

We suggest an embedding that goes beyond the scalar modulation of the generalized Laplacian Eq.(18) by using a linear mapping between the subspaces $\mathbf{W}^{BE}$ and $\mathbf{L}^{BE}$. We exploit the fact that the correspondence between the nodes represented by the two basis functions of $\mathbf{W}^{BE}$and $\mathbf{L}^{BE}$ is known, i.e., it is the identity map, $\pi : \mathbf{W}^{BE} \rightarrow \mathbf{L}^{BE}$ with $\pi(\mathbf{W}^{BE}(:,i)) = \mathbf{L}^{BE}(:,i)$ for each $i \in \mathcal{V}$ (i.e., the column index corresponding to each node $i \in \mathcal{V}$ in $\mathbf{W}^{BE}$ is the same in $\mathbf{L}^{BE}$). We propose using the solution of a Sylvester equation as the node feature representation, which is also the mapping between the nodes' representations. The resulting embedding is given by the solution to the discrete Sylvester equation (or the singular value decomposition associated with the solution) and will be composed of two hybrid representations associated with the graph networks and local connectivity. The proposed method is coined Graph Sylvester Embedding (GSE).

The *discrete-time* Sylvester operator $\mathbf{S}(\mathbf{X}) = \mathbf{AXB} - \mathbf{X}$ is used to express the eigenvalues and eigenvectors of $\mathbf{A}$ and $\mathbf{B}$ using a single operator $\mathbf{S}$, where we solve

$$\mathbf{S}(\mathbf{X}) = \mathbf{AXB} - \mathbf{X} = \mathbf{C} \tag{9}$$

using $\mathbf{C} = \mathbf{I}$, ($\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix), $\mathbf{A} = \mathbf{W}^{BE}, \mathbf{B} = \mathbf{L}^{BE}$.

Let $\left\{\phi_i^{\mathbf{W}^{BE}}\right\}_{l=1}^{N}$, $\left\{\lambda_l^{\mathbf{W}^{BE}}\right\}_{l=1}^{N}$, and $\left\{\phi_i^{\mathbf{L}^{BE}}\right\}_{l=1}^{N}$, $\left\{\lambda_l^{\mathbf{L}^{BE}}\right\}_{l=1}^{N}$ be the corresponding eigenvectors and eigenvalues of $\mathbf{A}$ and $\mathbf{B}$, respectively.

We will assume that the eigenvalues of $\mathbf{A}$ and $\mathbf{B}$ satisfy that $\lambda_i^{\mathbf{L}^{BE}}\lambda_j^{\mathbf{W}^{BE}} \neq 1$, $\forall i, j$, hence the operator $\mathbf{S}(\mathbf{X}) : \mathbb{R}^{N\times N} \rightarrow \mathbb{R}^{N\times N}$ is non-singular and has N^2 matrix eigenvalues and eigenvectors. Note that the Sylvester equation has a unique solution for any $\mathbf{C}$ if and only if $\mathbf{S}$ is non-singular, which occurs if and only if $\lambda_i^{\mathbf{L}^{BE}}\lambda_j^{\mathbf{W}^{BE}} \neq 1 \, \forall i, j$. By choosing $\mathbf{C} = \mathbf{I}$ and ordering the columns in $\mathbf{A}$ and $\mathbf{B}$ based on the identity map, $\pi : \mathbf{A} \rightarrow \mathbf{B}$ we obtain that each column in $\mathbf{X}$ provides a node representation to its corresponding node $i \in \mathcal{V}$.

Note that we can also express the solution $\mathbf{X}$ to the Sylvester equation Eq.(9) using the Kronecker product $\otimes$:

$$(\mathbf{I} - \mathbf{B}^T \otimes \mathbf{A})x = c \tag{10}$$

where we vectorize $\mathbf{X}$ and $\mathbf{C}$ to obtain its equivalent vector representation x = vec $(\mathbf{X})$ and c = vec$(\mathbf{C})$. Since the Sylvester equation can be written using the Kronecker product, then all known properties of the Kronecker product are carried on to the proposed graph Sylvester embedding. One important property is invariance to permutations.

Proposition 1 *Suppose that* $\mathbf{P} \in \mathbb{R}^{N\times N}$ *is a permutation matrix,* $\tilde{\mathbf{A}} = \mathbf{P}^T\mathbf{AP}$*, and* $\tilde{\mathbf{B}} = \mathbf{P}^T\mathbf{BP}$*, and assume that* $\mathbf{X}, \tilde{\mathbf{X}}$ *are two solutions to the discrete-time Sylvester equation Eq.(9) with the associate matrices* $\mathbf{A}, \mathbf{B}$ *and* $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$*, respectively, and* $\lambda_i^{\mathbf{L}^{BE}}\lambda_j^{\mathbf{W}^{BE}} \neq 1$*,* $\forall i, j$*, then*

$$\tilde{\mathbf{X}} = \mathbf{P}^T\mathbf{XP} \tag{11}$$

Proof: See Appendix.

The resulting solution of $\mathbf{X}$ can be described using the following conditions. Assume that $\lambda_i^{\mathbf{W}^{BE}}\lambda_j^{\mathbf{L}^{BE}} \neq 1 \, \forall i, j$, and that the graph representation encoded in $\mathbf{A}$ and $\mathbf{B}$ has the

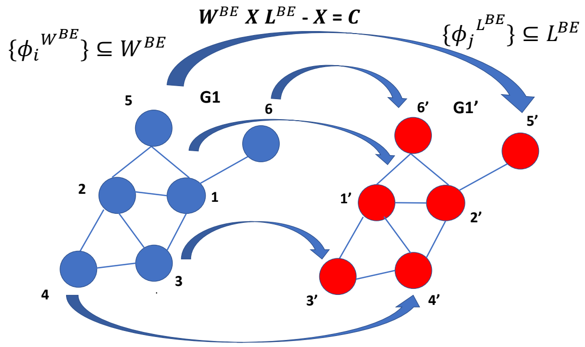

W^{BE} X L^{BE} - X = C
$\{\phi_i{}^{W^{BE}}\} \subseteq W^{BE}$
$\{\phi_j{}^{L^{BE}}\} \subseteq L^{BE}$
G1
G1'
5
6
2
1
4
3
6'
5'
1'
2'
3'
4'

Figure 3: In the proposed Graph Sylvester Embedding (GSE), the mapping between two basis functions is used to construct the proposed embedding, exploiting the known correspondence between the two alternative graph representations.

same order based on the identity map, and $\mathbf{C} = \mathbf{I}$. Then the solution $\mathbf{X}$ to the discrete-time Sylvester system Eq.(9) is unique where

$$\mathbf{X} = \mathbf{U}^{\mathbf{W}^{BE}} \tilde{\mathbf{C}} (\mathbf{V}^{\mathbf{L}^{BE}})^T \tag{12}$$

where $\tilde{\mathbf{C}}_{i,j} = \frac{(\phi_i^{\mathbf{W}^{BE}})^t \phi_j^{\mathbf{L}^{BE}}}{\lambda_i^{\mathbf{W}^{BE}} \lambda_j^{\mathbf{L}^{BE}} - 1}$. See proof in the appendix.

