Higher-order Sparse Convolutions in Graph Neural Networks

Jhony H. Giraldo, Sajid Javed, Arif Mahmood, Fragkiskos D. Malliaros, Thierry Bouwmans

HIGHER-ORDER SPARSE CONVOLUTIONS IN GRAPH NEURAL NETWORKS

Jhony H. Giraldo[⋆+]*, Sajid Javed*[‡]*, Arif Mahmood*[⊛]*, Fragkiskos D. Malliaros*[†]*, Thierry Bouwmans*[§]

[⋆]LTCI, Télécom Paris - Institut Polytechnique de Paris, France;
[‡]Khalifa University, United Arab Emirates; [⊛]Information Technology University, Pakistan;
[†]Université Paris-Saclay, CentraleSupélec, Inria, Centre for Visual Computing (CVN), France;
[§]Laboratoire MIA, La Rochelle Université, France

ABSTRACT

Graph Neural Networks (GNNs) have been applied to many problems in computer sciences. Capturing higher-order relationships between nodes is crucial to increase the expressive power of GNNs. However, existing methods to capture these relationships could be infeasible for large-scale graphs. In this work, we introduce a new higher-order sparse convolution based on the Sobolev norm of graph signals. Our Sparse Sobolev GNN (S-SobGNN) computes a cascade of filters on each layer with increasing Hadamard powers to get a more diverse set of functions, and then a linear combination layer weights the embeddings of each filter. We evaluate S-SobGNN in several applications of semi-supervised learning. S-SobGNN shows competitive performance in all applications as compared to several state-of-the-art methods.

***Index Terms*—** Graph neural networks, sparse convolutions, Sobolev norm

1. INTRODUCTION

Graph representation learning and its applications have gained significant attention in recent years. Notably, Graph Neural Networks (GNNs) have been extensively studied [1–6]. GNNs extend the concepts of Convolutional Neural Networks (CNNs) [7] to non-Euclidean data modeled as graphs. GNNs have numerous applications like semi-supervised learning [2], graph clustering [8], point cloud semantic segmentation [9], misinformation detection [10], and protein modeling [11]. Similarly, other graph learning techniques have been recently applied to image and video processing applications [12, 13].

Most GNNs update their node embeddings by computing specific operations in the neighborhood of each node. This updating is limited when we want to capture higher-order vertex relationships between nodes. Previous methods in GNNs have tried to capture these higher-order connections by taking powers of the sparse adjacency matrix [14], quickly converting this sparse representation into a dense matrix. The densification of the adjacency matrix results in memory and scalability problems in GNNs. Therefore, the use of these higher-order methods is limited for large-scale graphs.

In this work, we propose a new sparse GNN model that computes a cascade of higher-order filtering operations. Our model is inspired by the Sobolev norm in Graph Signal Processing (GSP) [15, 16]. We modify the Sobolev norm using concepts of the Hadamard product between matrices to maintain the sparsity of the adjacency matrix. We rely on spectral graph theory [17] and the Schur product theorem [18] to explain some mathematical properties of our filtering operation. Our Sparse Sobolev GNN (S-SobGNN) employs a linear combination layer at the end of each cascade of filters to select the best power functions. Thus, we improve expressiveness by computing a more diverse set of sparse graph-convolutional operations. We evaluate S-SobGNN in semi-supervised learning tasks in several domains like tissue phenotyping in colon cancer histology images [19], text classification of news [20], activity recognition with sensors [21], and recognition of spoken letters [22].

The main contributions of the current work are summarized as follows: 1) we propose a new GNN architecture that computes a cascade of higher-order filters inspired by the Sobolev norm in GSP, 2) some mathematical insights of S-SobGNN are introduced based on spectral graph theory [17] and the Schur product theorem [18], and 3) we perform experimental evaluations on four publicly available benchmark datasets and compared S-SobGNN to seven GNN architectures. Our algorithm shows the best performance against previous methods. The rest of the paper is organized as follows. Section 2 introduces the proposed GNN model. Section 3 presents the experimental framework and results. Finally, Section 4 shows the conclusions.

