A UNIFIED FRAMEWORK FOR CONVOLUTION-BASED GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Convolutional Networks (GCNs) have attracted a lot of research interest in the machine learning community in recent years. Although many variants have been proposed, we still lack a systematic view of different GCN models and deep understanding of the relations among them. In this paper, we take a step forward to establish a unified framework for convolution-based graph neural networks, by formulating the basic graph convolution operation as an optimization problem in the graph Fourier space. Under this framework, a variety of popular GCN models, including the vanilla-GCNs, attention-based GCNs and topology-based GCNs, can be interpreted as a same optimization problem but with different carefully designed regularizers. This novel perspective enables a better understanding of the similarities and differences among many widely used GCNs, and may inspire new approaches for designing better models. As a showcase, we also present a novel regularization technique under the proposed framework to tackle the oversmoothing problem in graph convolution. The effectiveness of the newly designed model is validated empirically.

1 INTRODUCTION

Recent years have witnessed a fast development in graph processing by generalizing convolution operation to graph-structured data, which is known as Graph Convolutional Networks (GCNs) (Kipf & Welling, 2017). Due to the great success, numerous variants of GCNs have been developed and extensively adopted in the field of social network analysis (Hamilton et al., 2017; Wu et al., 2019a; Veličković et al., 2018), biology (Zitnik et al., 2018), transportation forecasting (Li et al., 2017) and natural language processing (Wu et al., 2019b; Yao et al., 2019). Plenty of researches aim to achieve higher expressiveness by exploring higher-order information or introducing additional learning mechanisms like attention modules.

Although these approaches are proved effective on specific tasks, it is still unclear about their actual influences on the learned representations. Despite that some researches try to demonstrate their effectiveness by providing experimental results under specific settings or visualization results like t-SNE (Maaten & Hinton, 2008), these evaluations are predominantly qualitative. As a consequence, we still lack a systematic view of different GCN models and deep understanding of the relations among them.

To tackle the aforementioned problem, we resort to the techniques in graph signal processing and attempt to interpret GCN-based approaches from a general perspective. Specifically, in this paper, we present a unified graph convolution framework by establishing graph convolution operations with optimization problems in the graph Fourier domain. We consider a Laplacian regularized least squares optimization problem and show that most of the convolution-based approaches can be interpreted in this framework by adding carefully designed regularizers. Besides vanilla GCNs, we also extend our framework to formulating non-convolutional operations (Xu et al., 2018a; Hamilton et al., 2017), attention-based GCNs (Veličković et al., 2018; Thekumparampil et al., 2018) and topology-based GCNs (Klicpera et al., 2018; Kapoor et al., 2019), which cover a high fraction of the state-of-the-art graph learning approaches. This novel perspective provides a re-interpretation of graph convolution operations and enables a better understanding of the similarities and differences among many widely used GCNs, and may inspire new approaches for designing better models.

For conclusion, our approach leads to three potential contributions that bring clarity to the field of graph representation learning:

1. We introduce a novel unified framework for convolution-based graph neural networks and interpret various convolution filters as carefully designed regularizers in the graph Fourier domain, which provides a general methodology for evaluating and relating different graph learning modules.

2. Instead of the feature learning space that most researches investigated in, our approach provides new insights on tackling common problems and improving the generalization performance of current graph neural networks in the graph Fourier domain. As a showcase, we present a novel regularization technique under the proposed framework to tackle the oversmoothing problem in graph representation learning. Extensive experiments are conducted on both transductive and inductive settings, showing the effectiveness of the proposed method.

3. By establishing the unified framework, we provide a once-for-all platform for handy modifications on convolution-based approaches. As shown in section 3, the newly designed regularizer can be easily implemented on other convolution-based networks as a plug-in module with trivial adaptations. We believe that our platform can provide convenience for designing new graph learning modules and searching for better combinations.

2 Unified Graph Convolution Framework

With the success of GCNs, a wide variety of convolution-based graph learning approaches are proposed which progressively enhance the expressive power and generalization performance of graph neural networks. Despite the effectiveness of GCN and its derivatives on specific tasks, there is no general methodology for quantitatively understanding and relating different graph learning modules. To address this problem, we consider interpreting convolution-based approaches from the graph signal processing perspective. Formally, in this work, we establish the connections between graph convolution operations and optimization problems in graph Fourier space, showing the effectiveness of each module explicitly with specific regularizers.