Remark 1: Eq. (12) shows the analytical solution to the Sylvester equation. In practice, one can use fast iterative methods [28]. Also note that under the assumptions that $\lambda_i^{\mathbf{W}^{BE}} \lambda_j^{\mathbf{L}^{BE}} \neq 1 \, \forall i, j$, the solution $\mathbf{X}$ is unique [28]. We note that Sylvester's equation can be generalized to include multiple terms, thus allowing one to incorporate multiple basis functions. We illustrate the concept with two bases, to capture local and global statistics of the topology of the network, and leave extension to additional bases, that can be specific to particular domains or tasks, for future work.

Remark 2: One can view the solution $\mathbf{X}$ to the Sylvester equation as a polynomial of the matrices $\mathbf{A}$ and $\mathbf{B}$ ([23]), where using higher-order polynomials in $\mathbf{W}^{BE}$, $\mathbf{L}^{BE}$ can be interpreted as imposing smoothness in the embedding space.

Remark 3:Our analysis shows that the proposed transformation is invariant to permutation, which is an important property for ensuring stable graph embedding. Specifically, this property implies that nodes with the same topological neighborhood will yield the same output. Note that while it is a necessary condition in order to generalize well to new data or applications, it doesn't provide classification guarantees.

4.1 Spectral Representation of GSE

By our specific construction of matrices in the Sylvester equation Eq.(12) we have that the ith row of the solution $\mathbf{X}$ captures local and global statistics of the topology of the network with respect to the node i. This is an important property which was achieved by our choice of $\mathbf{C} = \mathbf{I}$, and using the same order of the graph representation encoded in $\mathbf{W}^{BE}$ and $\mathbf{L}^{BE}$. It is now possible to construct graph embedding from $\mathbf{X}$. We employ graph embedding using the spectral decomposition of $\mathbf{X}$.

Given the Singular Value Decomposition $\mathbf{X} = U \Lambda V^*$ and equally spaced scales $\{t_s\}$, we compute a node embedding for each node $i \in \mathcal{G}$ using the Spectral Kernel descriptor of $\mathbf{X}$ using

$$\psi(\mathbf{x}_i, t_s) = \sum_{l=1}^{m} c_{t_s} \exp\left(- \frac{(\log(t_s) - \log(\lambda_l))^2}{2\sigma^2} \right) (\mathbf{u}_l)^2(i) \tag{13}$$

where $U = \{\mathbf{u}_l\}$, $V = \{\mathbf{v}_l\}$, and $\Lambda = \{\lambda_l\}$ correspond to the left and right singular eigenvectors, and singular values, respectively, m is the number of the largest eigenvalues its associated eigenvectors used in the SVD decomposition, and c_{t_s} are normalizing constants.

Remark: Note that the spectral kernel descriptor in Eq.(13) is similar to the WKS descriptor [3] proposed to describe the spectral signature of the Laplace- Beltrami operator (LBO). For applications using Sylvester embedding shown in this work, we find the descriptor in Eq.(13) effective as it weighs the eigenvalues equally and separates the influence between different eigenvalues.

4.2 Spectral Graph Wavelet Descriptor

We demonstrate that while we used node embeddings that is based on the Spectral Kernel Descriptors of the solution of the Sylvester equation, different descriptors could be built on this basis. The basis functions and descriptors would ideally capture a unique signature of invariant geometric structure in multiple scales. The basis functions and descriptors would ideally capture a unique signature of invariant geometric structure in multiple scales. Capturing such a unique signature is very challenging for high dimensional noisy data. Spectral graph wavelet (SGW) transform [21] provide a useful tool to construct robust descriptors, due to their their vertex and spectral localization. We describe an

Algorithm 1: Graph Sylvester Embedding (GSE)

```
Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{W})$, embedding dimension $m$.
equally spaced scales $t_s$.
Output: Node Embedding $\mathbf{y}_i \in \mathbf{R}^m$ for each $i \in \mathcal{G}$
Step 1: Compute the $\bar{\mathcal{G}}$ using Eq.(2).
Step 2: Compute
$\mathbf{L}^{BE} = \mathbf{I} - (\mathbf{D}^{BE})^{-\frac{1}{2}}\mathbf{W}^{BE}(\mathbf{D}^{BE})^{-\frac{1}{2}}$ using $\bar{\mathcal{G}}$
computed in Step 1.
Step 3: Solve the discrete-time Sylvester equation:

        $\mathbf{AXB} - \mathbf{X} = \mathbf{C}$        (15)

using $\mathbf{A} = \mathbf{W}^{BE}$, $\mathbf{B} = \mathbf{L}^{BE}$, and $\mathbf{C} = \mathbf{I}$ .
Step 4: Compute the largest $m$ singular values and
associated singular vectors of $\mathbf{X} = U\Lambda V^*$.
Step 5: For all $i \in \mathcal{G}, \forall s = 1, 2, ..m$, compute the
spectral kernel descriptor $\psi(\mathbf{x}_i, t_s)$ in Eq.(13) using $t_s$
and the largest $m$ singular vectors and values
corresponding to $U$and $\Lambda$, respectively. Alternatively,
compute the Spectral Graph Wavelet descriptor
described in Sec. 4.2.
```

alternative way to construct the spectral descriptors which is based on the Spectral Graph Wavelets. The descriptors are constructed by taking the spectral graph wavelet transform using functions corresponding to each of the fixed dimensions of the the solution of the Sylvester equation $\mathbf{X}$. We describe the SGW transform for a fixed dimension m in $\mathbf{X}$. Given $\mathbf{x}_m \in \mathbb{R}^N$, a scalar function defined on $\mathcal{G}$, corresponding to the m'th fixed dimension of $\mathbf{X}$, we compute

$$\begin{aligned}\Psi_{\mathbf{x_m}}(i, t_j) &= \sum_{l=1}^{N} \kappa(t_j\lambda_l)\hat{\mathbf{x}}_{\mathbf{m}}(\lambda_l)\phi_l(i) \\ &= \sum_{l=1}^{N} \kappa(t_j\lambda_l) \sum_{j=1}^{N} \mathbf{x}_m(j)\phi_l(j)\phi_l(i)\end{aligned} \tag{14}$$

where $\kappa(t_j\lambda_l)$ is a smooth kernel function, and $\hat{\mathbf{x}}_m(\lambda_l)$ denotes the graph Fourier transform of $\mathbf{x}_m$. For a fixed dimension m we obtain J descriptors which aggregate information from all the J spectral bands, while also being localized in the vertex domain. Thus for an input vector corresponding to $\mathbf{x}_m$ we obtain as an output J vectors in $\mathbb{R}^N$. Then concatenate all the computed spectral graph transforms $\{\Psi_{\mathbf{x}_m}(:, t_j)\}$, in scales $t_j, 1 \leq j \leq J$, for each of the corresponding $\mathbf{x}_m$, $1 \leq\ m \leq N$, in a matrix form.