2. SPARSE SOBOLEV GRAPH NEURAL NETWORKS

2.1. Preliminaries

A graph is represented as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of N nodes and $\mathcal{E} = \{(i, j)\}$ is the set of edges

[+]Corresponding author: jhony.giraldo@telecom-paris.fr

between nodes i and j. $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix of the graph such that $\mathbf{A}(i,j) = a_{i,j} \in \mathbb{R}_+$ is the weight of the edge (i,j), and $\mathbf{A}(i,j) = 0 \; \forall \; (i,j) \notin \mathcal{E}$. As a result, $\mathbf{A}$ is symmetric for undirected graphs. A graph signal is a function $x : \mathcal{V} \to \mathbb{R}$ and is represented as $\mathbf{x} \in \mathbb{R}^N$. The degree matrix of G is a diagonal matrix given by $\mathbf{D} = \text{diag}(\mathbf{A1})$. $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the combinatorial Laplacian matrix, and $\mathbf{\Delta} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the symmetric normalized Laplacian [23]. The Laplacian matrix is a positive semi-definite matrix for undirected graphs with eigenvalues[1] $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and corresponding eigenvectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$. In GSP, the Graph Fourier Transform (GFT) of $\mathbf{x}$ is given by $\hat{\mathbf{x}} = \mathbf{U}^\mathsf{T}\mathbf{x}$, and the inverse GFT is $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$ [23]. In this work, we use the spectral definitions of graphs to analyze our filtering operation. However, the spectrum is not required for the implementation of S-SobGNN.

2.2. Sobolev Norm

The Sobolev norm in GSP has been used as a regularization term to solve problems in 1) video processing [13, 24], 2) modeling of infectious diseases [25], and 3) interpolation of graph signals [15, 16].

Definition 1. *For fixed parameters $\epsilon \geq 0$, $\rho \in \mathbb{R}$, the Sobolev norm is given by $\|\mathbf{x}\|_{\rho,\epsilon} \triangleq \|(\mathbf{L} + \epsilon\mathbf{I})^{\rho/2}\mathbf{x}\|$ [15].*

When $\mathbf{L}$ is symmetric, we have that $\|\mathbf{x}\|^2_{\rho,\epsilon}$ is given by:

$$\|\mathbf{x}\|^2_{\rho,\epsilon} = \mathbf{x}^\mathsf{T}(\mathbf{L} + \epsilon\mathbf{I})^\rho\mathbf{x}. \tag{1}$$

We divide the analysis of (1) into two parts: 1) when $\epsilon = 0$, and 2) when $\rho = 1$. For $\epsilon = 0$ in (1) we have:

$$\mathbf{x}^\mathsf{T}\mathbf{L}^\rho\mathbf{x} = \mathbf{x}^\mathsf{T}\mathbf{U}\mathbf{\Lambda}^\rho\mathbf{U}^\mathsf{T}\mathbf{x} = \hat{\mathbf{x}}^\mathsf{T}\mathbf{\Lambda}^\rho\hat{\mathbf{x}} = \sum_{i=1}^{N}\hat{\mathbf{x}}^2(i)\lambda_i^\rho. \tag{2}$$

Notice that the spectral components $\hat{\mathbf{x}}(i)$ are penalized with powers of the eigenvalues λ_i^ρ of $\mathbf{L}$. Since the eigenvalues are ordered in increasing order, the higher frequencies of $\hat{\mathbf{x}}$ are penalized more than the lower frequencies when $\rho = 1$, leading to a smooth function in G. For $\rho > 1$, the GFT $\hat{\mathbf{x}}$ is penalized with a more diverse set of eigenvalues. We can have a similar analysis for the adjacency matrix $\mathbf{A}$ using the eigenvalue decomposition $\mathbf{A}^\rho = (\mathbf{V}\mathbf{\Sigma}\mathbf{V}^\mathsf{H})^\rho = \mathbf{V}\mathbf{\Sigma}^\rho\mathbf{V}^\mathsf{H}$, where $\mathbf{V}$ is the matrix of eigenvectors, and $\mathbf{\Sigma}$ is the matrix of eigenvalues of $\mathbf{A}$. In the case of $\mathbf{A}$, the GFT $\hat{\mathbf{x}} = \mathbf{V}^\mathsf{H}\mathbf{x}$.