2.1 Preliminary

We start with an overview of the basic concepts of graph signal processing. Let $\mathcal{G} = (\mathbb{V}, \boldsymbol{A})$ denote a graph with node feature vectors where $\mathbb{V}$ represents the vertex set consisting of nodes $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N\}$ and $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{N\times N}$ is the adjacency matrix implying the connectivity between nodes in the graph. $\boldsymbol{D} = \mathrm{diag}(d(1), \ldots, d(N)) \in \mathbb{R}^{N\times N}$ is the degree matrix of $\boldsymbol{A}$ where $d(i) = \sum_{j\in\mathbb{V}} a_{ij}$ is the degree of vertex i. $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is the combinatorial Laplacian and $\tilde{\boldsymbol{L}} = \boldsymbol{I} - \boldsymbol{D}^{(-1/2)}\boldsymbol{A}\boldsymbol{D}^{(-1/2)}$ is the normalized Laplacian of $\mathcal{G}$. Additionally, we let $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ and $\tilde{\boldsymbol{D}} = \boldsymbol{D} + \boldsymbol{I}$ denote the augmented adjacency and degree matrices with added self-loops. Then $\tilde{\mathcal{L}}_{sym} = \boldsymbol{I} - \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$ and $\tilde{\mathcal{L}}_{rw} = \boldsymbol{I} - \tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{A}}$ are the augmented symmetric normalized and random walk normalized Laplacian of $\mathcal{G}$. $\tilde{\mathcal{A}}_{\mathrm{sym}} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$ and $\tilde{\mathcal{A}}_{\mathrm{rw}} = \tilde{\boldsymbol{D}}^{-1}\tilde{\boldsymbol{A}}$ are the symmetric normalized and the random walk normalized augmented adjacency matrices.

Let $\boldsymbol{x} \in \mathbb{R}^N$ be a signal on the vertices of the graph. The spectral convolution is defined as a function of a filter g_θ parameterized in the Fourier domain (Kipf & Welling, 2017):

$$g_\theta \star \boldsymbol{x} = \boldsymbol{U} g_\theta(\Lambda)\boldsymbol{U}^T\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{U}$ and Λ are the eigenvectors and eigenvalues of the normalized Laplacian $\tilde{\boldsymbol{L}}$. Also, we follow Hoang & Maehara (2019) and define the variation Δ and $\tilde{\boldsymbol{D}}$-inner product as:

$$\Delta(\boldsymbol{x}) = \sum_{i,j\in\mathbb{V}} a_{ij}(\boldsymbol{x}(i) - \boldsymbol{x}(j))^2 = \boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}, \quad (\boldsymbol{x}, \boldsymbol{y})_{\tilde{\boldsymbol{D}}} = \sum_{i\in\mathbb{V}}(d(i)+1)\boldsymbol{x}(i)\boldsymbol{y}(i) = \boldsymbol{x}^T\tilde{\boldsymbol{D}}\boldsymbol{y}, \tag{2}$$

which specifies the smoothness and importance of the signal respectively.

2.2 Unified Graph Convolution Framework

Several researches have proved that, in the territory of graph signal processing, the representative features are mostly contained in the low-frequency signals while noises are preserved in the high-frequency signals (Hoang & Maehara, 2019). Based on this observation, numerous graph representation learning methods are designed to decrease the high-frequency components, which can be viewed as low-pass filters in the graph Fourier space. With similar inspiration, we consider a Laplacian regularized least squares optimization problem with graph signal regularizers and attempt to build connections with these filters.

Definition 1 Unified Graph Convolution Framework. *Graph convolution filters can be achieved by solving the following Laplacian regularized least squares optimization:*

$$\min_{\bar{\boldsymbol{X}}} \sum_{i \in \mathbb{V}} \|\bar{\boldsymbol{x}}(i) - \boldsymbol{x}(i)\|_{\tilde{\boldsymbol{D}}}^2 + \lambda \mathcal{L}_{\text{reg}}, \tag{3}$$

where $\|\boldsymbol{x}\|_{\tilde{\boldsymbol{D}}} = \sqrt{(\boldsymbol{x}, \boldsymbol{x})_{\tilde{\boldsymbol{D}}}}$ *denotes the norm induced by* $\tilde{\boldsymbol{D}}$.

