Signed Graph Neural Networks: A Frequency Perspective

Anonymous authors
Paper under double-blind review

Abstract

Graph convolutional networks (GCNs) and its variants are designed for unsigned graphs containing only positive links. Many existing GCNs have been derived from the spectral domain analysis of signals lying over (unsigned) graphs and in each convolution layer they perform low-pass filtering of the input features followed by a learnable linear transformation. Their extension to signed graphs with positive as well as negative links imposes multiple issues including computational irregularities and ambiguous frequency interpretation, making the design of computationally efficient low pass filters challenging. In this paper, we address these issues via spectral analysis of signed graphs and propose two different signed graph neural networks, one keeps only low-frequency information and one also retains high-frequency information. We further introduce magnetic signed Laplacian and use its eigendecomposition for spectral analysis of directed signed graphs. We test our methods for node classification and link sign prediction tasks on signed graphs and achieve state-of-the-art performances.

1 Introduction

Graph neural networks (GNNs) learn powerful node representations by capturing local graph structure and feature information (Ma & Tang, 2021; Wu et al., 2021). The existing GNN architectures have focused almost exclusively on graphs with nonnegative edges, which encode some kind of similarity relation between the incident nodes. In contrast, negative edges are often useful to model dissimilarity relations (Kumar et al., 2016; Dittrich & Matz, 2020): for instance, in social networks, users may have common/opposite political views, trust/distrust one another's recommendations, or like/dislike each other. Such dissimilarity relations can be modeled using signed graphs by allowing the edges to take both positive or negative values. In this paper, we are interested in graph neural networks designs for signed graphs.

There are two major lines of research that consider signed links in the process of learning the node embeddings. On one hand, the network embedding methods such as SIDE (Kim et al., 2018) and SLF (Xu et al., 2019) learn node representations by optimizing an unsupervised loss that primarily aims to locate node embeddings closer to each other if they are connected by positive links and vice-versa for nodes connected by negative links. On the other hand, SGCN (Derr et al., 2018), SiGAT (Huang et al., 2019), and SNEA (Li et al., 2020) adopt GNN based models to learn node embeddings in a task-specific end-to-end manner. These GNN based models are based on structural balance theory for signed graphs (Heider, 1946; Cartwright & Harary, 1956).

In this work, we propose an alternative solution to GNNs for signed graphs from a frequency perspective via spectral domain analysis. Recall that the spectral domain analysis of unsigned graphs has been widely used to develop GNN architectures. Many well-known GNNs including spectral-GNN (Bruna et al., 2014), ChebNet (Defferrard et al., 2016), GCN (Kipf & Welling, 2017), AGCN (Li et al., 2018), and FAGCN (Bo et al., 2021) rely on spectral domain analysis. These designs are based on the frequency interpretation derived from the eigendecomposition of the normalized unsigned graph Laplacian. However, direct application of the existing spectral domain GNN designs to signed graphs is problematic, mainly due to (i) possible zero diagonal entries in the degree matrix making the normalization of the Laplacian prohibitive and (ii) possible negative eigenvalues of the graph Laplacian, making the frequency ordering somewhat ambiguous, i.e., whether the smallest negative, positive, or absolute value, eigenvalues should be used as low frequency (Knyazev, 2017).

To address these issues, we turn to signed graph signal processing (Dittrich & Matz, 2020) which provides frequency interpretation for features lying on the signed graphs, and propose spectral domain signed GNNs based on it. Specifically, we propose two different GNN designs for signed graphs: Spectral-SGCN-I and Spectral-SGCN-II. The former considers fixed low-pass filter keeping only low-frequency information during aggregation process, whereas the later is based on attention mechanism retaining low as well as high-frequency information. Extending these methods to directed signed graphs is another challenge. For handling directed signed graph, we further introduce spectral methods for directed signed graphs. We evaluate the performance of our methods on node classification and link sign prediction tasks on signed graphs. Our contributions are summarized below.

- We present a principled approach to designing graph neural networks for signed graphs based on the spectral domain analysis over signed graphs.
- We instantiate our approach with two graph neural network architectures for signed graphs, one behaves like a low-pass filter and one also retains high-frequency information.
- We introduce signed magnetic Laplacian (see Table 1) for spectral analysis of directed signed graphs and utilize it in feature aggregation process.
- We evaluate our method through extensive evaluations on node classification as well as link sign prediction tasks for signed graphs and achieve state-of-the-art performances.

Table 1: Different Laplacians and their applicability.

	Unsigned	Directed	Signed	Directed Signed
Laplacian $\mathbf{L}$	✓	✗	✗	✗
Magnetic Laplacian $\mathbf{L}^q$	✓	✓	✗	✗
Signed Laplacian $\bar{\mathbf{L}}$	✓	✗	✓	✗
Signed Magnetic Laplacian $\bar{\mathbf{L}}^q$	✓	✓	✓	✓

Related Work: There exist many different methods to learn node representations (embeddings) on signed graphs. Most of these methods are derived from the balance theory and can be classified into two categories: unsupervised network embeddings and GNN based methods. Unsupervised network embedding methods including SiNE (Wang et al., 2017), SIDE (Kim et al., 2018), SIGNet (Islam et al., 2018), SLF (Xu et al., 2019), and ASiNE (Lee et al., 2020) learn node representations in an unsupervised manner so that nodes connected via positive links are in close proximity to each other, whereas nodes connected via negative links are distant from each other. These node embeddings are then used for task in hand separately. Thus, these approaches are not end-to-end.

GNN based methods including signed graph convolutional networks (SGCN) (Derr et al., 2018), signed graph attention networks (SiGAT) (Huang et al., 2019), signed network embedding based on attention (SNEA) (Li et al., 2020), and group signed graph neural network (GS-GNN) (Liu et al., 2021) are jointly trained to learn node embeddings along with the task in hand in an end-to-end manner. SGCN is a state-of-the-art signed GNN model considering balanced and unbalanced paths motivated from the balance theory to aggregate local graph information with fixed coefficients. SNEA further extended SGCN to incorporate learnable attention coefficients for aggregating balanced and unbalanced paths. SiGAT is a motif-based GNN model to learn the node representation inspired by GAT (Veličković et al., 2018). SDGNN (Huang et al., 2021) is a recent work applicable to signed directed graphs based on balance and status theory. SSSNET (He et al., 2022a) is another GNN based work with a focus on clustering of signed graphs. For balanced graphs, the eigenvectors of the signed Laplacian follow certain properties as analyzed in (Dittrich & Matz, 2020). However, it is difficult to relate the signed spectral analysis with balance theory in case of unbalanced graphs. It is an interesting problem to explore the relationship between balance theory and spectral analysis for unbalanced graphs, which is out of scope of this work.

2 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}^+ \cup \mathcal{E}^-)$ be a directed signed graph, where $\mathcal{V}$ is the set of N number of nodes, $\mathcal{E}^+$ is the set of directed positive edges, and $\mathcal{E}^-$ is the set of directed negative edges. The adjacency matrix of the graph is denoted as $\mathbf{A} \in \mathbb{R}^{N\times N}$ and has entries from $\{+1, -1, 0\}$[1]. If (i, j) is not an edge of the graph, then the corresponding entry is $\mathbf{A}(i, j) = 0$. $\mathbf{A}(i, j) = +1$ denotes a positive edge from node i to j, whereas $\mathbf{A}(i, j) = -1$ denotes a negative edge from node i to j. Denote $\mathcal{N}_i^+$ as the set of neighbors connected to node i via positive edges and $\mathcal{N}_i^-$ as the set of neighbors connected to node i via negative edges. In addition, a feature matrix $\mathbf{X} \in \mathbb{R}^{N\times F}$ is utilized to describe nodes properties (input features), with $\mathbf{x}_i \in \mathbb{R}^N$ (column of $\mathbf{X}$) representing the i^{th} feature channel of $\mathbf{X}$ and F denotes the total number of feature channels.