For $\rho = 1$ in (1) we have $\|\mathbf{x}\|^2_{\rho,\epsilon} = \mathbf{x}^\mathsf{T}(\mathbf{L} + \epsilon\mathbf{I})\mathbf{x}$. The term $(\mathbf{L} + \epsilon\mathbf{I})$ is associated with a better condition number[2] than using $\mathbf{L}$ alone. For example, better condition numbers are associated with faster convergence rates in gradient descent methods as shown in [16]. For the Laplacian matrix $\mathbf{L}$, we know that $\kappa(\mathbf{L}) = \frac{|\lambda_{\max}(\mathbf{L})|}{|\lambda_{\min}(\mathbf{L})|} \approx \frac{\lambda_{\max}(\mathbf{L})}{0} \to \infty$, where $\kappa(\mathbf{L})$ is the condition number of $\mathbf{L}$, $\lambda_{\max}(\mathbf{L})$ is the maximum eigenvalue, and $\lambda_{\min}(\mathbf{L})$ is the minimum eigenvalue of $\mathbf{L}$. Since $\kappa(\mathbf{L}) \to \infty$, we have an ill-conditioned problem when relying on the Laplacian matrix alone. On the other hand, for the Sobolev term, we have that $\mathbf{L} + \epsilon\mathbf{I} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\mathsf{T} + \epsilon\mathbf{I} = \mathbf{U}(\mathbf{\Lambda} + \epsilon\mathbf{I})\mathbf{U}^\mathsf{T}$. Therefore, $\lambda_{\min}(\mathbf{L} + \epsilon\mathbf{I}) = \epsilon$, *i.e.*, $\mathbf{L} + \epsilon\mathbf{I}$ is positive definite ($\mathbf{L} + \epsilon\mathbf{I} \succ 0$) for $\epsilon > 0$, and:

$$\kappa(\mathbf{L}+\epsilon\mathbf{I}) = \frac{|\lambda_{\max}(\mathbf{L} + \epsilon\mathbf{I})|}{|\lambda_{\min}(\mathbf{L} + \epsilon\mathbf{I})|} = \frac{\lambda_{\max}(\mathbf{L}) + \epsilon}{\epsilon} < \kappa(\mathbf{L}) \, \forall \, \epsilon > 0. \tag{3}$$

Namely, $\mathbf{L}+\epsilon\mathbf{I}$ has a better condition number than $\mathbf{L}$. It might not be evident why a better condition number could help in GNNs, where the inverses of the Laplacian or adjacency matrices are not required to perform the propagation rules. However, some studies have indicated the adverse effects of badbehaved matrices. For example, Kipf and Welling [2] used a renormalization trick $(\mathbf{A} + \mathbf{I})$ in their filtering operation to avoid exploding/vanishing gradients. Similarly, Wu *et al*. [26] showed that adding the identity matrix to $\mathbf{A}$ shrinks the graph spectral domain, resulting in a low-pass-type filter.

The previous theoretical analysis shows the benefits of the Sobolev norm about 1) the diverse frequencies computation in (2), and 2) the better condition number in (3).

2.3. Sparse Sobolev Norm

The use of $\mathbf{L}$ or $\mathbf{A}$ in GNNs is computationally efficient because these matrices are usually sparse. Therefore, we can perform a small number of sparse matrix operations. For the Sobolev norm, the term $(\mathbf{L}+\epsilon\mathbf{I})^\rho$ can quickly become a dense matrix for large values of ρ, leading to scalability and memory problems. To mitigate this limitation, we use a sparse Sobolev norm to keep the same sparsity level.

Definition 2. *Let $\mathbf{L} \in \mathbb{R}^{N\times N}$ be the Laplacian matrix of G. For fixed parameters $\epsilon \geq 0$ and $\rho \in \mathbb{N}$, the sparse Sobolev term for GNNs is introduced as the ρ Hadamard multiplications of $(\mathbf{L} + \epsilon\mathbf{I})$ (the Hadamard power) such that:*

$$(\mathbf{L} + \epsilon\mathbf{I})^{(\rho)} = (\mathbf{L} + \epsilon\mathbf{I}) \circ (\mathbf{L} + \epsilon\mathbf{I}) \circ \cdots \circ (\mathbf{L} + \epsilon\mathbf{I}). \tag{4}$$