In the following sections, we will show that a wide range of convolution-based graph neural networks can be derived from Definition 1 with different carefully designed regularizers, providing new insights on understanding different graph learning modules from the graph signal perspective.

2.2.1 Graph Convolution Networks

Graph Convolution Networks (GCNs) (Kipf & Welling, 2017) are the foundation of numerous graph learning models and have earned plenty of research interests. Several researches have demonstrated that the vanilla GCN is essentially a type of Laplacian smoothing over the whole graph, which makes the features of the connected nodes similar. Therefore, to reformulate GCNs in the graph Fourier space, we consider utilizing the variation $\Delta(\boldsymbol{x})$ as the regularizer.

Definition 2 Vanilla GCNs. *Let* $\bar{\boldsymbol{x}}(i)_{i \in \mathbb{V}}$ *be the estimation of the input observation* $\boldsymbol{x}(i)_{i \in \mathbb{V}}$. *A low-pass filter:*

$$\bar{\boldsymbol{X}} = \tilde{\mathcal{A}}_{\text{rw}} \boldsymbol{X}, \tag{4}$$

is the first-order approximation of *the optimal solution of the following optimization:*

$$\min_{\bar{\boldsymbol{X}}} \sum_{i \in \mathbb{V}} \|\bar{\boldsymbol{x}}(i) - \boldsymbol{x}(i)\|_{\tilde{\boldsymbol{D}}}^2 + \sum_{i,j \in \mathbb{V}} a_{ij} \|\bar{\boldsymbol{x}}(i) - \bar{\boldsymbol{x}}(j)\|_2^2. \tag{5}$$

Derivations of the definitions are presented in Appendix **??**.

As the eigenvalues of the approximated filter $\tilde{\mathcal{A}}_{\text{rw}}$ are bounded by 1, it resembles a low-pass filter that removes the high-frequency signals. By exchanging $\tilde{\mathcal{A}}_{\text{rw}}$ with $\tilde{\mathcal{A}}_{\text{sym}}$ (which has the same eigenvalues as $\tilde{\mathcal{A}}_{\text{rw}}$), we obtain the same formulation adopted in GCNs.

It has been stated that the second term $\Delta(\boldsymbol{x})$ in Equation 5 measures the variation of the estimation $\bar{\boldsymbol{x}}$ over the graph structure. By adding this regularizer to the objective function, the obtained filter emphasizes the low-frequency signals through minimizing the variation over the local graph structure, while keeping the estimation close to the input in the graph Fourier space.

2.2.2 Non-convolutional Operations

Residual Connection. Residual connection is first proposed by He et al. (2016) and has been widely adopted in graph representation learning approaches. In the vanilla GCNs, norms of the eigenvalues of the filter $\tilde{\mathcal{A}}_{\text{rw}}$ (or $\tilde{\mathcal{A}}_{\text{sym}}$) are bounded by 1 which ensure numerical stability in the training procedure. However, on the other hand, the signal in all frequency band will shrink as the convolution layer stacks, leading to a consistent information loss. Therefore, adding the residual connection is deemed to preserve the strength of the input signal.

Definition 3 Residual Connection. *A graph convolution filter with residual connection:*

$$\bar{\boldsymbol{X}} = \tilde{\mathcal{A}}_{\text{rw}} \boldsymbol{X} + \epsilon \boldsymbol{X}, \tag{6}$$

where $\epsilon > 0$ *controls the strength of residual connection,* ***is the first-order approximation of*** *the optimal solution of the following optimization:*

$$\min_{\bar{\boldsymbol{X}}} \sum_{i \in \mathbb{V}} (\|\bar{\boldsymbol{x}}(i) - \boldsymbol{x}(i)\|_{\tilde{\boldsymbol{D}}}^2 - \epsilon \|\bar{\boldsymbol{x}}(i)\|_{\tilde{\boldsymbol{D}}}^2) + \sum_{i,j \in \mathbb{V}} a_{ij} \|\bar{\boldsymbol{x}}(i) - \bar{\boldsymbol{x}}(j)\|_2^2. \tag{7}$$

By adding the negative regularizer to penalize the estimations with small norms, we can induce the same formulation as the vanilla graph convolution with residual connection.