5 Experimental Results

We evaluate our method on real-world networks in several applications including material science and network alignment of Protein-Protein interactions (PPIs) networks using a recent dataset that has been used to study Covid-19, the SARS-CoV-2. We compare GSE to known and existing state of the art methods in the respective applications.

Evaluation metrics: In the problem of network alignment we use the percentage of correct node correspondence found to evaluate our method, based on the known ground truth correspondence between the two networks. In detecting failure edges in material science we measure the success rate using sensitivity. To detect failured edges, we use the graph embedding obtained by each method to cluster the data into two approximately equal size clusters using spectral clustering . We then choose the cluster with the largest betweenness centrality mean value. For further validation, we test the statistical significance of each method by computing the p values using hyper-geometric distribution (see the Appendix). Note that GSE(Wave kernel) corresponds to GSE computed using the spectral Wave Kernel Descriptor signature (WKS) and GSE(SGW) corresponds to the GSE computed using the Spectral Graph Wavelet descriptor described in Sec. 4.2.

Comparison with other methods: Our baseline for comparison with the proposed Graph Sylvester Embedding is using the concatenated spectral embedding corresponding to both the Laplacian and the affinity matrix. We coin the concatenated bases as St.($\mathbf{W}^{BE}, \mathbf{L}^{BE}$). We compare to the spectral Wave Kernel Descriptor signature (WKS) [3] (coined $\mathbf{L}$ desc.). $(-)$ in the Tables indicates methods that failed to converge or provided meaningless results.

We compare with a representative of graph embeddings methods including Laplacian Eigenmaps [5], node2vec [1], and NetMF [24]. We also test against methods that were specifically tailored to this applications we explored including methods for graph alignments [44, 45] and methods based on betweenness centrality for detecting failured edges [7].

5.1 Detect Failed Edges in Material Sciences

Forecasting fracture locations in a progressively failing disordered structure is a crucial task in structural materials [7]. Recently, networks were used to represent 2 dimensional (2D) disordered lattices and have been shown a promising ability to detect failures locations inside 2D disordered lattices. Due to the ability of BC edges that are above the mean to predict failed behaviors [7], we expect the edge embedding based on EBC to be effective. As shown in the experimental results below, our proposed *edge* embedding improves robustness in comparison to simpler methods such as the one employed in [7] or methods that are based on "standard" Laplacian embedding. We first describe how to employ edge embedding (proposed node embedding detailed in Sect. 4) for this task.

Edge Embedding using GSE: Edge embedding is constructed by first applying our node embedding and then using the concatenated nodes features to construct the *edge* embeddings. To forecast the failed edges, we construct a new graph where nodes correspond to the edge embedding features and then apply spectral clustering to cluster the edges into two groups. We apply the same strategy to test spectral descriptors which are based on the graph Laplacian.

Dataset: The set of disordered structures was derived from experimentally determined force networks in granular materials [6]. The network data is available in the Dryad repository [8]. We tested 6 different initial networks, with mean degrees $z = 2.40, 2.55, 2.60, 3.35, 3.0, 3.6$, following the same datasets corresponding to different initial granular configurations.

Implementation Details: In all experiments, we used $c_t = 1$, for all t, c_t corresponds to the coefficients in the wave kernel signature WKS (Eq. (13)). We used a fixed number of total 800 scales t_s in all experiments. We measure the success rate in detecting failure edges in terms of the sensitivity (true positive rate). We report experimental results for a varying number of singular eigenvectors $\mathbf{u}_l$ in the SVD of the solution $\mathbf{X}$ to the Sylvester equation, which is then employed to compute edge embeddings (as described above, based on the node embeddings detailed in Algorithm 1).

Table 1 shows a comparison of our method where we report the results using $m = 800$ for all methods which are based on eigensystem computation (this the same number of eigenvectors and associated eigenvalues computed for the spectral methods which are based on graph Laplacian such as LE and spectral descriptors).

Note that the networks tested in Table 1 are all extremely sparse, but their geometry is different. The high variance in the success detection rates of failed edges for some methods in Table 1 is related to the differences in the structure of the network topology in the tested networks. Incorporating a model that trades off between local similarity, connectivity, and global structure, our method was shown to be more robust when tested on different network topologies. Our framework relies on the edge betweenness centrality which is shown to be an effective measure to capture the network topology; however, it is non-local and prone to noise which affects the results on the networks due to the differences in their topology.

5.2 Network Alignment on PPI networks with human protein interactors of SARS-CoV-2

Effective representation of nodes with similar network topology is important for network alignment applications, where the network structure around each node provides valuable information for matching and aligning the networks. Network alignment of Protein-protein interactions (PPIs) networks is considered as an important first step for the analysis of biological processes in computational biology. Typically, popular methods in this domain gather statistics about each node in the graph (e.g: node degree) followed by an optimization methods to align the network. For PPI networks, it is assumed that a protein in one PPI network is a good match for a protein in another network if their respective neighborhood topologies are a good match [36]. In this case, features or nodes attributes are in general not available which makes this task of network alignment and finding node correspondence more challenging, with only the graph network to rely on.

Dataset: We test using the STRING network from the STRING database, a PPI network that consists of 18,886 nodes and 977,789 edges. The STRING network includes 332 human proteins that physically interact with SARS-CoV-2. It is likely to have both false positives and false negatives edges. We aim to find network alignment between two copies of the STRING network, using an additional STRING network which is created by randomly removing 10% of the network edges. We also test on the Lung network [38], which is a more recent PPI networks with known human protein interactors of SARS-CoV-2 proteins that consists of 8376 nodes, 48522 edges, and 252 human proteins that physically interact with SARS-CoV-2.

Implementation details: Given the node correspondence of the protein interactors of SARS-CoV-2, we connect each pair of nodes corresponding to the same human protein interactors of SARS-CoV-2 with an edge, which is resulted in a network composed of the two STRING networks. We then apply the proposed GSE to generate node embeddings. We used 5-fold cross validation, where in each experiment we used 50% of the known human proteins in the STRING and Lung networks.

We compare to network alignment methods, such as Isorank [36], Final [44] and iNEAT [45]. For the network alignments methods, we provide as an input the corresponding affinity graphs and a matrix with the known correspondence between the nodes that correspond to human proteins that physically interact with SARS-CoV-2. Additionally, we also tested graph embeddings including Laplacian Eigenmaps (LE) [4], Locally Linear Embedding (LLE) [34], Hope [32], and node2vec [1] which are given the same input network. Graph embedding methods are not effective for this task and thus some of the results are omitted. Network alignments methods perform better, while our proposed GSE outperforms all competing methods.

