Graph Coarsening with Preserved Spectral Properties

Yu Jin
University of Maryland

Andreas Loukas
EPFL

Joseph F. JaJa
University of Maryland

Abstract

In graph coarsening, one aims to produce a coarse graph of reduced size while preserving important graph properties. However, as there is no consensus on which specific graph properties should be preserved by coarse graphs, measuring the differences between original and coarse graphs remains a key challenge. This work relies on spectral graph theory to justify a distance function constructed to measure the similarity between original and coarse graphs. We show that the proposed spectral distance captures the structural differences in the graph coarsening process. We also propose graph coarsening algorithms that aim to minimize the spectral distance. Experiments show that the proposed algorithms can outperform previous graph coarsening methods in graph classification and stochastic block recovery tasks.

1 INTRODUCTION

Graphs are widely used to represent object relationships in real-world applications. As many applications involve large-scale graphs with complex structures, it is generally hard to explore and analyze the key properties directly from large graphs. Hence the graph coarsening techniques have been commonly used to facilitate the process (Liu et al., 2016; Chevalier and Safro, 2009).

Generally speaking, the aim of any graph reduction scheme is to reduce the number of nodes and edges of a graph, while also ensuring that the “essential properties” of the original graph are preserved. The question of what these properties should be preserved remains inconclusive, but there is significant evidence that they should relate to the spectrum of a graph operator, such as the adjacency or normalized Laplacian matrix (Spielman and Teng, 2011; Hermsdorff and Gunderson, 2019). A long list of theorems in spectral graph theory shows that the combinatorial properties of a graph are aptly captured by its spectrum. As such, graphs with the similar spectrum are generally regarded to share similar global and local structure (Van Dam and Haemers, 2003; Banerjee, 2008). Based on this realization, modern graph sparsification techniques (Spielman and Srivastava, 2011; Jovanović and Stanić, 2012; Batson et al., 2013) have moved on from previously considered objectives, such as cut and shortest-path distance preservation, and now aim to find sparse spectrally similar graphs.

In contrast to graph sparsification, in coarsening there has been little progress towards attaining spectrum preservation guarantees. The foremost roadblock seems to lie in defining what spectral similarity should entail for graphs of different sizes. The original and coarse graphs now have different numbers of eigenvalues and eigenvectors, which prohibit a direct comparison. To circumvent this issue, recent works have considered restricting the guarantees to a subset of the spectrum (Loukas and Vandergheynst, 2018; Loukas, 2019). However, focusing only on a subset of eigenvalues and eigenvectors also means that important information of the graph spectrum is ignored.

In this work, we start by reconsidering the fundamental spectral distance metric (Jovanović and Stanić, 2012; Gu et al., 2015; Jovanovic, 2015; Jovanović and Stanić, 2014), which compares two graphs by means of a norm of their eigenvalue differences. This metric is seemingly inappropriate as it necessitates that two graphs have the same number of eigenvalues. However, we find that in the context of coarsening, this difficulty can be circumvented by substituting the coarse graph with its lifted counterpart: the latter contains the same information as the former while also having the correct number of eigenvalues. Our analysis shows that the proposed distance naturally captures the graph changes in the graph coarsening process. In particular, when the graph coarsening merges nodes

Proceedings of the 23^{rd}International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

that have similar connections to the rest of the graph, the spectral distance is provably small. By merging similarly connected nodes, nodes and edges in coarse graphs are able to represent the connectivity patterns of the original graphs, thus preserving structural and connectivity information.

Our contributions are summarized as follows:

- We show how the spectral distance (Jovanović and Stanić, 2012; Gu et al., 2015; Jovanovic, 2015; Jovanović and Stanić, 2014), though originally restricted to graphs of the same size, can be utilized to measure how similar a graph is with its coarsened counterpart.
- We examine how the new spectral distance captures graph structural changes occurring during the graph coarsening process.
- We present two coarsening algorithms that provably minimize the spectral distance.
- We experimentally show that the proposed methods outperform other graph coarsening algorithms on two graph related tasks.

All proofs can be found in the supplementary material.

2 RELATED WORK

Recent works have proposed to coarsen graphs by preserving the spectral properties of the matrix representations of graphs (Loukas and Vandergheynst, 2018; Loukas, 2019; Durfee et al., 2019; Purohit et al., 2014; Hermsdorff and Gunderson, 2019). For example, Loukas (2019) proposed to preserve the action of the graph Laplacian with respect to an (eigen)-space of fixed dimension, arguing that this suffices to capture the global properties of graph relevant to partitioning and spectral clustering. Durfee et al. (2019) proposed to preserve the all-pairs effective resistance. Garg and Jaakkola (2019) defined a cost based on the theory of optimal transport. Saket et al. suggested a Minimum Description Length (MDL) principle relevant to unweighted graphs (Navlakha et al., 2008). Most of these distance functions are specific to particular applications; the question of how to define an application-independent graph coarsening framework remains a challenge.

There is a sizable literature dealing with the characterization of graphs in terms of their spectral properties (Jovanović and Stanić, 2012; Tsitsulin et al., 2018; Dong and Bindel, 2019). Previous work defined distance functions based on Laplacian eigenvalues which measure differences between graphs (Jovanović and Stanić, 2012; Gu et al., 2015). Spielman and Teng introduced a notion of spectral similarity for two graphs in their graph sparsification framework (Batson et al., 2013; Spielman and Teng, 2011). Recently, Tsitsulin et al. proposed an efficient graph feature extractor, based on Laplacian spectrum, for comparisons of large graphs (Tsitsulin et al., 2018). Dong uses spectral densities to visualize and estimate meaningful information about graph structures (Dong and Bindel, 2019). Nevertheless, despite the popularity of spectral methods, the graph spectrum remains little explored in the context of graph coarsening.