2.1 Traditional GSP and Spectral Domain GNN Designs

Popular spectral domain designs of graph neural networks including ChebNet (Defferrard et al., 2016), GCN (Kipf & Welling, 2017) and their further improvements such as AGCN (Li et al., 2018), Simplified GCN (Wu et al., 2019), and FAGCN (Bo et al., 2021) are based on the spectral analysis of signals (features) defined on an unsigned graph. The spectral analysis of graph signals has been studied under the umbrella of the graph signal processing (GSP) framework (Shuman et al., 2013; Ortega et al., 2018). The graph Fourier (spectral) analysis relies on the spectral decomposition of graph Laplacians. The traditional combinatorial graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, with $\mathbf{D} = \text{diag}\{d_1, d_2, \dots, d_N\}$ and $d_i = \sum_j A_{ij}$; its normalized version is $\mathbf{L}_\text{n} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. Based on the eigendecomposition of the graph Laplacian $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U} \in \mathbb{R}^{N\times N}$ comprises of orthonormal eigenvectors and $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \dots, \lambda_N\}$ is a diagonal matrix of eigenvalues, the graph Fourier transform is defined with eigenvectors of the graph Laplacian being the graph Fourier modes (harmonics) and the corresponding eigenvalues being the graph frequencies (Shuman et al., 2013). Assuming $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_N$, λ_1 corresponds to the lowest (zero) frequency and λ_N corresponds to the highest frequency of the graph. For the case of normalized Laplacian $\mathbf{L}_\text{n}$, all the graph frequencies lie in the range $[0, 2]$ (Shuman et al., 2013), with $\lambda_1 = 0$.

Let $\mathbf{x} \in \mathbb{R}^N$ be a single-channel input signal on the graph, then the graph Fourier transform and the inverse Fourier transform are defined as $\hat{\mathbf{x}} = \mathbf{U}^T\mathbf{x}$ and $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$, respectively. Graph convolution of the input graph signal $\mathbf{x}$ with a filter $\mathbf{g}$ is $\mathbf{x} * \mathbf{g} := \mathbf{U}\hat{\mathbf{G}}\mathbf{U}^T\mathbf{x}$, where $\hat{\mathbf{G}} := \text{diag}(\hat{\mathbf{g}}) = \text{diag}\{\hat{g}_1, \dots, \hat{g}_N\}$. For computational efficiency, the filter coefficients $\hat{g}_1, \dots, \hat{g}_N$ can be approximated via K^{th} order polynomials of the graph frequencies ($K << N$), i.e., $\hat{g}(\lambda_j) = \sum_{i=0}^{K} \theta_i \lambda_j^i$ with $\boldsymbol{\theta} \in \mathbb{R}^{K+1}$ being the (polynomial) filter coefficients. In GCN, (Kipf & Welling, 2017) simplified the graph convolution by assuming first order polynomial filter ($K = 1$) with $\theta_0 = 2\theta$ and $\theta_1 = -\theta$, and thereby reducing the graph convolution to

$$\mathbf{x} * \mathbf{g} \approx \theta\ (2\mathbf{I} - \mathbf{L}_\text{n})\ \mathbf{x} = \theta\ (\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})\ \mathbf{x}. \tag{1}$$

As a different interpretation, the above can also be viewed as a combination of two operations: (i) Feature aggregation via term $(\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})$ and (ii) Feature transformation via learnable parameter θ. Note that the feature aggregation operation corresponds to low pass filtering since the spectral response of the spatial filter $2\mathbf{I} - \mathbf{L}_\text{n}$ is $\hat{g}(\lambda) = 2 - \lambda$ which amplifies low-frequency signal ($\lambda \approx 0$) and restrains high-frequency signal ($\lambda \approx 2$). In its final design, for numerical stability, the feature aggregation operation in GCN is modified by adding self-loops for each node and as a result the modified graph convolution takes the form (Kipf & Welling, 2017; Wu et al., 2019)

$$\mathbf{x} * \mathbf{g} \approx \theta\ (\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2})\ \mathbf{x}, \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$. When generalized to multi-channel input $\mathbf{X}$, the output of the ℓ^{th} layer of the GCN reads

$$\mathbf{H}^{(\ell)} = \sigma(\mathbf{P}\ \mathbf{H}^{(\ell-1)}\ \mathbf{\Theta}^{(\ell)}), \qquad \mathbf{H}^{(0)} = \mathbf{X}, \tag{3}$$

where $\mathbf{P} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ is the low-pass feature aggregation filter, $\mathbf{\Theta}^{(\ell)}$ is a learnable transformation matrix, and σ is non-linearity such as ReLU.

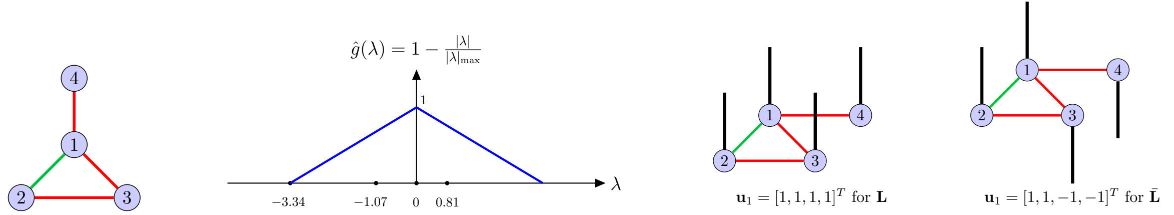
2.2 Issues with Signed Graphs

In each GCN layer, the feature aggregation operation corresponds to low pass filtering with the filter being first order polynomial in $\mathbf{L}_\text{n}$. However for signed graphs, the inverse of the degree matrix $\mathbf{D}$ (or $\tilde{\mathbf{D}}$) becomes

---

[1] For weighted signed graphs, the weighed adjacency matrix can be used taking any real value as edge weights. Our analysis and GNN models are applicable to general weighted signed graphs.

problematic since the degree values might be zero or negative values for some nodes and as a consequence the normalized Laplacian $\mathbf{L}_\mathrm{n}$ is not well defined. Since the normalized Laplacian is not well defined, one is tempted to interpret the aggregation operator as a low pass filter based on unnormalized Laplacian $\mathbf{L}$. This again poses difficulty in frequency ordering as the eigenvalues of the Laplacian $\mathbf{L}$ can take negative values for signed graphs. The graph frequencies (Laplacian eigenvalues) are ordered based on the total variation (TV) of the corresponding eigenvectors on the graph (Sandryhaila & Moura, 2014; Ortega et al., 2018). TV quantifies global smoothness (or variation) of a graph signal. For unsigned graphs, the quadratic form $\mathbf{x}^T\mathbf{L}\mathbf{x}$ is often used as TV of signal $\mathbf{x}$ on the graph (Shuman et al., 2013). However for signed graphs, the quadratic form $\mathbf{x}^T\mathbf{L}\mathbf{x}$ may take negative values and thereby invalidating its use as TV. Another definition of TV of a graph signal $\mathbf{x}$ is (Singh et al., 2016) $\mathrm{TV}(\mathbf{x}) = ||\mathbf{L}\mathbf{x}||_1$ and it can be shown that $\mathrm{TV}(\mathbf{u}_\mathrm{i}) > \mathrm{TV}(\mathbf{u}_\mathrm{j})$, if $|\lambda_i| > |\lambda_j|$. Thus the eigenvalues with smaller absolute values act as low frequencies and vice-versa.


4
1
2
3
$\hat{g}(\lambda) = 1 - \frac{|\lambda|}{|\lambda|_{\max}}$
1
λ
-3.34
-1.07
0
0.81
$\mathbf{u}_1 = [1, 1, 1, 1]^T$ for $\mathbf{L}$
$\mathbf{u}_1 = [1, 1, -1, -1]^T$ for $\bar{\mathbf{L}}$

(a) (b) (c) (d)

Figure 1: (a) A toy signed graph (Red color edges represent negative links and Green color edges represent positive links), (b) corresponding low-pass filter based on the unnormalized Laplacian, (c) Eigenvector corresponding to the lowest frequency based on the unsigned Laplacian, and (d) Eigenvector corresponding to the lowest frequency based on the signed Laplacian.

Although one can order the signed frequencies based on the absolute eigenvalues of the graph Laplacian $\mathbf{L}$, one needs new designs of low pass filters to be used as aggregation operator. For example, a low pass filter for the toy graph in Figure 1a with frequency response $\hat{g}(\lambda) = 1 - \frac{|\lambda|}{|\lambda|_{\max}}$ is shown in Figure 1b where $|\lambda|_{\max}$ is the maximum absolute frequency of the underlying graph. This filter design has certain drawbacks since it requires computation of $|\lambda|_{\max}$ and cannot be directly realized as a first order polynomial (in the graph Laplacian) in the spatial domain as the latter corresponds to a straight line. One can go for higher order filters with additional computational cost.