For example, $(\mathbf{L} + \epsilon\mathbf{I})^{(2)} = (\mathbf{L} + \epsilon\mathbf{I}) \circ (\mathbf{L} + \epsilon\mathbf{I})$. Thus, the sparse Sobolev norm is given by:

$$\|\mathbf{x}\|_{(\rho),\epsilon} \triangleq \|(\mathbf{L} + \epsilon\mathbf{I})^{(\rho/2)}\mathbf{x}\|. \tag{5}$$

Let $\langle \mathbf{x}, \mathbf{y} \rangle_{(\rho),\epsilon} = \mathbf{x}^\mathsf{T}(\mathbf{L} + \epsilon\mathbf{I})^{(\rho)}\mathbf{y}$ be the inner product between two graph signals $\mathbf{x}$ and $\mathbf{y}$ that induces the associated sparse Sobolev norm. We can easily prove that the sparse Sobolev norm $\|\mathbf{x}\|_{(\rho),\epsilon} \triangleq \|(\mathbf{L}+\epsilon\mathbf{I})^{(\rho/2)}\mathbf{x}\|$ satisfies the basic properties of vector norms[3] for $\epsilon > 0$ (for $\epsilon = 0$ we have a semi-norm). For the positive definiteness property, we need the Schur product theorem [18].

[1] $\lambda_N \leq 2$ in the case of the symmetric normalized Laplacian $\mathbf{\Delta}$.

[2] The condition number $\kappa(\mathbf{L})$ associated with the square matrix $\mathbf{L}$ is a measure of how well or ill-conditioned is the inversion of $\mathbf{L}$.

[3] We omit the proof due to space limitation.

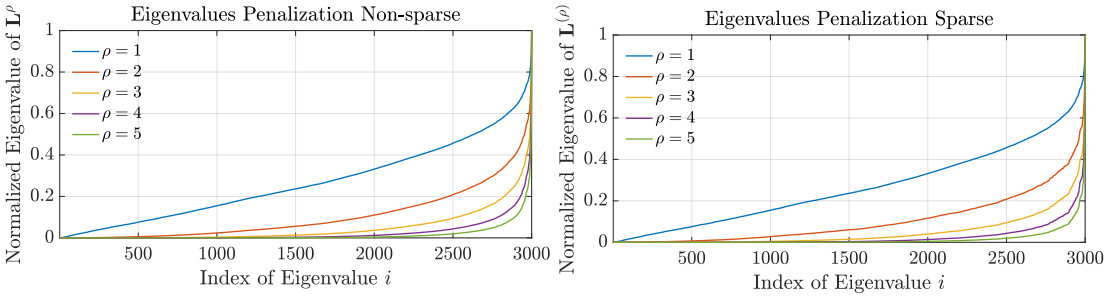

Fig. 1. Eigenvalues penalization for the non-sparse and sparse matrix multiplications of the combinatorial Laplacian matrix.

The sparse Sobolev term in (4) has the property of keeping the same sparsity level for any ρ. Notice that $(\mathbf{L}+\epsilon\mathbf{I})^{\rho}$ is equal to the sparse Sobolev term if 1) we restrict ρ to be in $\mathbb{N}$, and 2) we replace the matrix multiplication by the Hadamard product. The theoretical properties of the Sobolev norm in (2) and (3) do not extend trivially to its sparse counterpart. However, we can develop some theoretical insights using concepts of Kronecker products and the Schur product theorem [18].

Theorem 1. *Let* $\mathbf{L}$ *be any Laplacian matrix of a graph with eigenvalue decomposition* $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathsf{T}}$*, we have that:*

$$\mathbf{L} \circ \mathbf{L} = \mathbf{L}^{(2)} = \mathbf{P}_N^{\mathsf{T}}(\mathbf{U} \otimes \mathbf{U})(\mathbf{\Lambda} \otimes \mathbf{\Lambda})(\mathbf{U}^{\mathsf{T}} \otimes \mathbf{U}^{\mathsf{T}})\mathbf{P}_N, \quad (6)$$

where $\mathbf{P}_N \in \{0,1\}^{N^2 \times N}$ *is a partial permutation matrix.*