3 PRELIMINARIES

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ be a graph, with $\mathcal{V}$ a set of $N = |\mathcal{V}|$ nodes, $\mathcal{E}$ a set of $M = |\mathcal{E}|$ edges, and $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ the weighted adjacency matrix. We denote the node v_i the node by $\boldsymbol{w}(i) \in \mathbb{R}^N$ representing the vector of the weights of the edges incident on v_i and by $d(i) = \sum_{j=1}^{N} \boldsymbol{W}(i,j)$ the node degree of v_i. The graphs considered in this work are weighted, undirected, and possess no isolated nodes (i.e. $d(i) > 0$ for all v_i).

The combinatorial and normalized Laplacians of $\mathcal{G}$ are defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \quad \text{and} \quad \boldsymbol{\mathcal{L}} = \boldsymbol{I}_N - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}, \tag{1}$$

respectively, where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix and $\boldsymbol{D}$ is the diagonal degree matrix with $\boldsymbol{D}(i,i) = d(i)$.

3.1 Graph Coarsening

The coarse graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c, \boldsymbol{W}_c)$ with $n = |\mathcal{V}_c|$ is obtained from the original graph $\mathcal{G}$ by first selecting a set of non-overlapping graph partitions $\mathcal{P} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\} \subset \mathcal{V}$, which cover all the nodes in $\mathcal{V}$. Each partition $\mathcal{S}_p$ corresponds to a "super-node" denoted by s_p and the "super-edge" connecting the super-nodes $\boldsymbol{W}_c(p,q)$ has weight equal to the accumulative edge weights between nodes in the corresponding graph partitions $\mathcal{S}_p$ and $\mathcal{S}_q$:

$$\boldsymbol{W}_c(p,q) = w(\mathcal{S}_p, \mathcal{S}_q) := \sum_{v_i \in \mathcal{S}_p, v_j \in \mathcal{S}_q} \boldsymbol{W}(i,j) \tag{2}$$

Let $\boldsymbol{P} \in \mathbb{R}^{n \times N}$ be the matrix whose columns are partition indicator vectors:

$$\boldsymbol{P}(p,i) = \begin{cases} 1, & \text{if } v_i \in \mathcal{S}_p \\ 0, & \text{othewise.} \end{cases}$$

It is then well known that the weight matrix $\boldsymbol{W}_c$ of the coarse graph $\mathcal{G}_c$ satisfies

$$\boldsymbol{W}_c = \boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^\top.$$

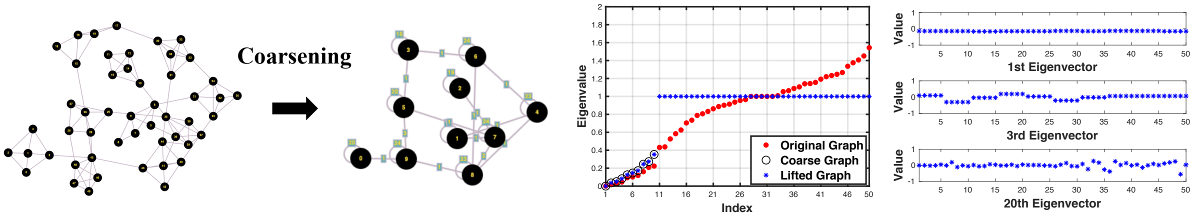

Figure 1: **Left**: an example illustrating the graph coarsening process. The original graph is a random graph sampled from stochastic block model with 50-node and 10 predefined blocks. The coarse graph is coarsened from the predefined partitions. **Right**: Eigenvalues and eigenvectors of normalized Laplacian matrices of original, coarse and lifted graphs. The eigenvalues of coarse graphs align with the eigenvalues of original graphs and the eigenvectors indicate the block membership information.

The definition of the coarsened Laplacian matrices follows directly:

$$\boldsymbol{L}_c = \boldsymbol{D}_c - \boldsymbol{W}_c \quad \text{and} \quad \boldsymbol{\mathcal{L}}_c = \boldsymbol{I}_n - \boldsymbol{D}_c^{-1/2}\boldsymbol{W}_c\boldsymbol{D}_c^{-1/2}.$$

Similarly to the adjacency matrix, the combinatorial Laplacian of the coarse graph can be obtained by the formula $\boldsymbol{L}_c = \boldsymbol{P}\boldsymbol{L}\boldsymbol{P}^\top$. The same however doesn't hold for the normalized Laplacian, as in general $\boldsymbol{P}\boldsymbol{\mathcal{L}}\boldsymbol{P}^\top \neq \boldsymbol{\mathcal{L}}_c$.

3.2 Graph Lifting

We define $\mathcal{G}_l = (\mathcal{V}, \mathcal{E}_l, \boldsymbol{W}_l)$ to be the graph lifted from the coarse graph $\mathcal{G}_c$ with respect to a set of non-overlapping partitions $\mathcal{P}$. In graph lifting, each node s_p of the coarse graph is lifted to $|\mathcal{S}_p|$ nodes and nodes in the lifted graph are connected by edges whose weight is equal to the coarse edge weight normalized by the sizes of partitions. Specifically, for any $v_i \in \mathcal{S}_p$ and $v_j \in \mathcal{S}_q$ we have:

$$\begin{aligned}\boldsymbol{W}_l(i,j) = \frac{w(\mathcal{S}_p, \mathcal{S}_q)}{|\mathcal{S}_p||\mathcal{S}_q|} &= \frac{\sum_{v'_i \in \mathcal{S}_p, v'_j \in \mathcal{S}_q} \boldsymbol{W}(i',j')}{|S_p||S_q|} \\ &= \frac{\boldsymbol{W}_c(p,q)}{|S_p||S_q|}. \end{aligned} \tag{3}$$

When $\mathcal{S}_p = \mathcal{S}_q = \mathcal{S}$, the weight $\boldsymbol{W}_l(i,j)$ can be seen to be equal to the weight of all edges in the subgraph induced by $\mathcal{S}$, after normalization by $|\mathcal{S}|^2$. It easily follows that if $\boldsymbol{W}(i,j)$ is the same for every $v_i, v_j \in \mathcal{S}$, then also $\boldsymbol{W}_l(i,j) = \boldsymbol{W}(i,j)$, i.e., in-partition weights are exactly preserved by successive coarsening and lifting in this case.