Moreover, the frequency interpretation also becomes ambiguous for signed graphs. Since the zero eigenvalue being the lowest frequency and the corresponding eigenvector being a constant signal on the graph (as shown in Figure 1c), it is intuitive only under similarity assumption (i.e. positive edges). For example, foes/enemies (users connected via negative links) having similar opinions (traits or features) suggests a high amount of variation and friends (users connected via positive links) having similar opinions constitutes small variation. However, this intuition is violated when using unsigned Laplacian: lowest frequency eigenvector values at nodes 1 and 3 connected via negative links in Figure 1c have same values.

3 Proposed Method

In this section we present our spectral domain analysis approach to graph neural networks for signed graphs. We then instantiate the approach to two specific network designs.

3.1 Spectral Domain Analysis of Signed Graphs

To address the issues mentioned in Section 2.2 while retaining the frequency interpretation of the aggregation operation, we turn to the signed graph signal processing (Dittrich & Matz, 2020). Instead of the standard graph Laplacian, we consider the signed Laplacian matrix (Kunegis et al., 2010; Dittrich & Matz, 2020) $\bar{\mathbf{L}} = \bar{\mathbf{D}} - \mathbf{A}$, where $\bar{\mathbf{D}} = \mathrm{diag}\{\bar{d}_1, \ldots, \bar{d}_N\}$ is a diagonal matrix with $\bar{d}_i = \sum_j |A_{ij}|$. In (Dittrich & Matz, 2020) the authors formalized the spectral domain analysis of the signals over signed graph via eigendecomposition of the signed Laplacian matrix with the eigenvalues being the signed graph frequencies and the corresponding eigenvectors being the signed graph harmonics.

More precisely, we consider the normalized signed Laplacian matrix

$$\bar{\mathbf{L}}_{\mathrm{n}} = \bar{\mathbf{D}}^{-1/2}\bar{\mathbf{L}}\bar{\mathbf{D}}^{-1/2} = \mathbf{I} - \bar{\mathbf{D}}^{-1/2}\mathbf{A}\bar{\mathbf{D}}^{-1/2}. \tag{4}$$

The eigenvalues of the normalized signed Laplacian lie in the range $[0, 2]$ with smaller eigenvalues corresponding to low frequencies, and vice-versa. This frequency ordering directly follows from using quadratic Laplacian form

$$\mathrm{TV}(\mathbf{x}) = \mathbf{x}^T\bar{\mathbf{L}}_{\mathrm{n}}\mathbf{x} \tag{5}$$

as the definition of TV on signed graphs. Using the eigenvectors of the signed Laplacian as graph harmonics provides natural frequency interpretation for signed graphs. The eigenvector corresponding to the lowest frequency of a signed graph is shown in Figure 1d. It can be seen that the nodes connected via negative links (foes) have opposite values and nodes connected via positive links (friends) have similar values thereby exhibiting small amount of variation; this phenomenon is intuitive for being a low frequency signal.

Our approach to signed graph neural networks naturally follows by redefining graph convolution based on the normalized signed Laplacian equation 4. Building on this idea, we propose below two specific graph network designs for signed graphs: Spectral-SGCN-I and Spectral-SGCN-II. The former behaves like a low-pass filtering and can be viewed as a signed graph counterpart of the vanilla GCN (Kipf & Welling, 2017). The latter is able to retain high-frequency information and can be viewed as a signed graph counterpart of FAGCN (Bo et al., 2021).

3.2 Spectral-Signed-GCN-I

Our first network design, Spectral-SGCN-I, is similar to the vanilla GCN (Kipf & Welling, 2017). It can be viewed as a low-pass feature aggregation on the underlying signed graph followed by feature transformation. At each layer, the features are first aggregated via low-pass filter $\mathbf{P} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ with $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}} = \bar{\mathbf{D}} + \mathbf{I}$. It resembles equation 2 but uses the signed Laplacian. Note that, just like equation 2, we adopt the renormalization trick to improve numerical stability (Kipf & Welling, 2017).

In more details, the aggregated features in ℓ^{th} layer is $\bar{\mathbf{H}}^{(\ell)} = \mathbf{P}\mathbf{H}^{(\ell-1)}$. Let $\bar{\mathbf{H}}^{(\ell)} = [\bar{\mathbf{h}}_1^{(\ell)}, \bar{\mathbf{h}}_2^{(\ell)}, \dots, \bar{\mathbf{h}}_N^{(\ell)}]^T$, then the aggregation can be written in the message passing form

$$\bar{\mathbf{h}}_i^{(\ell)} = \frac{1}{\bar{d}_i + 1}\mathbf{h}_i^{(\ell-1)} + \sum_{j\in\mathcal{N}_i^+}\frac{1}{\sqrt{(\bar{d}_i+1)(\bar{d}_j+1)}}\mathbf{h}_j^{(\ell-1)} - \sum_{k\in\mathcal{N}_i^-}\frac{1}{\sqrt{(\bar{d}_i+1)(\bar{d}_k+1)}}\mathbf{h}_k^{(\ell-1)}.$$

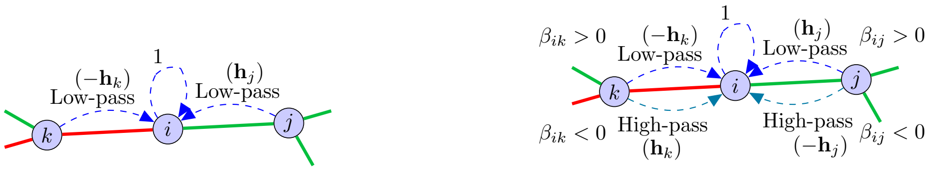
After aggregation, the features are transformed via a learnable parameter matrix along with non-linearity to give node representation output in ℓ^{th} layer as $\mathbf{H}^{(\ell)} = \sigma\left(\bar{\mathbf{H}}^{(\ell)}\ \mathbf{\Theta}^{(\ell)}\right)$.

The Spectral-SGCN-I aggregates only low frequency information via a low-pass aggregation filter as illustrated in Figure 2a. As it has been shown in (Wu et al., 2019) that removing nonlinearities and collapsing weight matrices between consecutive layers greatly simplifies the GCN complexity. Similarly, we propose spectral simplified signed graph convolution network (Spectral-S2GCN) such that the output of ℓ^{th} layer is $\mathbf{H}^{(\ell)} = \mathbf{P}^{\ell}\mathbf{X}\mathbf{\Theta}$, with $\mathbf{\Theta}$ being the only learnable parameter matrix.

3.3 Spectral-Signed-GCN-II

As has been noted in FAGCN (Bo et al., 2021), besides low frequency components, it is beneficial to incorporate the high frequency components during feature aggregation as well. Based on this idea and the frequency interpretation on signed graphs, we extend FAGCN to signed graphs by considering low as well as high frequency information. To this end, we use low pass filter $\mathbf{P}^{\mathrm{Low}} = \mathbf{I} + \bar{\mathbf{D}}^{-1/2}\mathbf{A}\bar{\mathbf{D}}^{-1/2} = 2\mathbf{I} - \bar{\mathbf{L}}_{\mathrm{n}}$ and high pass filter $\mathbf{P}^{\mathrm{High}} = \mathbf{I} - \bar{\mathbf{D}}^{-1/2}\mathbf{A}\bar{\mathbf{D}}^{-1/2} = \bar{\mathbf{L}}_{\mathrm{n}}$ along with attention to learn the proportion of low-frequency and high-frequency features to be propagated.

Let $\beta_{ij}^{\mathrm{Low}}$ be the coefficient of attention aggregation for low frequency features from node j to node i. Similarly, let $\beta_{ij}^{\mathrm{High}}$ be the coefficient of attention aggregation for high frequency features from node j to node i. Note



(a) Spectral-SGCN-I. (b) Spectral-SGCN-II.

Figure 2: Spectral-SGCN-I propagates only low-pass information from its neighbors. Spectral-SGCN-II propagates low-pass as well as high-pass information from its neighbors via attention coefficients.