Proof. For the spectral decomposition, we have that:

$$\mathbf{L} \otimes \mathbf{L} = (\mathbf{U} \otimes \mathbf{U})(\mathbf{\Lambda} \otimes \mathbf{\Lambda})(\mathbf{U}^{\mathsf{T}} \otimes \mathbf{U}^{\mathsf{T}}), \quad (7)$$

where we used the property of Kronecker products $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ [18]. Similarly, we know that $\mathbf{S} \circ \mathbf{T} = \mathbf{P}_n^{\mathsf{T}}(\mathbf{S} \otimes \mathbf{T})\mathbf{P}_m$, where $\mathbf{S}, \mathbf{T} \in \mathbb{R}^{n \times m}$, and $\mathbf{P}_n \in \{0,1\}^{n^2 \times n}, \mathbf{P}_m \in \{0,1\}^{m^2 \times m}$ are partial permutation matrices. If $\mathbf{S}, \mathbf{T} \in \mathbb{R}^{n \times n}$ are square matrices, we have that $\mathbf{S} \circ \mathbf{T} = \mathbf{P}_n^{\mathsf{T}}(\mathbf{S} \otimes \mathbf{T})\mathbf{P}_n$ (Theorem 1 in [27]). We can then get a general form of the spectrum of the Hadamard product for $\rho = 2$ using (7) and Theorem 1 in [27] as follows: $\mathbf{L} \circ \mathbf{L} = \mathbf{L}^{(2)} = \mathbf{P}_N^{\mathsf{T}}(\mathbf{U} \otimes \mathbf{U})(\mathbf{\Lambda} \otimes \mathbf{\Lambda})(\mathbf{U}^{\mathsf{T}} \otimes \mathbf{U}^{\mathsf{T}})\mathbf{P}_N$. □

Eq. (6) is a closed-form solution regarding the spectrum of the Hadamard power for $\rho = 2$. Thus, the spectrum of the Hadamard multiplication is a compressed form of the Kronecker product of its spectral components. The sparse Sobolev term we use in our S-SobGNN is given by $(\mathbf{L}+\epsilon\mathbf{I})^{(\rho)}$ so that the spectral components of the graph are changing for each value of ρ as shown in (6).

For the condition number of the Hadamard powers, we can use the Schur product theorem [18]. We know that $(\mathbf{L} + \epsilon\mathbf{I})^{(\rho)} \succ 0 \; \forall \, \epsilon > 0$ since $(\mathbf{L} + \epsilon\mathbf{I}) \succ 0 \; \forall \, \epsilon > 0$, and therefore $\kappa((\mathbf{L} + \epsilon\mathbf{I})^{(\rho)}) < \infty$. For the adjacency matrix, the eigenvalues of $\mathbf{A}$ lie into $[-d, d]$, where d is the maximal degree of G [28]. Therefore, we can bound the eigenvalues of $\mathbf{A}$ into $[-1, 1]$ by normalizing $\mathbf{A}$ such that $\mathbf{A}_N = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$. As a result, we know that $\mathbf{A}_N + \epsilon\mathbf{I} \succ 0 \; \forall \; \epsilon > 1$, and $(\mathbf{A}_N + \epsilon\mathbf{I})^{(\rho)} \succ 0 \; \forall \; \epsilon > 1$. We can say that the theoretical developments of the sparse Sobolev norm hold to some extent the same developments of Section 2.2, *i.e.*, a more diverse set of frequencies and a better condition number. Figure 1 shows five normalized eigenvalue penalizations for $\mathbf{L}^{\rho}$ (non-sparse) and $\mathbf{L}^{(\rho)}$ (sparse). We notice that the normalized spectrum of $\mathbf{L}^{\rho}$ and $\mathbf{L}^{(\rho)}$ are very similar. Finally, we should work with weighted graphs when using the adjacency matrix since $\mathbf{A}^{(\rho)} = \mathbf{A} \; \forall \, \rho \in \mathbb{N}$ for unweighted graphs.