Concatenation. Concatenation is practically a residual connection with different learning weights.
Definition 3' Concatenation. *A graph convolution filter concatenating with the input signal:*

$$\bar{\boldsymbol{X}} = \tilde{\mathcal{A}}_{\text{rw}} \boldsymbol{X} + \epsilon \boldsymbol{X} \Theta \Theta^T, \tag{8}$$

is the first-order approximation of *the optimal solution of the following optimization:*

$$\min_{\bar{\boldsymbol{X}}} \sum_{i\in\mathbb{V}} (\|\bar{\boldsymbol{x}}(i)-\boldsymbol{x}(i)\|^2_{\tilde{\boldsymbol{D}}} - \epsilon\|\bar{\boldsymbol{x}}(i)\Theta\|^2_{\tilde{\boldsymbol{D}}}) + \sum_{i,j\in\mathbb{V}} a_{ij}\|\bar{\boldsymbol{x}}(i)-\bar{\boldsymbol{x}}(j)\|^2_2, \tag{9}$$

where $\epsilon > 0$ controls the strength of concatenation and Θ is the learning coefficient.

Although the learning weights $\Theta\Theta^T$ has a constrained expressive capability, it can be compensated by the following feature learning modules.

2.2.3 Attention-based Convolutional Networks

Since the convolution filters in GCNs are dependent only on the graph structure, GCNs are proved to have restricted expressive power and may cause the oversmoothing problem. Several researches try to introduce the attention mechanism to the convolution filter, learning to assign different edge weights at each layer based on nodes and edges. GAT (Veličković et al., 2018) and AGNN (Thekumparampil et al., 2018) computes the attention coefficients as a function of the features of connected nodes, while ECC (Simonovsky & Komodakis, 2017) and GatedGCN (Bresson & Laurent, 2017) consider the activations for each connected edges. Although these approaches have different insights, they can be all formulated as (See details in Appendix **??**):

$$\mathrm{att}_{ij} = a_{ij} f_\theta(\boldsymbol{x}(i), \boldsymbol{x}(j), e_{ij}), \;\; i,j\in\mathbb{V}, \tag{10}$$

where e_{ij} denotes the edge representation if applicable. Therefore, we replace a_{ij} in Definition 2 with learned coefficients to enforce different regularization strength on the connected edges.

Definition 4 Attention-based GCNs. *An attention-based graph convolution filter:*

$$\bar{\boldsymbol{X}} = \mathrm{Att}\boldsymbol{X}, \tag{11}$$

is the first-order approximation of *the optimal solution of the following optimization:*

$$\min_{\bar{\boldsymbol{X}}} \sum_{i\in\mathbb{V}} \|\bar{\boldsymbol{x}}(i)-\boldsymbol{x}(i)\|^2_{\tilde{\boldsymbol{D}}} + \sum_{i,j\in\mathbb{V}} \mathrm{att}_{ij}\|\bar{\boldsymbol{x}}(i)-\bar{\boldsymbol{x}}(j)\|^2_2, \qquad s.t. \sum_{j\in\mathbb{V}} \mathrm{att}_{ij} = \tilde{\boldsymbol{D}}_{ii}, \forall i\in\mathbb{V}. \tag{12}$$

Notice that we use a *normalization trick* to constrain the degree of the attention matrix to be the same as the original degree matrix $\tilde{\boldsymbol{D}}$ as we want to preserve the strength of the regularization for each node. The formulated filter Att corresponds to the matrix $\tilde{\boldsymbol{D}}^{-1}\mathrm{att}$ with row sum equals to 1, which is also consistent with most of the attention-based approaches after normalization. Through adjusting the regularization strength for edges, nodes with higher attention coefficients tend to have similar features while the distance for nodes with low attention coefficients will be further.

2.2.4 Topology-based Convolution Networks

Attention-based approaches are mostly designed based on the local structure. Besides focusing on the first-order adjacency matrix, several approaches (Klicpera et al., 2018; 2019; Kapoor et al., 2019; Du et al., 2017) propose to adopt the structural information in the multi-hop neighborhood, which are referred to as topology-based convolution networks. We start with an analysis of PPNP (Klicpera et al., 2018) and then derive a general formulation for topology-based approaches.