$\beta_{ij}^{\text{Low}} = \beta_{ij}^{\text{High}} = 0$ if node i is not connected to node j. For target node i, define low-pass attention matrix as $\mathbf{B}_i^{\text{Low}} = \text{diag}\{\beta_{i1}^{\text{Low}}, \dots, \beta_{iN}^{\text{Low}}\}$ and high-pass attention matrix as $\mathbf{B}_i^{\text{High}} = \text{diag}\{\beta_{i1}^{\text{High}}, \dots, \beta_{iN}^{\text{High}}\}$. Let $\mathbf{H}^{(\ell-1)} = [\mathbf{h}_1^{(\ell-1)}, \mathbf{h}_2^{(\ell-1)}, \dots, \mathbf{h}_N^{(\ell-1)}]^T$ be the node embeddings at layer $\ell - 1$, then the ℓ^{th} GNN layer reads (assuming self-loops)

$$\begin{aligned}\mathbf{h}_i^{(\ell)} &= \frac{1}{2}\left(\mathbf{P}^{\text{Low}}\mathbf{B}_i^{\text{Low}}\mathbf{H}^{(\ell-1)}\right)_i + \frac{1}{2}\left(\mathbf{P}^{\text{High}}\mathbf{B}_i^{\text{High}}\mathbf{H}^{(\ell-1)}\right)_i \\ &= (\beta_{ii}^{\text{Low}} + \beta_{ii}^{\text{High}})\mathbf{h}_i^{(\ell-1)} + \sum_{j \in \mathcal{N}_i^+} \frac{(\beta_{ij}^{\text{Low}} - \beta_{ij}^{\text{High}})}{\sqrt{\bar{d}_i \bar{d}_j}}\mathbf{h}_j^{(\ell-1)} - \sum_{k \in \mathcal{N}_i^-} \frac{(\beta_{ik}^{\text{Low}} - \beta_{ik}^{\text{High}})}{\sqrt{\bar{d}_i \bar{d}_k}}\mathbf{h}_k^{(\ell-1)}.\end{aligned}$$

The coefficient $\beta_{ij}^{\text{Low}} + \beta_{ij}^{\text{High}}$ acts as a scaling factor and can be set to be 1 for simplicity. Now denote $\beta_{ij}^{\text{Low}} - \beta_{ij}^{\text{High}} = \beta_{ij}$, then the above becomes

$$\mathbf{h}_i^{(\ell)} = \mathbf{h}_i^{(\ell-1)} + \sum_{j \in \mathcal{N}_i^+} \frac{\beta_{ij}}{\sqrt{\bar{d}_i \bar{d}_j}}\mathbf{h}_j^{(\ell-1)} - \sum_{k \in \mathcal{N}_i^-} \frac{\beta_{ik}}{\sqrt{\bar{d}_i \bar{d}_k}}\mathbf{h}_k^{(\ell-1)}.$$

When the attention coefficients are constant and equal to 1, the above reduces to Spectral-SGCN-I. In Spectral-SGCN-II, the attention coefficients are learned as $\beta_{ij} = \tanh\left(\mathbf{a}^T[\mathbf{h}_i, \mathbf{h}_j]\right)$ taking values in range $[-1, 1]$, where $\mathbf{a}$ is learnable linear parameter. When $\beta_{ij} = \beta_{ij}^{\text{Low}} - \beta_{ij}^{\text{High}} > 0$, the low-frequency information is propagated from node j to node i and when $\beta_{ij} < 0$, the high frequency information is propagated, as illustrated in Figure 2b. Before passing the given input features $\mathbf{X}$ to the first layer, they are first transformed to get $\mathbf{h}_i^{(0)} = \sigma\left(\mathbf{\Theta}_1 \mathbf{x}_i\right)$ and after L number of stacked layer, we get the final output embeddings as $\mathbf{h}_i = \mathbf{\Theta}_2 \mathbf{h}_i^{(L)}$.

4 Directed Signed Graphs

The methods proposed in Section 3 are limited to undirected signed graphs. For unsigned graphs, the magnetic Laplacian (Fanuel et al., 2017; 2018; Furutani et al., 2019) has been utilized to encode the edge directionality information. Recently, Zhang et al. (2021) used magnetic Laplacian for designing GNNs for directed (unsigned) graphs. In its original form, the magnetic Laplacian is defined as

$$\mathbf{L}^q = \mathbf{D} - \mathbf{A}^q = \mathbf{D} - \mathbf{A}_s \odot \mathbf{\Phi}^q, \tag{6}$$

where $\mathbf{A}_s$ is symmetric adjacency matrix with entries $\mathbf{A}_s(i,j) = \frac{1}{2}(\mathbf{A}(i,j) + \mathbf{A}(j,i))$, $\mathbf{D} = \text{diag}\{d_1, d_2, \dots, d_N\}$ with $d_i = \sum_{j=1}^N \mathbf{A}_s(i,j)$. Moreover, $\mathbf{\Phi}^q$ is a Hermitian matrix with elements

$$\mathbf{\Phi}^q(i,j) = e^{\iota 2\pi q(\mathbf{A}(i,j) - \mathbf{A}(j,i))}, \tag{7}$$

where ι is an indeterminate satisfying $\iota^2 = -1$ and $q \in [0, 0.50)$ is the phase parameter. The normalized magnetic Laplacian is $\mathbf{L}_{\text{n}}^q = \mathbf{D}^{-1/2}\mathbf{L}^q\mathbf{D}^{-1/2}$.

It can be shown that $\mathbf{L}^q$ as well as $\mathbf{L}^q_{\mathrm{n}}$ are positive semidefinite for the unsigned case and the eigenvalues of $\mathbf{L}^q_{\mathrm{n}}$ lie in the interval $[0, 2]$. However, when the underlying graph is signed, the degree matrix can have zero diagonal entries and the normalized magnetic Laplacian is not well defined. Moreover, $\mathbf{L}^q$ becomes indefinite matrix for signed graphs. To handle these issues in directed signed graphs, we introduce *signed magnetic Laplacian*.

4.1 Signed Magnetic Laplacian

We define signed magnetic Laplacian as

$$\bar{\mathbf{L}}^q := \bar{\mathbf{D}} - \mathbf{A}^q = \bar{\mathbf{D}} - \mathbf{A}_s \odot \mathbf{\Phi}^q, \tag{8}$$

where $\mathbf{A}^q = \mathbf{A}_s \odot \mathbf{\Phi}^q$ contains the directional signed information via $\mathbf{\Phi}^q$ and the degree matrix is considered to be $\bar{\mathbf{D}} = \mathrm{diag}\{\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_N\}$ with $\bar{d}_i = \sum_{j=1}^N |\mathbf{A}_s(i,j)|$ representing the connection strength of node i. Note that the phase parameter $q \in [0, 0.25)$ for this signed directed settings (see Figure 3 for illustration of different scenarios). The normalized signed magnetic Laplacian is $\bar{\mathbf{L}}^q_{\mathrm{n}} := \bar{\mathbf{D}}^{-1/2}\bar{\mathbf{L}}^q\bar{\mathbf{D}}^{-1/2}$. The eigendecomposition of $\bar{\mathbf{L}}^q_{\mathrm{n}}$ can be used for spectral analysis of directed signed graphs and directed signed convolution operations can be defined as in Section 2.1. The spectrum of our signed magnetic Laplacians follow desirable properties as given by the following propositions.

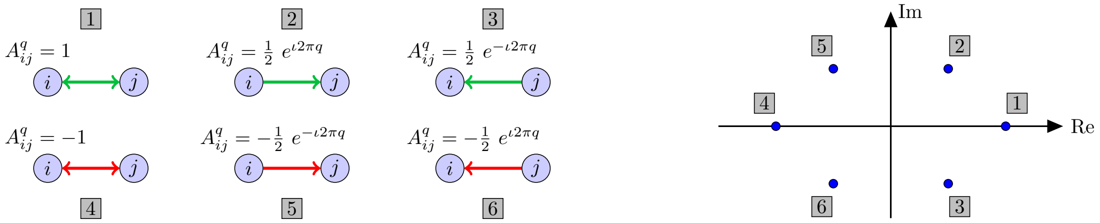

1
$A^q_{ij} = 1$
2
$A^q_{ij} = \frac{1}{2}\, e^{\iota 2\pi q}$
3
$A^q_{ij} = \frac{1}{2}\, e^{-\iota 2\pi q}$
$A^q_{ij} = -1$
$A^q_{ij} = -\frac{1}{2}\, e^{-\iota 2\pi q}$
$A^q_{ij} = -\frac{1}{2}\, e^{\iota 2\pi q}$
i
j
4
5
6
Im
Re
1
2
3
4
5
6

Figure 3: Illustration of different directed signed scenarios and singed directional phase locations on complex plane with $q = 0.125$.

Proposition 1. *The signed magnetic Laplacian $\bar{\mathbf{L}}^q$ and normalized signed magnetic Laplacian $\bar{\mathbf{L}}^q_{\mathrm{n}}$ are positive semidefinite.*

Proposition 2. *The eigenvalues of the normalized signed magnetic Laplacian $\bar{\mathbf{L}}^q_{\mathrm{n}}$ lie in* $[0, 2]$.