2.4. Graph Neural Network Architecture

Kipf and Welling [2] proposed one of the most successful yet simple GNN, called Graph Convolutional Networks (GCNs):

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \quad (8)$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$, $\mathbf{H}^{(l)}$ is the matrix of activations in layer l such that $\mathbf{H}^{(1)} = \mathbf{X}$ (data matrix), $\mathbf{W}^{(l)}$ is the matrix of trainable weights in layer l, and $\sigma(\cdot)$ is an activation function. The motivation of the propagation rule in (8) comes from the first-order approximation of localized spectral filters on graphs [1]. Kipf and Welling [2] used (8) to propose the vanilla GCN, which is composed of two graph convolutional layers as in (8). The first activation function is a Rectified Linear Unit ($\mathrm{ReLU}(\cdot) = \max(0, \cdot)$), and the final activation function is a softmax applied row-wise such that $\mathrm{softmax}(x_i) = \frac{1}{Q}\exp(x_i)$ where $Q = \sum_i \exp(x_i)$. Finally, the vanilla GCN uses cross-entropy as a loss function.

We introduce a new filtering operation based on the sparse Sobolev term where our propagation rule is such that:

$$\mathbf{B}_{\rho}^{(l+1)} = \sigma(\bar{\mathbf{D}}_{\rho}^{-\frac{1}{2}}\bar{\mathbf{A}}_{\rho}\bar{\mathbf{D}}_{\rho}^{-\frac{1}{2}}\bar{\mathbf{H}}^{(l)}\mathbf{W}_{\rho}^{(l)}), \quad (9)$$

where $\bar{\mathbf{A}}_{\rho} = (\mathbf{A} + \epsilon\mathbf{I})^{(\rho)}$ is the ρth sparse Sobolev term of $\mathbf{A}$, $\bar{\mathbf{D}}_{\rho}$ is the degree matrix of $\bar{\mathbf{A}}_{\rho}$, and $\bar{\mathbf{H}}^{(1)} = \mathbf{X}$. Notice that $\bar{\mathbf{A}}_{\rho} = \tilde{\mathbf{A}}$ when $\epsilon = 1$, and $\rho = 1$, *i.e.*, our propagation rule is a generalization of the GCN model. S-SobGNN computes a cascade of propagation rules as in (9) with several values of ρ in the set $\{1, 2, \ldots, \alpha\}$, and therefore a linear

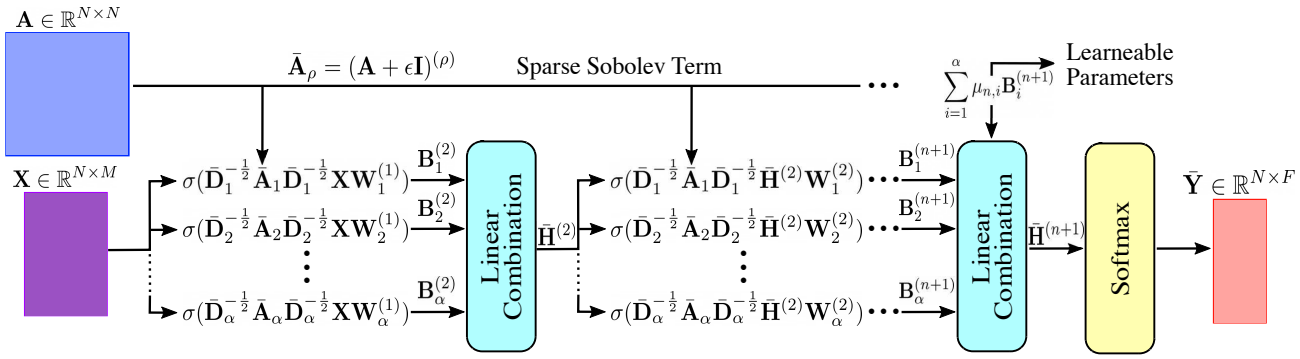

Fig. 2. Basic configuration of our S-SobGNN architecture with n layers and α filters per layer.