Table 2 shows the percentage of nodes whose nearest neighbor corresponds to its true node correspondence using GSE in the STRING and Lung networks. Our proposed GSE outperforms all competing methods.

Method /Network	Mean deg. (2.4)	Mean deg. (2.6)	Mean deg. (3.35)	Mean deg. (2.55)	Mean deg. (3)	Mean deg. (3.6)	Avg. ± std
FL[7]	85.7%	70%	60%	58.3%	48.4%	58.1%	63.4% ± 11.7
NetMF	28 %	40 %	72 %	45 %	**93.9** %	90.0 %	61.4% ± 25.2
node2vec	64.2%	55%	40%	79.0 %	69.6%	37.2%	61.4% ± 15.1
LE	64.2%	75%	68.0%	66.6%	63.6%	**83.7**%	70.1% ± 6.5
$\mathbf{L}$ desc.	57.1%	45%	84.0%	45.0%	54.5%	37.2%	53.8% ± 15
St.($\mathbf{W}^{BE}$, $\mathbf{L}^{BE}$)	64.2%	75.0%	80.0%	75.0%	78.7%	79.0%	75.3% ± 27.6
GSE(Wave kernel)	92.8%	80 %	76.0%	75.0%	72.2%	83.7%	79.9 % ± 6.8
GSE(SGW)	**92.8**%	**95**%	**96** %	**83.3** %	72.7%	81.4 %	**86.8**% ± 8.4

Table 1: Success rate (%) in detecting failed edges in granular material networks, comparing GSE to different methods. The network number associated with each Network in parenthesis (Mean deg.) corresponds to the characteristic of the network, given by its mean degree of edges per node. The column below Avg ± std shows the average and standard deviation summarized over all networks.

Table 2: Classification accuracy on network alignment using PPI interaction networks: (STRING(s) and Lung.

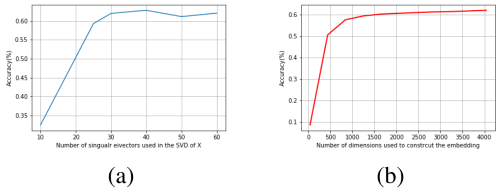
Method/Data	STRING (s)	Lung
node2vec	10.4 %	- %
IsoRank	23 %	44.2 %
IsoRank using $\bar{\mathcal{G}}$	20 %	44.2 %
Final	36 %	50.5 %
iNeat	37.0 %	53.5 %
St.($\mathbf{W}^{BE}$, $\mathbf{L}^{BE}$)	25.3 %	53.2 %
GSE	**61.2**%	**55.0 %**

6 Discussion

We have presented an embedding to represent the local and global structure of a graph network, constructed without the need for any supervision nor explicit annotation. Besides the cost of time and effort, annotating data can create privacy and security risks. Our focus is on developing expressive and flexible representations that can be used in a variety of downstream tasks without human intervention. Flexibility is exercised through the choice of bases, that are combined through Sylvester's equation. In the specific cases we have experimented with, the bases are chosen to capture local and structural similarity across the graph network.

Most of the computational burden of our method falls on the computation of the BCG, which is $O(mN)$ where m is the total number of edges and N is the total number of nodes, thus approximately $O(N^2)$ for sparse graphs. The execution time of our method with Python code implementation using Intel Core i7 7700 Quad-Core 3.6GHz with 64B Memory on a graph network with approximately 2700 nodes and 5429 edges takes about 24.9 seconds.

Computing BC measurements on large graphs is computationally heavy ($O(N^2)$ for sparse graphs), and developing fast methods for BC measurements is an ongoing research area. With significant progress that has been made in recent years (e.g: [29]) we can foresee extensions to scale our approach to networks with millions of nodes. In addition, the effectiveness of BC measurements degrades on larger graphs due to noise (for example when the graph includes a large number of nodes with clustering coefficient close to zero). Applying our approach to small sub-graphs before aggregating it to the entire graph may be one viable solution.



(a) (b)

Figure 4: Sensitivity to parameter selection in network alignment of the STRING network, concerning the number of singular eigenvectors and the number of scales used in the spectral signature of GSE (a) Accuracy (in percentage of nodes whose nearest neighbor corresponds to its true node correspondence) as a function of the number of singular eigenvectors used in the SVD of $\mathbf{X}$ in the network alignment of STRING. (b) Shows the accuracy as a function of the number of scales which was used to construct the Spectral Kernel descriptor for the network alignment of the STRING network.

In our study, we found that spectral embeddings obtained by solving the Sylvester equation of the edge betweenness centrality graph reveals the node's network structure, which is not necessarily possible with the original graph weights. It is also possible to compute edge betweenness centrality in a way that takes into account the original weight information. For example, by looking into the path evaluation function that assesses a path between two nodes that is combining both the sum of edge weights and the number of shortest distance path as was proposed in [37]. Another possible direction is to solve a generalized Sylvester equation which would incorporate both matrices corresponding to the original weight information and the edge betweenness centrality. Other future work includes expanding our approach to address applications that include dynamic graphs and multi-layer graphs, which would include solving Sylvester equation with time-varying coefficients.

Acknowledgements

Research supported by ONR N00014-22-1-2252:P00001 and ARO W911NF-17-1-0304. We thank all the referees for their numerous constructive comments and helpful suggestions. We wish to thank Mason Porter for helpful discussions and suggestions in the earlier formulation of the manuscript. We also want to thank Yoram Louzoun, Roded Sharan, Eran Halperin, and Jie Chen for their valuable feedback on the manuscript. We are also thankful to Simon Kasif, T. M. Murali, Mark Crovella, Catherine M. Della Santina, and Jeffrey Law for sharing insights from their work on Identifying Human Interactors of SARS-CoV-2 Proteins for COVID-19 using network label propagation, as well as sharing the STRING dataset.

References

[1] node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

[2] Ameya Agaskar and Yue M. Lu. A spectral graph uncertainty principle. *IEEE Transactions on Information Theory*, 59(7):4338–4356, 2013.

[3] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*, pages 1626–1633. IEEE Computer Society, 2011.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[5] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 2003.

[6] Estelle Berthier, Jonathan E. Kollmer, Silke E. Henkes, Kuang Liu, J. M. Schwarz, and Karen E. Daniels. Rigidity percolation control of the brittle-ductile transition in disordered networks. *Phys. Rev. Materials*, 2019.

[7] Estelle Berthier, Mason A. Porter, and Karen E. Daniels. Forecasting failure locations in 2-dimensional disordered lattices. *Proceedings of the National Academy of Sciences*, 116(34):16742–16749, Aug 2019.