4.2 Directed Signed Graph Convolution Network

Based on signed magnetic Laplacian, similar to MagNet (Zhang et al., 2021), we propose Signed-MagNet. In its aggregation operation, Signed-MagNet aggregates features by performing low-pass filtering as

$$\bar{\mathbf{h}}_i^{(\ell)} = \frac{1}{\bar{d}_i + 1}\mathbf{h}_i^{(\ell-1)} + \sum_{j \in \mathcal{N}_i} \frac{\mathbf{A}^q(i,j)}{\sqrt{(\bar{d}_i + 1)(\bar{d}_j + 1)}}\mathbf{h}_j^{(\ell-1)}, \tag{9}$$

where $\mathcal{N}_i$ is the set of all the nodes connected to/from node i. Note that the latent embeddings in signed-MagNet are complex and at the last layer, we concatenate the real and imaginary parts. After aggregation operation in each layer, the features are transformed via a learnable complex matrix $\mathbf{\Theta}$.

Table 2: Signed graph datasets.

	Wiki-Editor	Wiki-Elections	Wiki-RfA	Bitcoin-Alpha	Bitcoin-OTC	Slashdot
#Nodes	21535	7194	11381	3783	5881	79120
#Classes	2	2	2	-	-	-
#Positive Links	269251	81862	139345	22650	32029	392179
#Negative Links	79004	22497	39433	1563	3563	123218
%Positive Links	77.31 %	78.44 %	77.94 %	93.54 %	89.99 %	76.09 %

5 Experiments

We evaluate our proposed methods for node classification and link sign predictions tasks on signed networks. We used Deep Graph Library (DGL) (Wang et al., 2019) for implementation of our methods. We also utilized PyTorch Geometric Signed Directed (He et al., 2022b) for implementing existing signed GNN baselines for node classification tasks.

5.1 Datasets

We perform node classification task on three datasets: Wiki-Editor, Wiki-Election, and Wiki-RfA. Wiki-Editor is extracted from the UMDWikipedia dataset (Kumar et al., 2015). There is a positive edge between two users if their co-edits belong to the same categories and a negative edge represents the co-edits belonging to different categories. Each node is labeled as either benign or vandal. Wiki-RfA (West et al., 2014) and Wiki-Election (Leskovec et al., 2010) are datasets of editors of Wikipedia that request to become administrators. A request for adminship (RfA) is submitted, either by the candidate or by another community member and any Wikipedia member may give a supporting, neutral, or opposing vote. From these votes a signed network is built for each dataset, where a positive (resp. negative) edge indicates a supporting (resp. negative) vote by a user and the corresponding candidate. The label of each node in these networks is given by the output of the corresponding request: positive (resp. negative) if the editor is chosen (resp. rejected) to become an administrator. We use dataset extraction code provided by Mercado et al. (2019) [2].

For link sign prediction, we use three additional datasets: Bitcoin-Alpha, Bitcoin-OTC, and Slashdot [3]. Bitcoin-Alpha and Bitcoin-OTC (Kumar et al., 2016; 2018) are two exchanges in Bitcoins, where nodes represent Bitcoin users and edges represent the level of trust/distrust they have in other users. A positive edge implies trust, while a negative edge represents distrust (fraud). Slashdot dataset (Kunegis et al., 2009) is a network of interactions among users on Slashdot. Nodes represent users and edges represent friends (positive) or foes (negative). The dataset statistics are summarized in Table 2.

5.2 Results

We first present experiments on synthetic data generated by signed stochastic block model Cucuringu et al. (2019); He et al. (2022a) with different levels of imbalance. We simulate two clusters with intra-cluster edge probability of 0.02 having positive signs and inter-cluster probability of 0.01 having negative signs. Such a graph is a balanced graph. We then flip the inter-cluster as well as intra-cluster edge signs with different probabilities creating varying levels of imbalance. With only 2 node labels known per cluster for a total of 1000 nodes, the node classification performance is shown in Figure 4 (one hot vectors as input features and hidden dimension of 16). Clearly, our proposed methods outperform SGCN significantly.

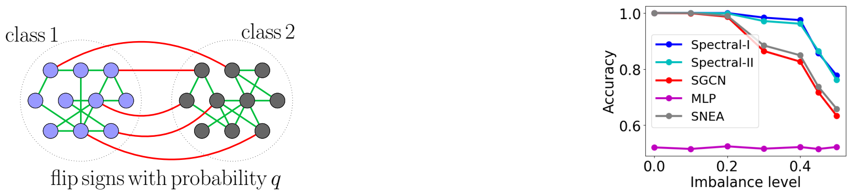


Figure 4: Performance with varying levels of imbalance.

5.2.1 Node Classification

Next, we perform node classification task in a semi-supervised setting, i.e., we have access to the test data, but not the test labels, during training. For all the three datasets, we use three different ratios for training (known labels): 1%, 2%, 5% of the total nodes. Out of the remaining nodes, we use 90% for testing, and use rest of the

---

[2] https://github.com/melopeo/GL

[3] https://networks.skewed.de

Table 3: Test accuracies averaged over 20 different seeds for data shuffles and initializations.

Dataset	WikiEditor			WikiElection			WikiRfA		
Known Labels	1 %	2 %	5 %	1 %	2 %	5 %	1 %	2 %	5 %
MLP	72.3 ± 1.7	73.6 ± 1.1	74.8 ± 0.7	90.3 ± 0.9	93.7 ± 0.8	95.5 ± 0.6	91.5 ± 0.8	92.4 ± 1.2	93.8 ± 0.5
GCN	73.0 ± 0.8	73.3 ± 0.9	74.5 ± 1.0	88.9 ± 3.4	89.8 ± 3.7	89.9 ± 3.8	88.1 ± 2.2	89.4 ± 2.2	90.0 ± 2.8
SAGE	70.3 ± 0.8	72.8 ± 0.9	76.1 ± 0.5	87.2 ± 1.2	89.7 ± 1.2	93.3 ± 0.7	87.9 ± 0.9	89.9 ± 0.8	93.3 ± 0.6
FAGCN	75.4 ± 1.1	77.2 ± 0.9	78.8 ± 0.5	92.1 ± 1.4	93.3 ± 0.9	95.4 ± 0.7	91.5 ± 1.0	92.4 ± 0.8	94.1 ± 0.4
SGCN	79.4 ± 0.7	80.3 ± 0.9	82.1 ± 0.4	87.5 ± 1.2	89.9 ± 0.7	93.8 ± 0.4	87.9 ± 1.2	89.5 ± 1.7	91.3 ± 1.5
SNEA	79.5 ± 1.1	79.9 ± 1.0	81.1 ± 1.3	93.1 ± 1.5	94.1 ± 1.7	95.2 ± 1.7	93.0 ± 1.0	94.0 ± 1.1	94.4 ± 1.2
SDGNN	80.5 ± 1.3	81.1 ± 0.9	82.2 ± 0.7	95.8 ± 1.0	96.1 ± 0.6	96.7 ± 0.8	94.8 ± 1.1	95.0 ± 0.7	95.3 ± 0.4
Spectral-SGCN-I	**81.1 ± 0.8**	81.3 ± 0.8	82.8 ± 0.4	95.7 ± 1.0	97.1 ± 0.5	97.6 ± 0.4	94.4 ± 0.7	95.1 ± 0.5	95.7 ± 0.6
Spectral-SGCN-II	80.0 ± 2.2	80.9 ± 1.5	82.3 ± 1.4	96.7 ± 1.0	97.4 ± 0.3	**98.0 ± 0.3**	94.9 ± 0.4	95.6 ± 0.4	96.0 ± 0.2
Spectral-S2GC	81.1 ± 0.8	81.8 ± 0.7	82.7 ± 0.7	96.6 ± 0.9	97.3 ± 0.5	97.8 ± 0.2	94.3 ± 0.7	95.2 ± 0.4	95.9 ± 0.2
Signed-MagNet	81.0 ± 1.1	**82.0 ± 0.8**	**83.0 ± 0.4**	**97.2 ± 0.6**	**97.6 ± 0.3**	98.0 ± 0.2	**96.0 ± 0.4**	**96.2 ± 0.3**	**96.4 ± 0.2**

nodes for validation. Since the features are not given for the nodes, we use truncated SVD of the symmetric adjacency matrix with dimension of 64 as input features. For comparison, we use traditional unsigned GNN designs namely GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), and FAGCN (Bo et al., 2021). For these methods, we do not consider the sign of the links, since the signed edge information is not applicable for these methods. We also use the state-of-the-art GNN designs based on balance theory including SGCN (Derr et al., 2018), SNEA (Li et al., 2020), and SDGNN (Huang et al., 2021) for comparison. For fair comparison, we use two layer networks with hidden dimension of 64 for all the GNN-based methods. Binary cross-entropy loss based on the known labels is used as a loss function. We use ReLU as the non-linearity function in between the layers. Adam is used as the optimizer along with ℓ_2-regularization to avoid overfitting. We tune the learning rate and weight decay (ℓ_2-regularization) hyperparameters over validation data using a grid search. For Signed-MagNet implementation, we fix $q = 0.125$ for all the experiments. Further details on implementation and hyperparameter tuning are provided in Appendix.