The above combinatorial definition can be expressed in an algebraic form in terms of the the pseudo-inverse $\boldsymbol{P}^+$ of $\boldsymbol{P}$, (i.e., $\boldsymbol{P}\boldsymbol{P}^+ = \boldsymbol{I}$), whose elements are:

$$\boldsymbol{P}^+(j,p) = \begin{cases} \frac{1}{|\mathcal{S}_p|} & \text{if } v_j \in \mathcal{S}_p \\ 0 & \text{otherwise.} \end{cases}$$

With this in place, the adjacency matrices of the lifted and coarse graphs are connected by the following relations:

$$\boldsymbol{W}_l = \boldsymbol{P}^+\boldsymbol{W}_c\boldsymbol{P}^\mp \quad \text{and} \quad \boldsymbol{W}_c = \boldsymbol{P}\boldsymbol{W}_l\boldsymbol{P}^\top.$$

The following equation reveals that lifting preserves the connectivity up to a projection onto the partitions:

$$\begin{aligned}\boldsymbol{W}_l = \boldsymbol{P}^+\boldsymbol{W}_c\boldsymbol{P}^\mp &= \boldsymbol{P}^+\boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^\top\boldsymbol{P}^\mp \\ &= \boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}^\top = \boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}, \end{aligned} \tag{4}$$

where $\boldsymbol{\Pi} = \boldsymbol{P}^+\boldsymbol{P}$ is a projection matrix, with $\boldsymbol{\Pi}\boldsymbol{\Pi} = \boldsymbol{P}^+\boldsymbol{P}\boldsymbol{P}^+\boldsymbol{P} = \boldsymbol{P}^+\boldsymbol{P} = \boldsymbol{\Pi}$.

The lifted Laplacian matrices are given by

$$\boldsymbol{L}_l = \boldsymbol{P}^+\boldsymbol{L}_c\boldsymbol{P}^\mp \quad \text{and} \quad \boldsymbol{\mathcal{L}}_l = \boldsymbol{C}^\top\boldsymbol{\mathcal{L}}_c\boldsymbol{C}, \tag{5}$$

where $\boldsymbol{C} \in \mathbb{R}^{n \times N}$ is the *normalized coarsening matrix* whose entries are given by:

$$\boldsymbol{C}(p,i) = \begin{cases} \frac{1}{\sqrt{|\mathcal{S}_p|}} & \text{if } v_i \in \mathcal{S}_p \\ 0 & \text{otherwise,} \end{cases}$$

such that $\boldsymbol{C}^\top = \boldsymbol{C}^+$ and $\boldsymbol{C}^\top\boldsymbol{C} = \boldsymbol{P}^\mp\boldsymbol{P} = \boldsymbol{\Pi}$. In this manner, we have

$$\boldsymbol{L}_c = \boldsymbol{P}\boldsymbol{L}_l\boldsymbol{P}^\top \quad \text{and} \quad \boldsymbol{\mathcal{L}}_c = \boldsymbol{C}\boldsymbol{\mathcal{L}}_l\boldsymbol{C}^\top. \tag{6}$$

For a more in-depth discussion of the mathematics of graph coarsening and graph lifting, we refer the interested reader to (Loukas, 2019).

4 SPECTRAL DISTANCE

We start by briefly reviewing some basic facts about the spectrum associated with the Laplacian matrix of a coarse graph. We then demonstrate how to exploit these properties in order to render the classical spectral distance metric amenable to (coarse) graphs of different sizes.

4.1 Properties of the Coarse Laplacian Spectrum

Denote the eigenvalues and eigenvectors of the normalized Laplacian matrices as $\boldsymbol{\lambda}$ and $\boldsymbol{u}$, respectively, with $\boldsymbol{\mathcal{L}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ where the i-th column of $\boldsymbol{U}$ corresponds to $\boldsymbol{u}_i$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$. The eigenvalues are ordered in non-decreasing order.

Property 4.1 (**Interlacing**. Section 5.3 in (Butler, 2008))**.** *The normalized Laplacian eigenvalues of the original and coarsened graphs satisfy*

$$\boldsymbol{\lambda}(i) \leq \boldsymbol{\lambda}_c(i) \leq \boldsymbol{\lambda}(i+N-n) \quad \textit{for all} \quad i = 1, \dots, n.$$

Property 4.1 is a general interlacing inequality that captures pairwise difference between the eigenvalues of the original and coarse graph Laplacians (Chung, 1997; Butler, 2007). Since it holds for any graph and coarsening, the inequality will, in some cases, be loose.

Property 4.2 (**Eigenvalue Preservation**)**.** *The normalized Laplacian eigenvalues of the lifted graph contain all eigenvalues of the coarse graph and additional eigenvalues* 1 *with* $(N-n)$ *multiplicity.*

Property 4.3 (**Eigenvector Preservation**)**.** *The eigenvectors of the coarse graph lifted by* $\boldsymbol{C}$*, i.e.* $\boldsymbol{u}_l = \boldsymbol{C}\boldsymbol{u}_c$ *are the eigenvectors of* $\boldsymbol{\mathcal{L}}_l$*.*

Property 4.2 and 4.3 state that the action of lifting preserves most spectral properties of the coarse graph. Thus, we may use the lifted graph as a proxy to define the distance function (Toivonen et al., 2011). Figure 1 shows an example illustrating the graph coarsening process as well as the effect on the graph spectrum.