combination layer weights the outputs of each filter. Figure 2 shows the basic configuration of S-SobGNN. Notice that our graph convolution is efficiently computed since the term $\bar{\mathbf{D}}_\rho^{-\frac{1}{2}}\bar{\mathbf{A}}_\rho\bar{\mathbf{D}}_\rho^{-\frac{1}{2}}$ $\forall\, \rho \in \{1, 2, \ldots, \alpha\}$ is the same in all layers (so we can compute it offline), and also, these terms are sparse for any value of ρ (given that $\mathbf{A}$ is also sparse). S-SobGNN uses ReLU as the activation function for each filter, $\mathrm{softmax}$ at the end of the network, and the cross-entropy loss function. The basic configuration of S-SobGNN is defined by the number of filters α in each layer, the parameter ϵ, the number of hidden units of each $\mathbf{W}_\rho^{(l)}$, and the number of layers n. When we construct weighted graphs with Gaussian kernels, the weights of the edges are in the interval $[0, 1]$. As a consequence, large values of ρ could make $\bar{\mathbf{A}}_\rho = \mathbf{0}$, and the diagonal elements of $\bar{\mathbf{D}}_\rho^{-\frac{1}{2}}$ could become ∞. Similarly, large values of α make very wide architectures with a high parameter budget, so it is desirable to maintain a reasonable value for α. The computational complexity of S-SobGNN is $\mathcal{O}(n\alpha|\mathcal{E}|+n\alpha)$. For comparison, the computational complexity of a n-layers GCN is $\mathcal{O}(n|\mathcal{E}|)$. The exact complexity of both methods also depends on the feature dimension, the hidden units, and the number of nodes in the graph, which we omit for simplicity.

3. EXPERIMENTS AND RESULTS

S-SobGNN is compared to eight GNN architectures: Chebyshev filters (Cheby) [1], GCN [2], GAT [3], SIGN [14], SGC [26], ClusterGCN [29], SuperGAT [30], and Transformers [31]. We test S-SobGNN in several semi-supervised learning tasks including, cancer detection in images [20], text classification of news (20News) [20], Human Activity Recognition using sensors (HAR) [21], and recognition of isolated spoken letters (Isolet) [22]. We frame the semi-supervised learning problem as a node classification task in graphs, where we construct the graphs with a k-Nearest Neighbors (k-NN) method and a Gaussian kernel with $k = 30$. We split the data into train/validation/test sets with $10\%/45\%/45\%$, where we first divide the data into a development set and a test set. This is done once to avoid using the test set in the hyperparameter optimization. We tune the hyperparameters of each GNN with a random search with 100 repetitions and five different seeds for the validation set. We report average accuracies on the test set using 50 different seeds with 95% confidence intervals calculated by bootstrapping with $1,000$ samples. Table 1 shows the experimental results. S-SobGNN shows the best performance against state-of-the-art methods.

Table 1. Accuracy (in %) for the baseline methods and our S-SobGNN algorithm in four datasets for semi-supervised learning, inferring the graphs with a k-NN method.

Model	Cancer	20News	HAR	Isolet
Cheby [1]	87.55 ± 3.91	70.36 ± 1.14	73.14 ± 7.01	69.70 ± 1.47
GCN [2]	76.71 ± 4.47	51.76 ± 2.11	66.26 ± 4.91	55.55 ± 2.72
GAT [3]	73.51 ± 4.87	48.72 ± 2.21	59.13 ± 6.30	60.00 ± 2.21
SIGN [14]	***89.55*** $\pm$ ***0.38***	***71.79*** $\pm$ ***0.25***	***90.98*** $\pm$ ***0.25***	***84.02*** $\pm$ ***0.30***
SGC [26]	72.80 ± 4.71	54.68 ± 1.99	42.19 ± 3.44	41.55 ± 1.91
ClusterGCN [29]	74.45 ± 5.37	60.56 ± 2.18	57.70 ± 5.48	63.99 ± 2.24
SuperGAT [30]	70.56 ± 5.14	57.52 ± 1.93	56.04 ± 5.32	58.49 ± 2.27
Transformer [31]	71.10 ± 5.45	57.48 ± 2.29	66.01 ± 6.10	66.24 ± 2.39
S-SobGNN (ours)	**93.11** $\pm$ **0.45**	**72.18** $\pm$ **0.32**	**92.85** $\pm$ **0.58**	**86.17** $\pm$ **0.34**

The best and second-best performing methods on each dataset are shown in **red** and ***blue***, respectively.

4. CONCLUSIONS

In this work, we extended the concept of Sobolev norms using the Hadamard product between matrices to keep the sparsity level of the graph representations. We introduced a new Sparse GNN architecture using the proposed sparse Sobolev norm. Similarly, certain theoretical notions of our filtering operation were provided in Sections 2.2 and 2.3. Finally, S-SobGNN outperformed several methods of the literature in four semi-supervised learning tasks.