The classification results are summarized in Table 3. The experiments were run for 300 epochs and the results are averaged over 20 different random splits of training and test data. The average of best accuracies along with the standard deviation over 20 runs is reported. The best performing model for each dataset is in bold. We observe that our proposed signed GNNs consistently outperform the other methods in all the three datasets. For the two datasets Wiki-Election and Wiki-RfA, even traditional methods without signed information outperform SGCN.

5.2.2 Link Sign Prediction

Finally, we evaluate our methods for the task of link sign prediction that aims to predict the missing sign of a given edge. There exist three type of links in a signed graph: positive link, negative link, and no link. Denote this as a set $\mathcal{S} \in \{+, -, ?\}$, with "?" representing no link. Specifically, the training data contains a set of nodes $\mathcal{V}_t$ and a set of link triplets $\mathcal{T}$ consisting of triplets of the form (u, v, s_{uv}) with $u, v \in \mathcal{V}$ being node pairs and $s_{uv} \in \mathcal{S}$ denoting type of link between u and v. The final embeddings (obtained from the GNN model) of the two nodes u and v are concatenated together $[\mathbf{h}_u, \mathbf{h}_v]$ as the set of features for the edge and then fed to a three-class MLP classifier. The models are trained using the labeled edges from the training data. Let one-hot encoded vector of link type s_{uv} be $\mathbf{s}_{uv} \in \{0, 1\}^{|\mathcal{S}|}$. We use multi-class (three) cross entropy loss:

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{W}) = -\frac{1}{|\mathcal{T}|} \sum_{(u,v,s_{uv}) \in \mathcal{T}} \sum_{c=1}^{|\mathcal{S}|} \mathbf{s}_{uv}(c) \log(\hat{\mathbf{s}}_{uv}(c)),$$

where $\hat{\mathbf{s}}_{uv}(c)$ is the predicted probability for class c via softmax function. In the above loss function, $\mathbf{\Theta}$ denotes the set of GNN parameters and $\mathbf{W}$ denotes the parameters of MLP classifier. We use 80% of the links for training and rest 20% for testing. We use twice the number of training links as "no links" obtained using negative sampling.

Table 4: Link prediction results with Macro-F1 and Micro-F1 scores over 10 different runs.

Method	Bitcoin-Alpha		Bitcoin-OTC		Slashdot		WikiElection		WikiEditor		WikiRfA	
SiNE	0.6790	0.9440	0.6832	0.9328	0.7238	0.8192	0.6940	0.7832	0.7858	0.8244	0.7263	0.8160
SLF	0.7475	0.9453	0.7483	0.9466	**0.7943**	**0.8564**	0.7616	0.8487	0.7521	0.8442	**0.7881**	**0.8640**
SGCN	0.6648	0.9184	0.7420	0.9012	0.7422	0.8260	0.7363	0.8360	0.8225	0.8548	0.7510	0.8430
SNEA	0.6796	0.9210	0.7580	0.9038	0.7431	0.8308	0.7388	0.8372	0.8290	0.8672	0.7542	0.8497
SDGNN	0.7412	0.9480	0.8020	0.9346	0.7820	0.8504	0.7550	0.8508	0.8529	0.8810	0.7832	0.8598
Spectral-SGCN-I	0.7518	0.9547	0.8371	0.9437	0.7705	0.8410	0.7466	0.8390	0.8560	0.9128	0.7740	0.8410
Spectral-SGCN-II	0.7630	**0.9562**	0.8356	0.9538	0.7724	0.8438	0.7645	0.8512	**0.8825**	**0.9237**	0.7735	0.8476
Spectral-S2GCN	0.7030	0.9418	0.7820	0.9284	0.7352	0.8194	0.7182	0.8390	0.8448	0.9083	0.7221	0.8278
Singned-Magnet	**0.7880**	0.9432	**0.8548**	**0.9580**	0.7781	0.8463	**0.7818**	**0.8649**	0.8652	0.9184	0.7692	0.8543

Table 5: Link prediction results with AUC scores over 10 different runs.

Method	Bitcoin-Alpha	Bitcoin-OTC	Slashdot	WikiElection	WikiEditor	WikiRfA
SiNE	0.8351 ± 0.0126	0.8575 ± 0.0053	0.8108 ± 0.0021	0.8040 ± 0.0072	0.8631 ± 0.0044	0.7963 ± 0.0260
SLF	0.8438 ± 0.0151	0.8670 ± 0.0052	**0.8846 ± 0.0040**	0.8803 ± 0.0025	0.9090 ± 0.0027	0.8709 ± 0.0012
SGCN	0.8420 ± 0.0147	0.8780 ± 0.0103	0.8543 ± 0.0064	0.8516 ± 0.0030	0.9068 ± 0.0040	0.8361 ± 0.0078
SNEA	0.8453 ± 0.0062	0.8792 ± 0.0028	0.8621 ± 0.0154	0.8412 ± 0.0083	0.9278 ± 0.0062	0.8259 ± 0.0036
SDGNN	0.9008 ± 0.0081	0.9128 ± 0.0073	0.8734 ± 0.0187	0.8763 ± 0.0134	0.9430 ± 0.0126	**0.8870 ± 0.0048**
Spectral-SGCN-I	0.9005 ± 0.0345	0.9079 ± 0.0176	0.8345 ± 0.0121	0.8559 ± 0.0316	0.9447 ± 0.0194	0.8228 ± 0.0333
Spectral-SGCN-II	0.9146 ± 0.0066	0.9309 ± 0.0044	0.8677 ± 0.0090	0.8840 ± 0.0018	**0.9818 ± 0.0047**	0.8420 ± 0.0026
Spectral-S2GCN	0.8670 ± 0.0176	0.8936 ± 0.0095	0.8273 ± 0.0291	0.8149 ± 0.0315	0.9375 ± 0.0121	0.8242 ± 0.0189
Signed-MagNet	**0.9227 ± 0.0097**	**0.9410 ± 0.0076**	0.8615 ± 0.0074	**0.8881 ± 0.0026**	0.9567 ± 0.0144	0.8612 ± 0.0032

We use SiNE (Wang et al., 2017), SLF (Xu et al., 2019), SGCN (Derr et al., 2018), SNEA (Li et al., 2020), and SDGNN (Huang et al., 2021) as baselines for comparison on link sign prediction tasks. We use a two layer GNN model along with a single hidden layer MLP classifier. Truncated SVD of the symmetric adjacency matrix with dimension of 30 is used as input features. We utilize Marco-F1 and Micro-F1 scores for evaluation, since the positive and negative links are unbalanced. The comparison results in terms of F1 scores are listed in Table 4. Moreover, the results in terms of Area Under the receiver operating characteristic Curve (AUC) scores in Table 5. The results reported in these tables are average scores over 10 different runs. We observe that our methods outperform the baselines for all the dataset except for Slashdot and WikiRfA. Note that all the numbers for other algorithms in the table are obtained by running the official codes provided by the respective authors. These could be slightly different from the numbers reported in those papers that may use filtered/truncated versions of datasets.

6 Conclusion

In this paper, we proposed a new framework for GNN design for signed graphs based on spectral domain analysis over signed graphs, as opposed to existing balance theory based GNN methods. We also introduced signed magnetic Laplacian for handling directed signed graphs. We evaluated our methods for node classification as well as link sign prediction tasks on signed graphs and achieved state-of-the-art performance.

Limitations and Ethical Considerations: One limitation of our method is that it does not consider one scenario in directed signed graphs when there exist two opposite (directed) edges with different signs signs between a pair of nodes. All of our data is publicly available for research purposes and does not contain personally identifiable information or offensive content. The methods presented here has no greater or lesser impact on society than other graph neural network algorithms.

References

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3950–3957, 2021.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and deep locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277, 1956.

Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. SPONGE: A generalized eigenproblem for clustering signed networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1088–1098. PMLR, 2019.

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

Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 929–934. IEEE, 2018.