4.2 Spectral Distance

In the following, we propose two notions of the spectral distance to quantify the difference between original and coarse graphs. We first use the lifted graph as the "proxy" of the coarse graph and define the *full spectral distance*:

Definition 4.4. *The full spectral distance between graph* $\mathcal{G}$ *and* $\mathcal{G}_c$ *is defined as follows:*

$$SD_{full}(\mathcal{G}, \mathcal{G}_c) = \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_l\|_1 = \sum_{i=1}^{N} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_l(i)|,$$

where vectors $\boldsymbol{\lambda}$ *and* $\boldsymbol{\lambda}_l$ *contain the eigenvalues of the original and lifted graphs.*

As the original and lifted graphs have the same number of nodes, we may directly use a norm to measure the pairwise differences between eigenvalues.

On the flip side, the definition requires computing all eigenvalues of original graphs regardless of the coarse graph size, which is computationally expensive, especially for large graphs. The limitation motivates us to define the *partial spectral distance* by selecting part of the terms in the full spectral distance definition.

Let k_1 and k_2 be defined as $k_1 = \arg\max_i\{i : \boldsymbol{\lambda}_c(i) < 1\}, k_2 = N - n + k_1$. We expand the full spectral distance into three terms as follows:

$$\begin{aligned} SD_{\text{full}}(\mathcal{G}, \mathcal{G}_c) &= \sum_{i=1}^{N} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_l(i)| \\ &= \sum_{i=1}^{k_1} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_l(i)| + \sum_{i=k_1+1}^{k_2} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_l(i)| \\ &\quad + \sum_{i=k_2+1}^{N} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_l(i)| \qquad (7) \\ &= \sum_{i=1}^{k_1} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_c(i)| + \sum_{i=k_1+1}^{k_2} |\boldsymbol{\lambda}(i) - 1| \\ &\quad + \sum_{i=k_2+1}^{N} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_c(i-N+n)| \qquad (8) \end{aligned}$$

The last equation is from the Property 4.2 where $\boldsymbol{\lambda}_l$ contains eigenvalues of the coarse graph as well as eigenvalue 1 with $N-n$ multiplicity. Eigenvalue $\boldsymbol{\lambda}_l$ satisfies:

$$\boldsymbol{\lambda}_l(i) = \begin{cases} \boldsymbol{\lambda}_c(i) & i \leq k_1 \\ 1 & k_1 + 1 \leq i \leq k_2 \\ \boldsymbol{\lambda}_c(i-N+n) & i > k_2 \end{cases}$$

With this in place, we define the *partial spectral distance* to be equal to the full spectral distance minus the $N-n$ terms for which $\boldsymbol{\lambda}_l = 1$:

Definition 4.5. *The partial spectral distance between graph* $\mathcal{G}$ *and* $\mathcal{G}_c$ *is defined as*

$$SD_{part}(\mathcal{G}, \mathcal{G}_c) = \sum_{i=1}^{k} |\boldsymbol{\lambda}(i) - \boldsymbol{\lambda}_c(i)| + \sum_{i=k+1}^{n} |\boldsymbol{\lambda}_c(i) - \boldsymbol{\lambda}(i+N-n)|,$$

where $k = \arg\max_i\{i : \boldsymbol{\lambda}_c(i) < 1\}$*.*

For the partial spectral distance, we only need to compute n rather than N eigenvalues of the normalized Laplacian of the original graph, which significantly reduces the computational cost when $n \ll N$.

The full and partial spectral distances are related by,

$$SD_{\text{full}}(\mathcal{G}, \mathcal{G}_c) = SD_{\text{part}}(\mathcal{G}, \mathcal{G}_c) + \sum_{i=k_1+1}^{k_2} |\boldsymbol{\lambda}(i) - 1|$$

The excluded terms $\sum_{i=k_1+1}^{k_2} |\boldsymbol{\lambda}(i) - 1|$ measure the closeness of the original Laplacian eigenvalues and eigenvalue 1. The two definitions are equivalent when the normalized Laplacian of the original graph $\boldsymbol{\mathcal{L}}_N$ contains eigenvalue 1 with $N-n$ multiplicity. The condition is equivalent to asserting that the adjacency matrix W is *singular* with $N-n$ algebraic multiplicity of the eigenvalue 0 (Sciriha, 2007; AL-Tarimshawy, 2018). We have observed empirically that, when coarsening nodes that have similar connections, the adjacency matrix has eigenvalues close to 0. In such situations, the terms of the full spectral distance that are excluded by the partial spectral distance are almost 0 and the partial distance closely approximates the full one.

Note that both definitions of spectral distance are proper distance metrics over the space of graph Laplacian eigenvalues. However, the spectral distance is not able to distinguish graphs with the same sets of Laplacian eigenvalues (referred to as cospectral graphs (Van Dam and Haemers, 2003)). Thus, there could exist multiple coarse graphs corresponding to the same spectral distance.

4.3 Relation to Graph Coarsening

To illustrate the connections between spectral distance and graph coarsening, we first consider the ideal case when merged nodes within have the same normalized edge weights:

Proposition 4.1. *Let the graph $\mathcal{G}_c$ be obtained by coarsening $\mathcal{G}$ with respect to a set of partitions $\mathcal{P} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\}$. If $\mathcal{P}$ is selected such that every node in a partition has the same normalized edge weights,*

$$\frac{\boldsymbol{w}(i)}{d(i)} = \frac{\boldsymbol{w}(j)}{d(j)} \quad \textit{for all } v_i, v_j \in \mathcal{S} \quad \textit{and} \quad \mathcal{S} \in \mathcal{P} \tag{9}$$

then

$$SD_{full}(\mathcal{G}, \mathcal{G}_c) = 0 \quad \textit{and} \quad SD_{part}(\mathcal{G}, \mathcal{G}_c) = 0.$$

Therefore, the ideal graph coarsening attains a minimal (full and partial) spectral distance.