[8] Estelle et al Berthier. Rigidity percolation control of the brittle-ductile transition in disordered networks dryad, dataset,. *Phys. Rev. Materials*, 2019.

[9] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 2005.

[10] Mihai Cucuringo, Puck Romback, Sang Hoon Lee, and Mason A. Porter. Detection of core–periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics*, page 846–887, 2016.

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29*. 2016.

[12] Shay Deutsch, Andrea L. Bertozzi, and Stefano Soatto. Zero shot learning with the isoperimetric loss. In *AAAI 2020*.

[13] Shay Deutsch, Soheil Kolouri, Kyungnam Kim, Yuri Owechko, and Stefano Soatto. Zero shot learning via multi-scale manifold regularization. *CVPR*, 2017.

[14] Shay Deutsch and Gerard Medioni. Intersecting manifolds: Detection, segmentation, and labeling. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[15] Shay Deutsch and Gerard Medioni. Unsupervised learning using the tensor voting graph. In *Scale Space and Variational Methods in Computer Vision (SSVM)*, 2015.

[16] Shay Deutsch and Gérard G. Medioni. Learning the geometric structure of manifolds with singularities using the tensor voting graph. *Journal of Mathematical Imaging and Vision*, 57(3):402–422, 2017.

[17] Shay Deutsch, Antonio Ortega, and Gerard Medioni. Manifold denoising based on spectral graph wavelets. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[18] Shay Deutsch, Antonio Ortega, and Gérard G. Medioni. Robust denoising of piece-wise smooth manifolds. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 2786–2790, 2018.

[19] Boxin Du and Hanghang Tong. Fasten: Fast sylvester equation solver for graph mining. KDD '18, page 1339–1347, 2018.

[20] L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 1977.

[21] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 2011.

[22] Matthias Hein and Markus Maier. Manifold denoising. In *Advances in Neural Information Processing Systems*, 2006.

[23] Qingxi Hu Hu and Daizhan Cheng. The polynomial solution to the sylvester matrix equation. *Applied Mathematics Letters*, 2006.

[24] Qiu Jiezhong, Dong Yuxiao, Ma Hao, Kuansan Jian, and Tang Jie. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh International Conference on Web Search and Data Mining, WSDM*, 2018.

[25] Jiashun Jin. Fast community detection by SCORE. *The Annals of Statistics*, feb 2015.

[26] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[27] Jeffrey N Law, Kyle Akers, Nure Tasnina, Catherine M Della Santina, Shay Deutsch, Meghana Kshirsagar, Judith Klein-Seetharaman, Mark Crovella, Padmavathy Rajagopalan, Simon Kasif, and T M Murali. Interpretable network propagation with application to expanding the repertoire of human proteins that interact with sars-cov-2. In *GigaScience*, 2021.

[28] Tiexiang Li, Peter Chang-Yi Weng, Eric King-Wah Chu, and Wen-Wei Lin. Large-scale stein and lyapunov equations, smith method, and applications. *Numer. Algorithms*, 2013.

[29] S. Maurya, Xin Liu, and T. Murata. Fast approximations of betweenness centrality with graph neural networks. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[30] M E Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 2002.

[31] Felix Opolka, Yin-Cong Zhi, Pietro Lió, and Xiaowen Dong. Adaptive gaussian processes on graphs via spectral graph wavelets. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*. PMLR, 28–30 Mar 2022.

[32] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *KDD*, 2016.

[33] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, 2014.

[34] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[35] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 2013.

[36] Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 2008.

[37] Jadwiga Sosnowska and Oskar Skibski. Path evaluation and centralities in weighted graphs - an axiomatic approach. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3856–3862. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[38] Alexey Stukalov, Virginie Girault, Vincent Grass, Valter Bergant, Ozge Karayel, Christian Urban, Darya A. Haas, Yiqi Huang, Lila Oubraham, Anqi Wang, Sabri M. Hamad, Antonio Piras, Maria Tanzer, Fynn M. Hansen, Thomas Enghleitner, Maria Reinecke, Teresa M. Lavacca, Rosina Ehmann, Roman Wölfel, Jörg Jores, Bernhard Kuster, Ulrike Protzer, Roland Rad, John Ziebuhr, Volker Thiel, Pietro Scaturro, Matthias Mann, and Andreas Pichlmair. Multi-level proteomics reveals host-perturbation strategies of sars-cov-2 and sars-cov. *bioRxiv*, 2020.

[39] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 2009.

[40] O. Teke and P. P. Vaidyanathan. Uncertainty principles and sparse eigenvectors of graphs. *IEEE Transactions on Signal Processing*, 65(20):5406–5420, 2017.

[41] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

[42] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[43] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, 2017.

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

[45] Si Zhang, Hanghang Tong, Jie Tang, Jiejun Xu, and Wei Fan. Incomplete network alignment: Problem definitions and fast solutions. *ACM Trans. Knowl. Discov. Data*, 14(4), 2020.

[46] Dengyong Zhou and B. Schölkopf. A regularization framework for learning from graph data. In *ICML 2004*, 2004.

A Appendix

The lower boundary of the curve enclosing the feasibility domain of the spreads $(\mathbf{s}_{\mathbf{L}^{BE}}(\mathbf{x}), \mathbf{s}_{\mathbf{W}^{BE}}(\mathbf{x}))$, with respect to a unit norm vector $\mathbf{x} \in l^2(\mathcal{G})$ is defined as

$$\begin{cases} \Gamma_{\mathbf{s}_{\mathbf{W}^{BE}}}(\mathbf{s}_{\mathbf{L}^{BE}}) = \min_{\mathbf{x}} g_{\mathbf{W}^{BE}}(\mathbf{x}) \\ \text{s.t } g_{\mathbf{L}^{BE}}(\mathbf{x}) = \mathbf{s}_{\mathbf{L}^{BE}} \text{ and } \mathbf{x}^T\mathbf{x} = 1 \end{cases} \tag{16}$$

To solve (16), we use the Lagrangian

$$L(\mathbf{x}, \beta, \gamma) = \ \mathbf{x}^T\mathbf{W}^{BE}\mathbf{x} - \beta(\mathbf{x}^T\mathbf{L}^{BE}\mathbf{x}) - \gamma(\mathbf{x}^T\mathbf{x} - 1) \tag{17}$$

with $\beta \in \mathbb{R}$. Differentiating and comparing to zero, we obtain the following eigenvalue problem:

$$(\mathbf{W}^{BE} - \beta\mathbf{L}^{BE})\mathbf{x} = \gamma\mathbf{x} \tag{18}$$

where the eigenvector $\mathbf{x}$ solving (18) is also a minimizer for (16). Denoting

$$\tilde{\mathbf{L}}(\beta) = \mathbf{W}^{BE} - \beta\mathbf{L}^{BE} \tag{19}$$

we can write the generalized eigenvalue problem (18) using the matrix pencil $\tilde{\mathbf{L}}(\beta)$

$$\tilde{\mathbf{L}}(\beta)\mathbf{x} = \gamma\mathbf{x} \tag{20}$$

The scalar $\beta \in \mathbb{R}$ controls the trade-off between the total vertex and spectral spreads. When $|\beta|$ is large, the eigenvectors of $\tilde{\mathbf{L}}^{BE}(\beta)$ reveal structure which is similar to the eigenvectors of $\mathbf{L}^{BE}$, while small values of $|\beta|$ produce structure which is similar to those corresponding to $\mathbf{W}^{BE}$.