Thomas Dittrich and Gerald Matz. Signal processing on signed graphs: Fundamentals and potentials. *IEEE Signal Processing Magazine*, 37(6):86–98, 2020. doi: 10.1109/MSP.2020.3014060.

Michaël Fanuel, Carlos M Alaiz, and Johan AK Suykens. Magnetic eigenmaps for community detection in directed networks. *Physical Review E*, 95(2):022302, 2017.

Michaël Fanuel, Carlos M Alaíz, Ángela Fernández, and Johan AK Suykens. Magnetic eigenmaps for the visualization of directed networks. *Applied and Computational Harmonic Analysis*, 44(1):189–199, 2018.

Satoshi Furutani, Toshiki Shibahara, Mitsuaki Akiyama, Kunio Hato, and Masaki Aida. Graph signal processing for directed graphs based on the hermitian laplacian. In *ECML/PKDD (1)*, pp. 447–463, 2019.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yixuan He, Gesine Reinert, Songchao Wang, and Mihai Cucuringu. SSSNET: Semi-supervised signed network clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 244–252. SIAM, 2022a.

Yixuan He, Xitong Zhang, Junjie Huang, Benedek Rozemberczki, Mihai Cucuringu, and Gesine Reinert. PyTorch Geometric Signed Directed: A Survey and Software on Graph Neural Networks for Signed and Directed Graphs. *arXiv preprint arXiv:2202.10793*, 2022b.

Fritz Heider. Attitudes and cognitive organization. *The Journal of psychology*, 21(1):107–112, 1946.

Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Signed graph attention networks. In *International Conference on Artificial Neural Networks*, pp. 566–577. Springer, 2019.

Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. SDGNN: Learning node representation for signed directed networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 196–203, 2021.

Mohammad Raihanul Islam, B Aditya Prakash, and Naren Ramakrishnan. SIGNet: Scalable embeddings for signed networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 157–169. Springer, 2018.

Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang. SIDE: representation learning in signed directed networks. In *Proceedings of the 2018 World Wide Web Conference*, pp. 509–518, 2018.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Andrew V Knyazev. Signed Laplacian for spectral clustering revisited. *arXiv preprint arXiv:1701.01394*, 1, 2017.

Srijan Kumar, Francesca Spezzano, and VS Subrahmanian. VEWS: A Wikipedia vandal early warning system. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 607–616, 2015.

Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 221–230. IEEE, 2016.

Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and VS Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 333–341. ACM, 2018.

Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. The Slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pp. 741–750, 2009.

Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 559–570. SIAM, 2010.

Yeon-Chang Lee, Nayoun Seo, Kyungsik Han, and Sang-Wook Kim. ASiNE: Adversarial signed network embedding. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 609–618, 2020.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pp. 641–650, 2010.

Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yu Li, Yuan Tian, Jiawei Zhang, and Yi Chang. Learning signed network embedding via graph attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4772–4779, 2020.

Haoxin Liu, Ziwei Zhang, Peng Cui, Yafeng Zhang, Qiang Cui, Jiashuo Liu, and Wenwu Zhu. Signed graph neural network with latent groups. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1066–1075, 2021.

Yao Ma and Jiliang Tang. *Deep learning on graphs*. Cambridge University Press, 2021.

Pedro Mercado, Jessica Bosch, and Martin Stoll. Node classification for signed social networks using diffuse interface methods. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 524–540. Springer, 2019.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

Aliaksei Sandryhaila and Jose MF Moura. Discrete signal processing on graphs: Frequency analysis. *IEEE Transactions on Signal Processing*, 62(12):3042–3054, 2014.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

Rahul Singh, Abhishek Chakraborty, and BS Manoj. Graph Fourier transform based on directed Laplacian. In *2016 International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–5. IEEE, 2016.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In *Proceedings of the 2017 SIAM international conference on data mining*, pp. 327–335. SIAM, 2017.

Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2: 297–310, 2014.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24, 2021.

Pinghua Xu, Wenbin Hu, Jia Wu, and Bo Du. Link prediction with signed latent factors in signed social networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1046–1054, 2019.

Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. Magnet: A neural network for directed graphs. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

A Proof of Proposition 1

Proof. Let $\mathbf{X}^{\dagger}$ be the conjugate transpose of $\mathbf{X}$ and let $\mathbf{x} \in \mathbb{C}^N$. It is easy to see that $\bar{\mathbf{L}}^q$ is a Hermitian matrix and therefore, the imaginary part $Im(\mathbf{x}^{\dagger}\bar{\mathbf{L}}^q\mathbf{x}) = 0$. Denoting $\mathbf{\Phi}^q(m,n) = e^{\iota \mathbf{\Theta}^q(m,n)}$, where $\mathbf{\Theta}^q(m,n) = 2\pi q(\mathbf{A}(m,n) - \mathbf{A}(n,m))$. The real part

$$\begin{aligned}
Re(\mathbf{x}^{\dagger}\bar{\mathbf{L}}^q\mathbf{x}) &= \sum_{n=1}^{N} \bar{\mathbf{D}}(n,n)\mathbf{x}(n)\mathbf{x}^*(n) - \sum_{m,n=1}^{N} \mathbf{A}_s(m,n)\mathbf{x}(m)\mathbf{x}^*(n)\cos(\mathbf{\Theta}^q(m,n)) \\
&= \sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)||\mathbf{x}(m)|^2 - \sum_{m,n=1}^{N} \mathbf{A}_s(m,n)\mathbf{x}(m)\mathbf{x}^*(n)\cos(\mathbf{\Theta}^q(m,n)) \\
&\geq \frac{1}{2}\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)||\mathbf{x}(m)|^2 + \frac{1}{2}\sum_{n,m=1}^{N} |\mathbf{A}_s(n,m)||\mathbf{x}(n)|^2 \\
&\quad - \sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)\mathbf{x}(m)\mathbf{x}^*(n)\cos(\mathbf{\Theta}^q(m,n))| \\
&= \frac{1}{2}\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)| \left(|\mathbf{x}(m)|^2 + |\mathbf{x}(n)|^2 - 2|\mathbf{x}(m)\mathbf{x}^*(n)\cos(\mathbf{\Theta}^q(m,n))|\right) \\
&\geq \frac{1}{2}\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)| \left(|\mathbf{x}(m)|^2 + |\mathbf{x}(n)|^2 - 2|\mathbf{x}(m)||\mathbf{x}(n)|\right) \\
&= \frac{1}{2}\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)| \left(|\mathbf{x}(m)| - |\mathbf{x}(n)|\right)^2 \\
&\geq 0.
\end{aligned}$$

Letting $\mathbf{y} = \bar{\mathbf{D}}^{-1/2}\mathbf{x}$ and by definition of $\bar{\mathbf{L}}^q_{\mathrm{n}}$, we have

$$\mathbf{x}^{\dagger}\bar{\mathbf{L}}^q_{\mathrm{n}}\mathbf{x} = \mathbf{x}^{\dagger}\bar{\mathbf{D}}^{-1/2}\bar{\mathbf{L}}^q\bar{\mathbf{D}}^{1/2}\mathbf{x} = \mathbf{y}^{\dagger}\bar{\mathbf{L}}^q\mathbf{y} \geq 0.$$

□

B Proof of Proposition 2

Proof. Since $\bar{\mathbf{L}}^q_{\mathrm{n}}$ is positive semidefinite due to Proposition 1, we just show that the largest eigenvalue $\lambda_N \leq 2$. We know that the eigenvalue with largest absolute value is

$$\lambda_N = \max_{\mathbf{x}\neq 0} \frac{\mathbf{x}^{\dagger}\bar{\mathbf{L}}^q_{\mathrm{n}}\mathbf{x}}{\mathbf{x}^{\dagger}\mathbf{x}}.$$

Letting $\mathbf{y} = \bar{\mathbf{D}}^{-1/2}\mathbf{x}$, we have

$$\lambda_N = \max_{\mathbf{x}\neq 0} \frac{\mathbf{x}^{\dagger}\bar{\mathbf{D}}^{-1/2}\bar{\mathbf{L}}^q\bar{\mathbf{D}}^{1/2}\mathbf{x}}{\mathbf{x}^{\dagger}\mathbf{x}} = \max_{\mathbf{y}\neq 0} \frac{\mathbf{y}^{\dagger}\bar{\mathbf{L}}^q\mathbf{y}}{\mathbf{y}^{\dagger}\bar{\mathbf{D}}\mathbf{y}}.$$