We next provide a more general result on how the spectral distance can capture the structural changes in the graph coarsening framework. Consider the basic coarsening where the coarse graph is formed by merging one pair of nodes (i.e. $n = N-1$). In this setting, we prove the following:

Proposition 4.2. *Suppose the graph $\mathcal{G}_c$ is obtained from $\mathcal{G}$ by merging a pair of nodes $v(a)$ and $v(b)$. If the normalized edge weights of merged nodes satisfy*

$$\left\| \frac{\boldsymbol{w}(a)}{d(a)} - \frac{\boldsymbol{w}(b)}{d(b)} \right\|_1 \leq \epsilon,$$

then the spectral distance between the original and coarse graphs is bounded by

$$SD_{full}(\mathcal{G}, \mathcal{G}_c) \leq N\epsilon \textit{ and } SD_{part}(\mathcal{G}, \mathcal{G}_c) \leq n\epsilon.$$

Algorithm 1 Multilevel Graph Coarsening (**MGC**)

```
Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ and target size of the coarse graph $n$.
$s \leftarrow N$
while $s > n$ do
    for $v_i \in \mathcal{V}_s$ do
        for $v_j \in \mathcal{N}_i$ do
            $d_s(i,j) = \left\| \frac{\boldsymbol{w}(i)}{d(i)} - \frac{\boldsymbol{w}(j)}{d(j)} \right\|_1$
    $i_{\min}, j_{\min} = \arg\min_{i,j} d_s(i,j)$
    $s \leftarrow s - 1$
    Merge nodes $v_{i_{\min}}$ and $v_{j_{\min}}$ to form the coarse graph $\mathcal{G}_s$.
return $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n, \boldsymbol{W}_n)$
```

The above proposition states that the spectral distance is bounded by the discrepancy of normalized edge weights of merged nodes. The bound implies that minimizing the nodes' edge weights within the same partitions results in bounded spectral perturbations.

5 ALGORITHMS

We propose two graph coarsening algorithms to produce coarse graphs with minimal small spectral distance. The first follows from Proposition 4.2 in that the coarse graphs are formed by iteratively merging graph nodes with similar normalized edge weights. The second algorithm is inspired by spectral clustering: we leverage on the combinations of normalized Laplacian eigenvectors combined with k-means clustering to find the graph partitions and the corresponding coarse graphs. Though different, both algorithms are shown to generate coarse graphs of bounded spectral distance.

5.1 Multilevel Graph Coarsening

The Multilevel Graph Coarsening (**MGC**) algorithm iteratively merges pairs of nodes which share similar connections. During each iteration, **MGC** searches for the pair of nodes with the most similar normalized edge weights and merges them into super-nodes. To reduce the computational cost, we constraint the candidate pairs of graph nodes to be within 2-hop distance. We denote $\mathcal{N}_i$ as the set of nodes that are within 2-hops distance from node v_i. The pseudo-code of **MGC** is presented in Algorithm 1.

Analysis. The following corollary bounds the spectral distance of **MGC** algorithm:

Corollary 5.1. *Suppose the graph $\mathcal{G}_c$ is coarsened from $\mathcal{G}$ by iteratively merging pairs of nodes $v(a_s)$ and $v(b_s)$ for s from N to $n+1$, if the normalized edge weights of merged nodes satisfy,*

$$\left\| \frac{\boldsymbol{w}(a_s)}{d(a_s)} - \frac{\boldsymbol{w}(b_s)}{d(b_s)} \right\|_1 \leq \epsilon_s,$$

then the spectral distance between the original and coarse graphs is bounded by

$$SD_{full}(\mathcal{G}, \mathcal{G}_c) \leq N \sum_{s=N}^{n+1} \epsilon_s, \quad SD_{part}(\mathcal{G}, \mathcal{G}_c) \leq n \sum_{s=N}^{n+1} \epsilon_s$$

The bound is a direct corollary Proposition 4.2.

Time complexity. The time complexity of **MGC** is $O(M(N+n)(N-n))$, which is derived as follows: For each iteration, the computational cost of the 1-norm in line 6 is $O(s)$. Then the time complexity of the while loop in line 3 is $O(\sum_{s=n}^{N} s \cdot M) = O(M\frac{n+N}{2}(N-n)) = O(M(N+n)(N-n))$. When $n \approx N$, the complexity reduces to $O(MN)$. On the other hand, for $n \ll N$ the complexity becomes $O(MN^2)$.

5.2 Spectral Graph Coarsening

The spectral graph coarsening (**SGC**) algorithm identifies the coarsening partitions by attempting to minimize the k-means cost of rows of Laplacian eigenvectors. Different from traditional spectral clustering, we select eigenvectors with the eigenvalues corresponding to the head and tail eigenvalues as in the definition of partial spectral distance in Definition 4.5. The procedure is described in Algorithm 2. Notice that, since k_1 is unknown at the start, **SGC** algorithm iterates over different possible combinations of eigenvectors and selects the coarsening with minimum k-means cost.