Acknowledgments: This work was supported by the DATAIA Institute as part of the "Programme d'Investissement d'Avenir", (ANR-17-CONV-0003) operated by CentraleSupélec, and by ANR (French National Research Agency) under the JCJC project GraphIA (ANR-20-CE23-0009-01).

5. REFERENCES

[1] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016. 1, 3, 4

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017. 1, 2, 3, 4

[3] P. Veličković et al., "Graph attention networks," in *ICLR*, 2018. 1, 4

[4] V. N. Ioannidis, A. G. Marques, and G. B. Giannakis, "A recurrent graph neural network for multi-relational data," in *IEEE ICASSP*, 2019. 1

[5] Z. Zhao et al., "Distributed scheduling using graph neural networks," in *IEEE ICASSP*, 2021. 1

[6] S. Pfrommer, A. Ribeiro, and F. Gama, "Discriminability of single-layer graph neural networks," in *IEEE ICASSP*, 2021. 1

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. 1

[8] A. Duval and F. D. Malliaros, "Higher-order clustering and pooling for graph neural networks," in *ACM CIKM*, 2022. 1

[9] G. Li et al., "DeepGCNs: Can GCNs go as deep as CNNs?," in *IEEE ICCV*, 2019. 1

[10] A. Benamira et al., "Semi-supervised learning and graph neural networks for fake news detection," in *IEEE/ACM ASONAM*, 2019. 1

[11] P. Gainza et al., "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning," *Nat. Methods*, vol. 17, no. 2, pp. 184–192, 2020. 1

[12] H. E. Egilmez, Y. H. Chao, and A. Ortega, "Graph-based transforms for video coding," *IEEE T-IP*, vol. 29, pp. 9330–9344, 2020. 1

[13] J. H. Giraldo, S. Javed, and T. Bouwmans, "Graph moving object segmentation," *IEEE T-PAMI*, vol. 44, no. 5, pp. 2485–2503, 2022. 1, 2

[14] F. Frasca et al., "SIGN: Scalable inception graph neural networks," in *ICML-W*, 2020. 1, 4

[15] I. Pesenson, "Variational splines and Paley–Wiener spaces on combinatorial graphs," *Constructive Approximation*, vol. 29, no. 1, pp. 1–21, 2009. 1, 2

[16] J. H. Giraldo et al., "Reconstruction of time-varying graph signals via Sobolev smoothness," *IEEE T-SIPN*, vol. 8, pp. 201–214, 2022. 1, 2

[17] F. R. K. Chung, *Spectral graph theory*, Number 92. American Mathematical Society, 1997. 1

[18] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge university press, 2012. 1, 2, 3

[19] J. N. Kather et al., "Multi-class texture analysis in colorectal cancer histology," *Scientific Reports*, vol. 6, pp. 27988, 2016. 1

[20] K. Lang, "NewsWeeder: Learning to filter netnews," in *JMLR*, 1995. 1, 4

[21] D. Anguita et al., "A public domain dataset for human activity recognition using smartphones," in *ESANN*, 2013. 1, 4

[22] M. Fanty and R. Cole, "Spoken letter recognition," in *NeurIPS*, 1991. 1, 4

[23] A. Ortega et al., "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018. 2

[24] J. H Giraldo and T. Bouwmans, "GraphBGS: Background subtraction via recovery of graph signals," in *ICPR*, 2020. 2

[25] J. H. Giraldo and T. Bouwmans, "On the minimization of Sobolev norms of time-varying graph signals: Estimation of new Coronavirus disease 2019 cases," in *IEEE MLSP*, 2020. 2

[26] F. Wu et al., "Simplifying graph convolutional networks," in *ICML*, 2019. 2, 4

[27] G. Visick, "A quantitative version of the observation that the Hadamard product is a principal submatrix of the Kronecker product," *LAA*, vol. 304, no. 1-3, pp. 45–68, 2000. 3

[28] B. Nica, *A brief introduction to spectral graph theory*, European Mathematical Society, 2018. 3

[29] W.L. Chiang et al., "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *ACM SIGKDD*, 2019. 4

[30] D. Kim and A. Oh, "How to find your friendly neighborhood: Graph attention design with self-supervision," in *ICLR*, 2021. 4

[31] Y. Shi et al., "Masked label prediction: Unified message passing model for semi-supervised classification," in *IJCAI*, 2021. 4