Since the numerator in the above

$$
\begin{aligned}
\mathbf{y}^\dagger \bar{\mathbf{L}}^q \mathbf{y} &= \sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)||\mathbf{y}(m)|^2 - \sum_{m,n=1}^{N} \mathbf{A}_s(m,n)\mathbf{y}(m)\mathbf{y}^*(n)\cos(\boldsymbol{\Theta}^q(m,n)) \\
&= \frac{1}{2}\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)||\mathbf{y}(m)|^2 + \frac{1}{2}\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)||\mathbf{y}(n)|^2 \\
&\quad - \sum_{m,n=1}^{N} \mathbf{A}_s(m,n)\mathbf{y}(m)\mathbf{y}^*(n)\cos(\boldsymbol{\Theta}^q(m,n)) \\
&\le \frac{1}{2}\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)| \left(|\mathbf{y}(m)| + |\mathbf{y}(n)|\right)^2 \\
&\le \sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)| \left(|\mathbf{y}(m)|^2 + |\mathbf{y}(n)|^2\right) \\
&= 2\sum_{m,n=1}^{N} |\mathbf{A}_s(m,n)| \; |\mathbf{y}(m)|^2 \\
&= 2\sum_{m=1}^{N} \left(\sum_{n=1}^{N} |\mathbf{A}_s(m,n)|\right) \; |\mathbf{y}(m)|^2 \\
&= 2\sum_{m=1}^{N} \bar{\mathbf{D}}(m,m) \; |\mathbf{y}(m)|^2 \\
&= 2\mathbf{y}^\dagger \bar{\mathbf{D}} \mathbf{y},
\end{aligned}
$$

and thus, $\lambda_N \le 2$. □

C Further Implementation Details

All the experiments were run on Intel Core i9-9900 machine equipped with NVIDIA GeForce RTX 2080 Ti GPU. We use two layer networks with hidden dimension of 64 for all the GNN-based methods (a standard practice in unsigned GNN literature). ReLU nonlinearity was used in all the experiments. For the implementation of Signed-Magnet, we used ReLU non-linearity for real and complex parts separately. The only hyperparameters to tune are learning rate and regularization (weight decay) coefficient. For all of our methods we use feature dropout with a rate of 0.5. For Spectral-SGCN-II, we use attention and feature dropout with dropout rate of 0.5. We tune the learning rate with different values (on log scale) in the range $[1e^{-3}, 1e^{-1}]$ and regularization rate in the range $[1e^{-6}, 1e^{-3}]$. For node classification task, the hyperparameters were tuned based on the validation accuracy with 1% known training labels for each dataset.

Complexity Analysis: Although the time complexity of the existing signed GNN designs (balance theory based) and our methods is $\mathcal{O}(|E|)$ for sparse graphs or $\mathcal{O}(|V|^2)$ in worst-case, the number of parameters per layer for these methods are different. For example, SDGNN utilizes four different encoders for capturing different directionality scenarios. In contrast, our signed spectral methods employ a single (transformation) encoder in each layer. We provide training times per epoch (averaged over 300 epochs) for different signed GNN methods in Figure 5.

D Clustering of Directed Signed Graphs

We present experiments on synthetic data generated by directed signed stochastic block model with different levels of imbalance. In particular, we simulate two clusters (classes) with directed intra-cluster edge probability of 0.05 having positive signs and directed inter-cluster edge probability of 0.05 having negative signs. Such a

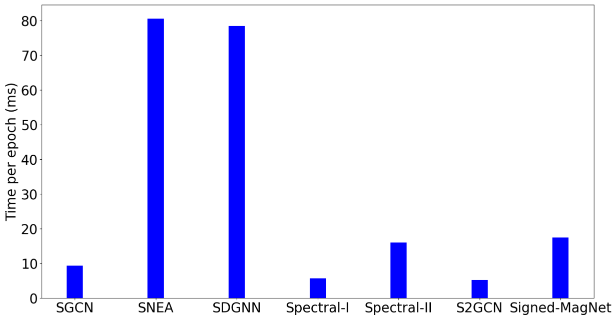
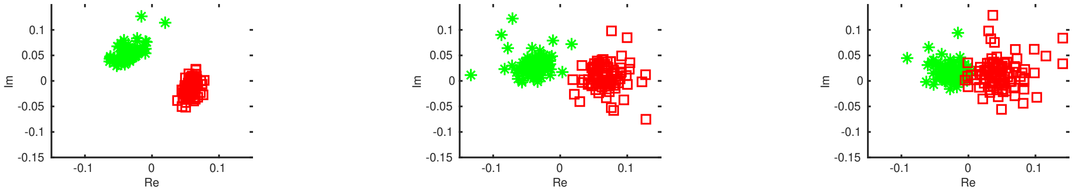


Figure 5: Time taken per epoch for different methods (on WikiRfA dataset for node classification task with batch size = 64, number of GNN layers = 2, hidden dimension = 64).

graph is a balanced graph. We then flip the inter-cluster as well as intra-cluster edge signs with different probabilities creating varying levels of imbalance.



(a) imbalance prob. = 0.

(b) imbalance prob. = 0.1.

(c) imbalance prob. = 0.2.

Figure 6: Clustering results with 100 nodes per cluster for directed signed stochastic block model.

The nodes can be clustered based on the first (complex) eigenvector of the signed magnetic Laplacian corresponding to the eigenvalue with smallest absolute value plotted on the complex plane. Figure 6 shows clustering results on directed signed graphs with varying levels of imbalance.

E Connections to SGCN

SGCN (Derr et al., 2018) in its design consider balanced and unbalanced node sets based on balance theory in feature aggregation process. The balanced node set for a target node i is the set of nodes that have even number of negative links along a path connecting to i. An ℓ-hop balanced set of nodes for target node i is denoted as $B_i(\ell)$ and unbalanced set of nodes as $U_i(\ell)$. For example graph in Figure 1, $B_1(1) = \{2\}$ and $U_1(1) = \{3, 4\}$.

The node representations for these balanced and unbalanced sets are treated separately in feature aggregation process and are concatenated together to form final node embeddings. In the ℓ^{th} layer of the model, it reads

$$\mathbf{h}_i^{B(\ell)} = \sigma\left(\boldsymbol{\Theta}^{B(\ell)}\left[\sum_{j\in\mathcal{N}_i^+}\frac{\mathbf{h}_j^{B(\ell-1)}}{|\mathcal{N}_i^+|}, \sum_{k\in\mathcal{N}_i^-}\frac{\mathbf{h}_j^{U(\ell-1)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{B(\ell-1)}\right]\right)$$
$$\mathbf{h}_i^{U(\ell)} = \sigma\left(\boldsymbol{\Theta}^{U(\ell)}\left[\sum_{j\in\mathcal{N}_i^+}\frac{\mathbf{h}_j^{U(\ell-1)}}{|\mathcal{N}_i^+|}, \sum_{k\in\mathcal{N}_i^-}\frac{\mathbf{h}_j^{B(\ell-1)}}{|\mathcal{N}_i^-|}, \mathbf{h}_i^{U(\ell-1)}\right]\right),$$

where $\boldsymbol{\Theta}^{B(\ell)}$ and $\boldsymbol{\Theta}^{U(\ell)}$ are linear transformation parameters for balanced and unbalanced paths, respectively and the node representation at ℓ^{th} layer is the concatenation of the two embeddings $\mathbf{h}^{(\ell)} = [\mathbf{h}_i^{B(\ell)}, \mathbf{h}_i^{U(\ell)}]$.

Instead of treating positive and negative neighbors separately, in our architecture of Spectral-SGCN-I we are aggregating them weighted by their signs and corresponding (absolute) degrees as can be seen from equation 9.

F Effect of Input Feature Size

We also perform the study on the classification accuracy with varying number of input feature sizes. Figure 7 shows the performance of Spectral-SGCN-II with respect to varying number of input features. As expected, the performance improves with increase in the dimension of input features.

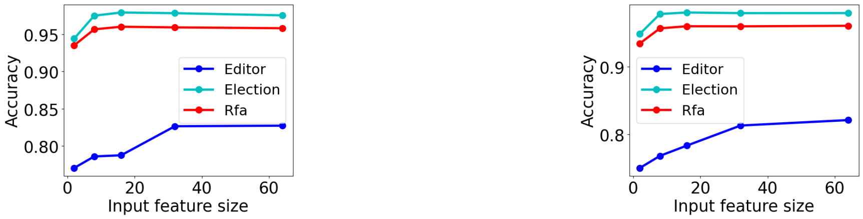


Figure 7: Spectral-SGCN-I (left) and Spectral-SGCN-II (right) performance with varying dimensions of input features (5% known labels).