The analysis above yields the generalized graph Laplacian in Eq.(18) whose solutions can be used for graph embedding. One could use different scaling coefficients for the two matrix coefficients, thus giving different weight to the Laplacian and affinity terms in the resulting embedding. The embedding (coined GSSE), obtained by solving Eq.(19), is composed of solutions to the generalized eigenvalue problem that characterize the relationships between these two quantities, specifically the lower bound of their feasibility domain.

A.1 Node Embeddings using Total Vertex and Spectral Spreads (GSSE)

We present an additional embedding method which is based on the analysis in section 3.2 (Structural and local similarity trade-off Via Total Vertex and Spectral Spreads) which derived the generalized Laplacian $\tilde{\mathbf{L}}$ in Eq.(8). In order to employ $\tilde{\mathbf{L}}$ for analysis and practical considerations it is often useful to encode the network captured in $\tilde{\mathbf{L}}$ using a semi-positive definite operator. For practical consideration, we transform $\tilde{\mathbf{L}}$ into a semi-positive definite matrix $\mathbf{L}_\triangle$ using a simple perturbation matrix Δ, where $\Delta = \mu\mathbf{I}$, $\mu = -\tilde{\lambda}_0$, where $\tilde{\lambda}_0$ is the smallest eigenvalue of $\tilde{\mathbf{L}}$, and $\mathbf{I}$ is the identity matrix. Setting

$$\mathbf{L}_\triangle = \tilde{\mathbf{L}} + \Delta \tag{21}$$

ensures that $\mathbf{L}_\triangle$ is a semi-positive definite (SPD) matrix. Letting $\left\{\tilde{\lambda}_i\right\}_{i=1}^N$ and $\left\{\lambda_i^\triangle\right\}_{i=1}^N$ be the eigenvalues corresponding to $\tilde{\mathbf{L}}$ and $\mathbf{L}_\triangle$, respectively, we can see that the choice made in (21) ensures that the original spacing in eigenvalues of $\tilde{\mathbf{L}}$ is preserved in $\mathbf{L}_\triangle$ $\lambda_{i+1}^\triangle - \lambda_i^\triangle = \tilde{\lambda}_{i+1} - \tilde{\lambda}_i$. The embedding method proposed using $\mathbf{L}_\triangle$ is coined Graph Spectral Spread Embedding (GSSE). Note that there is a geometric interpretation that is related to the way $\tilde{\mathbf{L}}(\beta)$ was obtained from the lower boundary curve enclosing the feasibility domain $\mathbb{D}_{(s_\mathbf{L}, s_\mathbf{W})}$; Setting instead another perturbation matrix $\Delta = \mu\mathbf{I}$ with $\mu = -\lambda_N^W$ and defining

$$q(\beta) = \min(\lambda(\mathbf{L}_\triangle(\beta))) \tag{22}$$

where $\min(\lambda(\mathbf{L}_\triangle(\beta)))$ corresponds to the minimum eigenvalue of $\mathbf{L}_\triangle$. we have that

$$\mathbf{g}_\mathbf{W}(\mathbf{x}) - \beta\mathbf{g}_\mathbf{L}(\mathbf{x}) \geq q(\beta) \tag{23}$$

which defines a half plane in $\mathbb{D}_{(s_\mathbf{L}, s_\mathbf{W})}$.

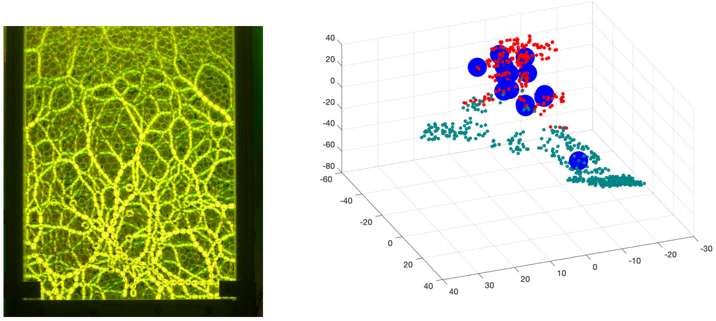
Properties: The feasibility domain $\mathbb{D}_{(s_{\mathbf{L}^{BE}}, s_{\mathbf{W}^{BE}})}$ is a bounded set since $\forall(s_{\mathbf{L}^{BE}}, s_{\mathbf{W}^{BE}}) \in \mathbb{D}_{(s_{\mathbf{L}^{BE}}, s_{\mathbf{W}^{BE}})}$

$$0 \leq s_{\mathbf{L}^{BE}} \leq \lambda_N^{BE} \text{ and } -\lambda_N^{W^{BE}} \leq s_{\mathbf{W}^{BE}} \leq \lambda_N^{W^{BE}} \tag{24}$$

where $\lambda_N^{W^{BE}}$corresponds to the largest eigenvalue of the affinity graph.

Table 3: Additional validation on the granular material datasets represented by irregular networks: we report p values, testing the statistical significance of our model for detecting failed edges. The p values are computed using the hypergeometric distribution.

Method /Network	Mean deg. (2.4)	Mean deg. (2.6)	Mean deg. (3.35)	Mean deg. (2.55)	Mean deg. (3)	Mean deg. (3.6)
node2vec	$1.9 \cdot 10^{-1}$ 10^{-1}	$3 \cdot 10^{-2}$	$3.4 \cdot 10^{-2}$	$2 \cdot 10^{-1}$	$6.3 \cdot 10^{-1}$	$5 \cdot 10^{-2}$
NetMF	$8 \cdot 10^{-1}$	$8 \cdot 10^{-1}$	$7.3 \cdot 10^{-3}$	$5 \cdot 10^{-1}$	$2.2 \cdot 10^{-9}$	$2.4 \cdot 10^{-9}$
LE	$7.4 \cdot 10^{-1}$	$8.3 \cdot 10^{-3}$	$2.5 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	$5 \cdot 10^{-3}$	$2 \cdot 10^{-2}$
$\mathbf{L}$ desc.	$4 \cdot 10^{-1}$	$7 \cdot 10^{-1}$	$2.1 \cdot 10^{-2}$	$5 \cdot 10^{-1}$	$6.5 \cdot 10^{-1}$	$7 \cdot 10^{-1}$
$\mathbf{W}^{BE}$ desc.	$3.4 \cdot 10^{-1}$	$6.1 \cdot 10^{-1}$	$3.8 \cdot 10^{-3}$	$9.9 \cdot 10^{-3}$	$0.1 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$
St.($\mathbf{W}^{BE}, \mathbf{L}^{BE}$)	$2.4 \cdot 10^{-3}$	$7.1 \cdot 10^{-4}$	$3 \cdot 10^{-2}$	$3 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$
GSE (Wave kernel)	$2 \cdot 10^{-4}$	$2.6 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$4 \cdot 10^{-5}$	$1.6 \cdot 10^{-3}$	$2.9 \cdot 10^{-2}$
GSE (SGW)	$1.7 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$4.5 \cdot 10^{-2}$	$3 \cdot 10^{-2}$	$7 \cdot 10^{-2}$