Analysis. The following theorem relates the partial spectral distance with the k-means cost:

Theorem 5.2. *Let the coarse graph $\mathcal{G}_c$ be obtained from Algorithm 2 with graph partition $\mathcal{P}^*$, suppose that the graph coarsening is consistent, i.e., $\boldsymbol{\mathcal{L}}_c = \boldsymbol{C}\boldsymbol{\mathcal{L}}\boldsymbol{C}^\top$, and let the k-means cost satisfy $\mathcal{F}(\boldsymbol{U}, \mathcal{P}^*) < 1$. Then, the partial spectral distance is bounded by*

$$SD_{part}(\mathcal{G}, \mathcal{G}_c) \leq \frac{(n+2)\mathcal{F}(\boldsymbol{U}, \mathcal{P}^*) + 4\sqrt{\mathcal{F}(\boldsymbol{U}, \mathcal{P}^*)}}{1 - \mathcal{F}(\boldsymbol{U}, \mathcal{P}^*)}.$$

The theorem states that the spectral distance is bounded by the k-means clustering cost. Further, when the graph eigenvectors point to well-separated clusters and the k-means cost is small, the spectral distance is smaller. The main assumption posed is that $\boldsymbol{\mathcal{L}}_c = \boldsymbol{C}\boldsymbol{\mathcal{L}}\boldsymbol{C}^\top$, which may not hold for some graphs. For situations when this assumption is not met, the claim can be readily reworked to hold for the combinatorial Laplacian matrix for which the relation $\boldsymbol{L}_c = \boldsymbol{P}\boldsymbol{L}\boldsymbol{P}^\top$ always holds.

Algorithm 2 Spectral Graph Coarsening (**SGC**)

```
1: Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, eigenvectors $\boldsymbol{U}$ of the normalized Laplacian $\boldsymbol{\mathcal{L}}$, target size $n$.
2: if $\boldsymbol{\lambda}(N) \leq 1$ then
3:     Set $k_1 \leftarrow n$                    ▷ Spectral Clustering
4: else
5:     Set $k_1 \leftarrow \arg\min_k \{k : \boldsymbol{\lambda}(k) \leq 1, k \leq n, \boldsymbol{\lambda}(N - n + k + 1) > 1\}$    ▷ Iterative Spectral Coarsening
6: $k_2 \leftarrow N - n + k_1$.
7: while $k_1 \leq n$ do
8:     $\boldsymbol{U}_{k_1} \leftarrow [\boldsymbol{U}(1 : k_1); \boldsymbol{U}(k_2 + 1 : N)]$
9:     Apply $k$-means clustering algorithm on the rows of $\boldsymbol{U}_{k_1}$ to obtain graph partitions $\mathcal{P}^*_{k_1}$ that optimizes the following $k$-means cost:
           $$\mathcal{F}(\boldsymbol{U}_{k_1}, \mathcal{P}^*_{k_1}) = \sum_{i=1}^{N} \left( \boldsymbol{r}(i) - \sum_{j \in \mathcal{S}_i} \frac{\boldsymbol{r}(j)}{|\mathcal{S}_i|} \right)^2$$
       where $\boldsymbol{r}(i)$ is the $i^{\text{th}}$ row of $\boldsymbol{U}_{k_1}$.
10:    $k_1 \leftarrow k_1 + 1$, $k_2 = N - n + k_1$
11: return coarse graph $\mathcal{G}_c$ generated with respect to the partitions with minimum $k$-means clustering cost as
           $$\mathcal{P}^* = \arg\min_{k_1} \mathcal{F}(\boldsymbol{U}_{k_1}, \mathcal{P}^*_{k_1})$$
```

Time complexity. Excluding the one-time partial sparse eigenvalue decomposition that takes roughly $O(R(Mn + Nn^2))$ time using Lanczos iteration with R restarts and a graph of M edges (we need the smallest and largest n eigenvalues and eigenvectors) (Tremblay and Loukas, 2020), the time complexity of **SGC** is $O(KTNn^2)$, where K refers to the number of times the while loop is executed with $K \leq n$ and $O(TNn^2)$ is the complexity of the k-means clustering (whereas T bounds the number of k-means iterations).

6 EXPERIMENTS

We proceed to empirically evaluate the proposed graph coarsening algorithms on tasks involving real-world and synthetic graphs. Our first experiment considers the classification of coarsened graphs, whereas the second examines how well one may recover the block structure of graphs sampled from the stochastic block

Table 1: Classification accuracy on coarse graphs that are five times smaller.

Datasets	MUTAG	ENZYMES	NCI1	NCI109	PROTEINS	PTC
EM	78.90	18.92	62.81	61.35	63.72	48.56
LV	79.01	24.68	63.59	60.49	62.72	50.24
METIS	77.62	24.79	59.74	61.64	63.70	49.34
SC	80.37	24.40	63.14	62.57	64.08	50.16
SGC	80.34	29.19	63.94	**63.69**	64.70	**52.76**
MGC	**81.53**	**30.89**	**66.07**	63.55	**65.26**	52.28
Original	86.58	37.32	66.39	64.93	66.60	53.72

model. We show that our graph coarsening algorithms, which optimize the spectral distance, yield minimal classification accuracy degradation and can recover the block structures with high accuracy. Codes for both experiments are publicly available[1].

Baseline Algorithms We compare our methods with the following graph coarsening and partitioning algorithms as,

- **Edge Matching (EM).** The coarse graphs are formed by maximum-weight matching with the weight calculated as $\boldsymbol{W}(i,j)/\max\{d(i),d(j)\}$ (Dhillon et al., 2007).
- **Local Variation (LV).** Local variation methods coarsen a graph in a manner that approximately preserves a subset of its spectrum (Loukas, 2019). Here, we used the neighborhood-based variant and aimed to preserve the first $\max(10,n)$ eigenvectors and eigenvalues. Alternative choices for the preserved eigenspace may yield different results.
- **METIS.** This is a standard graph partitioning algorithm based on multi-level partitioning schemes that are widely used various domains, such as finite element methods and VLSI (Karypis and Kumar, 1998).
- **Spectral Clustering (SC).** Spectral clustering is a widely used graph clustering algorithm that finds densely connected graph partitions determined from the eigenvectors of the graph Laplacian (Von Luxburg, 2007). For a review of recent results on the fast approximation of SC, see (Tremblay and Loukas, 2020).

Note that to apply graph partitioning algorithms for coarsening purposes, we coarsen the graphs with respect to the graph partitions following the standard coarsening process as in equation 2.

6.1 Graph Classification with Coarse Graphs

Graph classification is a well studied graph machine learning problem, with a variety of applications to material design, drug discovery and computational neuroscience (Tsitsulin et al., 2018; Jin and JaJa, 2018; Xu et al., 2018; Park and Friston, 2013). However, some graph classifiers are not scalable for large graphs, such as those encountered in social network analysis and computational neuroscience (Park and Friston, 2013; Jin et al., 2015). Graph coarsening can reduce the graph sizes in the datasets, which provides acceleration on the training and inference of graph classification models. However, if the coarsening is not carefully done, it can also result in loss of useful information and, thus, of classification accuracy. In the following, we quantify the effect of different coarsening choices to graph classification. We utilize various graph coarsening methods to reduce the size of graphs in the datasets before passing them to the graph classifier. We then evaluate the quality of graph coarsening based on the classification accuracy drop (as compared to the same classifier on the original graphs).

Evaluation We coarsen the graph samples until $n = N/5$, i.e., until their number of nodes is reduced by a factor of five. The classification performance are evaluated based on 10-fold cross validation—in accordance to previous works (Tsitsulin et al., 2018; Dai et al., 2016; Xu et al., 2018).

Datasets. We use five standard graph classification datasets for graph classification evaluation (Shervashidze et al., 2011; Kersting et al., 2016; Jin and JaJa, 2018). Each dataset contains a set of variable-sized graphs stemming from a variety of applications. The graph statistics can be found in the supplementary materials.

The graph classifier. We use the Network Laplacian Spectral Descriptor (**NetLSD**) combined with a 1-NN classifier as the graph classification method (Tsitsulin et al., 2018). **NetLSD** was shown

[1] https://github.com/yuj-umd/spectral-coarsening

Table 2: Recovery Accuracy of Block Structures from Random Graphs in Stochastic Block Model

p, q	Type	EM	LV	METIS	SC	MGC	SGC
$0.2, 0.01$	Associative	0.1819	0.3076	0.7792	**0.7845**	0.3664	**0.7845**
	Disassortative	0.0956	0.1071	0.0815	0.0877	**0.1093**	0.0850
	Mixed	0.1052	0.1944	0.2389	0.3335	0.6062	**0.7107**
$0.5, 0.1$	Associative	0.1015	0.1902	0.7820	**0.7930**	0.2868	**0.7930**
	Disassortative	0.0854	0.1068	0.0602	0.0788	0.1474	**0.7901**
	Mixed	0.0848	0.2241	0.2883	0.4074	**0.7343**	**0.7699**
$0.8, 0.3$	Associative	0.0823	0.1139	0.5596	**0.6532**	0.1172	**0.6532**
	Disassortative	0.0836	0.0976	0.0776	0.1342	**0.7784**	**0.7931**
	Mixed	0.0888	0.1503	0.2929	0.3909	0.5428	**0.7209**

as an efficient graph feature extractor and achieve state-of-the art classification performance (Tsitsulin et al., 2018). Note that **NetLSD** extracts graph features that only depend on the graph structure and does not consider node and edge features.

Results Table 1 shows the graph classification performance on coarse graphs. In all cases, the proposed graph coarsening algorithms yield better classification accuracy than alternative methods. Interestingly, for four out of the six datasets (**NCI1**, **NCI109**, **PROTEINS**, and **PTC**) there is almost no degradation to the classification accuracy induced by coarsening, even if the graphs in the coarse dataset are five times smaller—this, we believe, is an encouraging result.

6.2 Block Recovery in the Stochastic Block Model

In this experiment, we test whether coarsening algorithms can be used to recover the block structures of random graphs sampled from stochastic block models.

The stochastic block model is a random graph model that is commonly used to evaluate graph partitioning and clustering algorithms (Abbe, 2017; Abbe et al., 2015). The model is parameterized by a probability matrix $\boldsymbol{B} \in [0,1]^{n\times n}$, with graph nodes in blocks i and j being connected with probability $\boldsymbol{B}(i,j)$. Random graphs can be generated from the stochastic block model by sampling the upper triangular entries $\boldsymbol{W}(i,j)$ in accordance with the edge probability. The lower triangular entries are then set as $\boldsymbol{W}(j,i) = \boldsymbol{W}(i,j)$.

We parameterize $\boldsymbol{B}$ with $\boldsymbol{p}$ and $\boldsymbol{q}$ as follows:

- *Assortative.* The diagonal entries of $\boldsymbol{B}$ are $\boldsymbol{p}$ and the off-diagonal entries are $\boldsymbol{q}$.
- *Disassortative.* The diagonal entries of $\boldsymbol{B}$ are $\boldsymbol{q}$ and the off-diagonal entries are $\boldsymbol{p}$.
- *Mixed.* The entries of $\boldsymbol{B}$ are randomly assigned with $\boldsymbol{p}$ and $\boldsymbol{q}$ (each with probability 1/2).

Evaluation We evaluate the performance of graph coarsening algorithms by measuring the discrepancy between the recovered graph partitions and the ground-truth blocks. We use the *Normalized Mutual Information* (**NMI**) to measure the recovery error between any two graph partitions. The definition of **NMI** can be found in the supplementary material.

For each stochastic block model setting, we set $N = 200$ and $n = 10$, with 20 nodes for each partition. We repeat the experiment 10 times and report the average **NMI** metric achieved by each method.