(a)

(b)

Figure 5: Graph embeddings used for forecasting failed edges using t-SNE embedding (a) Contact network (yellow), which is extracted from the force chains recorded in a 2D assembly of frictional photoelastic disks overlaid on the reconstructed “pseudo-image" [7]. (b) shows t-SNE visualization of edge embedding using the proposed Sylvester embedding. Points with green color correspond to edges whose value is below the mean, and points with red color correspond to edges whose value is above the mean. The blue enlarged dots correspond to the failed edges in the system which were successfully detected by each method.

A.2 Forecasting failures edges: additional details and comparisons

We report additional experimental results on the granular material datasets [7]. In all experiments, we used $c_t = 1$, for all t, c_t corresponds to the coefficients in the wave kernel signature WKS. We used a fixed number of 800 scales t_s and a fixed and number of singular eigenvectors corresponding to the number of points. The experiments reported in Table 3) test the statistical significance of each method by computing the p values using a hypergeometric probability distribution. Specifically, the total number of edges corresponds to the total population size parameter in the hypergeometric distribution with the feature that contains K failed edges, the size of the cluster is the number of draws and the number of observed successes corresponds to the number of edges correctly classified as failed edges.

B Network Alignment

We report additional results and details on the problem of network alignment using the STRING network from the STRING database. In the experimental results, we used all 332 human proteins known to interact with the Sars-Cov-2 as the available node correspondence between the two networks we tested on network alignment. From the 18,886 nodes of the String network we extract the 1000 nodes which correspond to the nodes with the largest diffusion scores and the 332 nodes corresponding to the human proteins known to interact with the Sars-Cov-2, based on the method suggested in [27]. Alignment is performed between two copies of the STRING network, where 10% and 20% edges were removed from the sub-network consisting of a total of 1332 nodes. It is evident that graph embedding methods such as LLE and LE, which are

Table 4: Network Alignment using the STRING network: percentage of nodes whose nearest neighbor corresponds to its true node correspondence using GSE compared to Graph Embedding methods (using all 332 available correspondence).

Method/noisy edges percentage	10%	20%
LE [4]	5.38%	-
LLE [34]	1.5%	1.3%
RL [46]	1%	-
HOPE [32]	2%	-
Isorank [36]	41%	40 %
Final [44]	58.2 %	56.6%
iNEAT [45]	63.8%	56.1%
GSSE	48 %	20.21 %
GSE	**76.4**%	**60**%

rooted in manifold learning that is biased to local similarity and heavily relies on graph smoothness are not effective for this task.

C The Sylvester operator

The discrete-time Sylvester operator $\mathbf{S}(\mathbf{X}) = \mathbf{AXB} - \mathbf{X}$ is used to express the eigenvalues and eigenvectors of $\mathbf{A}$ and $\mathbf{B}$ using a single operator $\mathbf{S}$, where we solve

$$\mathbf{S}(\mathbf{X}) = \mathbf{C} \tag{25}$$

using $\mathbf{C} = \mathbf{I}$, ($\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix), $\mathbf{A} = \mathbf{W}^{BE}, \mathbf{B} = \mathbf{L}^{BE}$.

Proposition 2 *Suppose that* $\mathbf{P} \in \mathbb{R}^{N\times N}$ *is a permutation matrix,* $\tilde{\mathbf{A}} = \mathbf{P}^T\mathbf{AP}$*, and* $\tilde{\mathbf{B}} = \mathbf{P}^T\mathbf{BP}$*, and assume that* $\mathbf{X}, \tilde{\mathbf{X}}$ *are two solutions to the discrete-time Sylvester equation Eq.(9) with the associated matrices* $\mathbf{A}, \mathbf{B}$ *and* $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$*, respectively, and* $\lambda_i^{\mathbf{L}^{BE}}\lambda_j^{\mathbf{W}^{BE}} \neq 1$, $\forall i, j$*, then*

$$\tilde{\mathbf{X}} = \mathbf{P}^T\mathbf{XP}. \tag{26}$$

Proof: Given a permutation matrix $\mathbf{P} \in \mathbb{R}^{N\times N}$, let $\tilde{\mathbf{A}} = \mathbf{P}^T\mathbf{AP}$, and $\tilde{\mathbf{B}} = \mathbf{P}^T\mathbf{BP}$. Assume that $\mathbf{X}, \tilde{\mathbf{X}}$ are two solutions to the discrete-time Sylvester equation Eq.(25) with the associate matrices $\mathbf{A}, \mathbf{B}$ and $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$, respectively, $\mathbf{AXB} - \mathbf{X} = \mathbf{I}$ and $\tilde{\mathbf{A}}\tilde{\mathbf{X}}\tilde{\mathbf{B}} - \tilde{\mathbf{X}} = \mathbf{I}$. Next, note that $\mathbf{P}^T\mathbf{AP}\tilde{\mathbf{X}}\mathbf{P}^T\mathbf{BP} - \tilde{\mathbf{X}} = \mathbf{I}$. Since $\mathbf{P}$ is a permutation matrix, we have that $\mathbf{P}^{-1} = \mathbf{P}^T$ and by multiplying $\mathbf{P}$ and $\mathbf{P}^T$ from the left-hand and right-hand sides, we obtain $\mathbf{AP}\tilde{\mathbf{X}}\mathbf{P}^T\mathbf{B} - \tilde{\mathbf{X}} = \mathbf{I}$. Denoting $\mathbf{Y} = \mathbf{P}\tilde{\mathbf{X}}\mathbf{P}^T$, since $\lambda_i^{\mathbf{L}^{BE}}\lambda_j^{\mathbf{W}^{BE}} \neq 1$, $\forall i, j$, then $\mathbf{X}$ is a unique solution, hence we obtain that $\mathbf{X} = \mathbf{Y} = \mathbf{P}\tilde{\mathbf{X}}\mathbf{P}^T$.□

The resulting solution of $\mathbf{X}$ is described in the next Lemma below. Note that since $\mathbf{A} = \mathbf{W}^{BE}, \mathbf{B} = \mathbf{L}^{BE}$ we have $\mathbf{A}$ and $\mathbf{B}$ that are diagonalizable, (with the matrices $\Phi^{\mathbf{W}^{BE}}, \Phi^{\mathbf{L}^{BE}}$ corresponding to the associated eigenvectors of $\mathbf{W}^{BE}, \mathbf{L}^{BE}$, respectively).

Lemma 1 *Let* $\mathbf{A}, \mathbf{B}$*, and* $\mathbf{C} \in \mathbb{R}^{N\times N}$ *in the Sylvester equation (25), with* $\mathbf{A} = \mathbf{W}^{BE}, \mathbf{B} = \mathbf{L}^{BE}$*. Let* $\{u_l\}_{l=1}^N$, $\{\lambda_l\}_{l=1}^N$*, and* $\{v_l\}_{l=1}^N$, $\{\mu_l\}_{l=1}^N$ *be the corresponding eigenvectors and eigenvalues of* $\mathbf{A}$ *and* $\mathbf{B}$*, respectively. Assume that* $\lambda_i^{\mathbf{W}^{BE}}\lambda_j^{\mathbf{L}^{BE}} \neq 1\, \forall i, j$*, and that the graph representation encoded in* $\mathbf{A}$ *and* $\mathbf{B}$ *have the same order based on the identity map,* $\pi : \mathbf{A} \to \mathbf{B}$ *with* $\pi(\mathbf{A}(:, i)) = \mathbf{B}(:, i)$ *for each* $i \in \mathcal{V}$*. Then, if* $\mathbf{C} = \mathbf{I}$*, the solution* $\mathbf{X}$ *to the discrete-time Sylvester system (25) is*

$$\mathbf{X} = \Phi^{\mathbf{W}^{BE}}\tilde{\mathbf{C}}(\Phi^{\mathbf{L}^{BE}})^T \tag{27}$$

where $\tilde{\mathbf{C}}_{i,j} = \frac{(\phi_i^{\mathbf{W}^{BE}})^T\phi_j^{\mathbf{L}^{BE}}}{\lambda_i^{\mathbf{W}^{BE}}\lambda_j^{\mathbf{L}^{BE}}-1}$

Proof: Using $\mathbf{A} = \mathbf{W}^{BE}, \mathbf{B} = \mathbf{L}^{BE}$ we have that $\mathbf{A}$ and $\mathbf{B}$ are diagonalizable, $\mathbf{A} = \Phi^{\mathbf{W}^{BE}}\Lambda(\Phi^{\mathbf{W}^{BE}})^{-1}$ and $\mathbf{B} = \Phi^{\mathbf{L}^{BE}}\Lambda(\Phi^{\mathbf{W}^{BE}})^{-1}$ where $\Lambda^{\mathbf{W}^{BE}} = \text{diag}(\lambda_1^{\mathbf{W}^{BE}}, ..., \lambda_N^{\mathbf{W}^{BE}})$ and $\Lambda^{\mathbf{L}^{BE}} = \text{diag}(\lambda_1^{\mathbf{L}^{BE}}, ..., \lambda_N^{\mathbf{L}^{BE}})$. Since the matrices $\Phi^{\mathbf{W}^{BE}}, \Phi^{\mathbf{L}^{BE}}$ are orthogonal, we have

$$\Phi^{\mathbf{W}^{BE}}\Lambda^{\mathbf{W}^{BE}}(\Phi^{\mathbf{W}^{BE}})^T\mathbf{X}\Phi^{\mathbf{L}^{BE}}\Lambda^{\mathbf{L}^{BE}}(\Phi^{\mathbf{L}^{BE}})^T - \mathbf{X} = \mathbf{C} \tag{28}$$

Multiplying by $(\Phi^{\mathbf{W}^{BE}})^T$ and $\Phi^{\mathbf{L}^{BE}}$ we obtain

$$\Lambda^{\mathbf{W}^{BE}}(\Phi^{\mathbf{W}^{BE}})^T\mathbf{X}\Phi^{\mathbf{L}^{BE}}\Lambda^{\mathbf{L}^{BE}} - (\Phi^{\mathbf{W}^{BE}})^T\mathbf{X}\Phi^{\mathbf{L}^{BE}} = (\Phi^{\mathbf{W}^{BE}})^T\mathbf{C}(\Phi^{\mathbf{L}^{BE}}) \tag{29}$$

Setting $\tilde{\mathbf{C}} = (\Phi^{\mathbf{W}^{BE}})^T\mathbf{X}\Phi^{\mathbf{L}^{BE}}$ and using $\mathbf{C} = \mathbf{I}$, we have

$$\Lambda^{\mathbf{W}^{BE}}\tilde{\mathbf{C}}\Lambda^{\mathbf{L}^{BE}} - \tilde{\mathbf{C}} = (\Phi^{\mathbf{W}^{BE}})^t(\Phi^{\mathbf{L}^{BE}}) \tag{30}$$

Since $\Lambda^{\mathbf{W}^{BE}}$ is a diagonal matrix, we have that the ith row of $\Lambda^{\mathbf{W}^{BE}}\tilde{\mathbf{C}}\Lambda^{\mathbf{L}^{BE}}$ is λ_i times the ith row of $\tilde{\mathbf{C}}\Lambda^{\mathbf{L}^{BE}}$and since $\Lambda^{\mathbf{L}^{BE}}$ is a diagonal matrix, then the ith column of $\Lambda^{\mathbf{W}^{BE}}\tilde{\mathbf{C}}\Lambda^{\mathbf{L}^{BE}}$ is $\lambda_i^{\mathbf{L}^{BE}}$ times the ith column of $\Lambda^{\mathbf{W}^{BE}}\tilde{\mathbf{C}}$. Combining the two properties, we obtain that for the (i,j) entry

$$\lambda_j^{\mathbf{L}^{BE}}\lambda_i^{\mathbf{W}^{BE}}\tilde{c}_{i,j} - \tilde{c}_{i,j} = (\phi_i^{\mathbf{W}^{BE}})^T\phi_j^{\mathbf{L}^{BE}}$$

hence: $\tilde{\mathbf{C}}_{i,j} = \frac{(\phi_i^{\mathbf{W}^{BE}})^T\phi_i^{\mathbf{L}^{BE}}}{\lambda_i^{\mathbf{W}^{BE}}\lambda_j^{\mathbf{L}^{BE}}-1}$, with $\mathbf{X} = \Phi^{\mathbf{W}^{BE}}\tilde{\mathbf{C}}(\Phi^{\mathbf{L}^{BE}})^T$.$\square$