We compare our graph coarsening algorithms with the graph coarsening and partitioning algorithms mentioned earlier. Table 2 reports the average **NMI** in three different stochastic block model configurations. Our proposed methods outperform other methods in almost all cases. In particular, our methods achieve high recovery accuracy for disassortative and mixed settings, where traditional graph partitioning algorithms fail to recover accurately. The **EM** and **LV** coarsening algorithms are not optimized for block recovery and thus exhibit far worse performance on this task.

7 CONCLUSION

In this work, we propose a new framework for graph coarsening. We leverage the spectral properties of normalized Laplacian matrices to define a new notion of graph distance that quantifies the differences between original and coarse graphs. We argue that the proposed spectral distance naturally captures the structural changes in the graph coarsening process, and we propose graph coarsening algorithms that guarantee that the coarse graphs exhibit a bounded spectral distance. Experiments show that our proposed methods can outperform other graph coarsening algorithms on graph classification and block recovery tasks.

References

Abbe, E. (2017). Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531.

Abbe, E., Bandeira, A. S., and Hall, G. (2015). Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487.

AL-Tarimshawy, A. S. (2018). Singular graphs. *arXiv preprint arXiv:1806.07786.*

Banerjee, A. (2008). *The spectrum of the graph Laplacian as a tool for analyzing structure and evolution of networks.* PhD thesis, Verlag nicht ermittelbar.

Batson, J., Spielman, D. A., Srivastava, N., and Teng, S.-H. (2013). Spectral sparsification of graphs: theory and algorithms. *Communications of the ACM*, 56(8):87–94.

Butler, S. (2007). Interlacing for weighted graphs using the normalized laplacian. *Electronic Journal of Linear Algebra*, 16(1):8.

Butler, S. K. (2008). *Eigenvalues and structures of graphs.* ProQuest.

Chevalier, C. and Safro, I. (2009). Comparison of coarsening schemes for multilevel graph partitioning. In *International Conference on Learning and Intelligent Optimization*, pages 191–205. Springer.

Chung, F. R. (1997). *Spectral graph theory*, volume 92. American Mathematical Soc.

Dai, H., Dai, B., and Song, L. (2016). Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711.

Dhillon, I. S., Guan, Y., and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957.

Dong, Kun Benson, A. R. and Bindel, D. (2019). Network density of states. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM.

Durfee, D., Gao, Y., Goranci, G., and Peng, R. (2019). Fully dynamic spectral vertex sparsifiers and applications. *arXiv preprint arXiv:1906.10530.*

Garg, V. K. and Jaakkola, T. (2019). Solving graph compression via optimal transport. *arXiv preprint arXiv:1905.12158.*

Gu, J., Hua, B., and Liu, S. (2015). Spectral distances on graphs. *Discrete Applied Mathematics*, 190:56–74.

Hermsdorff, G. B. and Gunderson, L. M. (2019). A unifying framework for spectrum-preserving graph sparsification and coarsening. *arXiv preprint arXiv:1902.09702.*

Jin, Y. and JaJa, J. F. (2018). Learning graph-level representations with gated recurrent neural networks. *arXiv preprint arXiv:1805.07683.*

Jin, Y., JaJa, J. F., Chen, R., and Herskovits, E. H. (2015). A data-driven approach to extract connectivity structures from diffusion tensor imaging data. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 944–951. IEEE.

Jovanović, I. and Stanić, Z. (2012). Spectral distances of graphs. *Linear Algebra and its Applications*, 436(5):1425–1435.

Jovanović, I. and Stanić, Z. (2014). Spectral distances of graphs based on their different matrix representations. *Filomat*, 28(4):723–734.

Jovanovic, I. M. (2015). Some results on spectral distances of graphs. *Rev. Un. Mat. Argentina*, 56(2).

Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392.

Kersting, K., Kriege, N. M., Morris, C., Mutzel, P., and Neumann, M. (2016). Benchmark data sets for graph kernels.

Liu, Y., Dighe, A., Safavi, T., and Koutra, D. (2016). A graph summarization: A survey. *arXiv preprint arXiv:1612.04883.*

Loukas, A. (2019). Graph reduction with spectral and cut guarantees. *Journal of Machine Learning Research*, 20(116):1–42.

Loukas, A. and Vandergheynst, P. (2018). Spectrally approximating large graphs with smaller graphs. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3237–3246, Stockholmsmässan, Stockholm Sweden. PMLR.

Navlakha, S., Rastogi, R., and Shrivastava, N. (2008). Graph summarization with bounded error. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 419–432. ACM.

Park, H.-J. and Friston, K. (2013). Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411.

Purohit, M., Prakash, B. A., Kang, C., Zhang, Y., and Subrahmanian, V. (2014). Fast influence-based coarsening for large networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1296–1305. ACM.

Sciriha, I. (2007). A characterization of singular graphs. *Electronic Journal of Linear Algebra*, 16(1):38.

Shervashidze, N., Schweitzer, P., Leeuwen, E. J. v., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561.

Spielman, D. A. and Srivastava, N. (2011). Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926.

Spielman, D. A. and Teng, S.-H. (2011). Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025.

Toivonen, H., Zhou, F., Hartikainen, A., and Hinkka, A. (2011). Compression of weighted graphs. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 965–973. ACM.

Tremblay, N. and Loukas, A. (2020). Approximating spectral clustering via sampling: a review. In *Sampling Techniques for Supervised or Unsupervised Tasks*, pages 129–183. Springer.

Tsitsulin, A., Mottin, D., Karras, P., Bronstein, A., and Müller, E. (2018). Netlsd: hearing the shape of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2347–2356. ACM.

Van Dam, E. R. and Haemers, W. H. (2003). Which graphs are determined by their spectrum? *Linear Algebra and its applications*, 373:241–272.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.