PPNP. PPNP provides insights towards the propagation scheme by combining message-passing function with personalized PageRank. As is proved in Xu et al. (2018b), the influence of node i on node j is proportional to a k-step random walk, which converges to the limit distribution with multiple stacked convolution layers. By involving the restart probability, PPNP is able to preserve the *starting node i*'s information. Similarly, in Definition 2, the first term can also be viewed as a regularization of preserving the original signal information. Therefore, we may achieve the same purpose by adjusting the regularization strength.

Definition 5 PPNP. *A graph convolution filter with personalized propagation (PPNP):*

$$\bar{\boldsymbol{X}} = \alpha(\boldsymbol{I}_n - (1-\alpha)\tilde{\mathcal{A}}_{\mathrm{rw}})^{-1}\boldsymbol{X}, \tag{13}$$

is equivalent to *the optimal solution of the following optimization:*

$$\min_{\bar{\boldsymbol{X}}} \; \alpha\sum_{i\in\mathbb{V}} \|\bar{\boldsymbol{x}}(i)-\boldsymbol{x}(i)\|^2_{\tilde{\boldsymbol{D}}} + (1-\alpha)\sum_{i,j\in\mathbb{V}} a_{ij}\|\bar{\boldsymbol{x}}(i)-\bar{\boldsymbol{x}}(j)\|^2_2, \tag{14}$$

where $\alpha \in (0, 1]$ is the restart probability.

Higher α means a higher possibility to teleport back to the starting node, which is consistent with the higher regularization on the original signal in (14).

Multi-hop PPNP. One of the possible weakness of the original PPNP is that personalized PageRank only utilizes the regularizer over the local structure. Therefore, we may improve the expressive capability by involving multi-hop information, which is equivalent to adding regularizers for higher-order variations.

Definition 6 Multi-hop PPNP. *Let t be the highest order adopted in the algorithm. A graph convolution filter with multi-hop personalized propagation (Multi-hop PPNP):*

$$\bar{\boldsymbol{X}} = \alpha_0(\boldsymbol{I}_n - \sum_{k=1}^{t} \alpha_k \tilde{\mathcal{A}}_{\mathrm{rw}}^k)^{-1}\boldsymbol{X}, \tag{15}$$

*where $\sum_{k=0}^{t} \alpha_k = 1$, $\alpha_0 > 0$ and $\alpha_k \geq 0, k = 1, 2, \ldots, t$, **is equivalent to** the optimal solution of the following optimization:*

$$\min_{\bar{\boldsymbol{X}}} \ \alpha_0 \sum_{i \in \mathbb{V}} \|\bar{\boldsymbol{x}}(i) - \boldsymbol{x}(i)\|_{\tilde{\boldsymbol{D}}}^2 + \sum_{k=1}^{t} \alpha_k \sum_{i,j \in \mathbb{V}} a_{ij}^{(k)} \|\bar{\boldsymbol{x}}(i) - \bar{\boldsymbol{x}}(j)\|_2^2, \tag{16}$$

where $a_{ij}^{(k)}$ is proportional to the transition probability of the k-step random walk and the same normalization trick in Section 2.2.3 is adopted on $\{a_{ij}^{(k)}\}$.

Solving Equation 15 directly is computationally expensive. Therefore, we derive an approximation by Taylor expansion and result in the form of:

$$\bar{\boldsymbol{X}} = (\sum_{i=0}^{T} c_i \tilde{\mathcal{A}}_{\mathrm{rw}}^i)\boldsymbol{X} + O(\tilde{\mathcal{A}}_{\mathrm{rw}}^T \boldsymbol{X}), \tag{17}$$

where $c_i, i = 0, 1, \ldots, T$, can be pre-computed as a function of αs. As the norm of the eigenvalues of $\mathcal{A}_{\mathrm{rw}}$ are bounded by 1, we can keep the first term in Equation 17 as a close approximation.

By comparing the approximated solution with topology-based graph convolution networks, we find that most of the approaches can be reformulated as a particular instance of Definition 6. For example, the formulation for Mixhop (Kapoor et al., 2019) can be derived as an approximation of Equation 17 if we let $t = 2$ and $\alpha_0 = \alpha_1 = \alpha_2 = 1/3$. Different learning weights can be applied to each hop as Section 2.2.2 to achieve the concatenation of the multi-hop signals. See more examples in Appendix **??**.

2.3 Remarks

In this section, we build a bridge between graph convolution operations and optimization problems in the graph Fourier space and provide insights into interpreting graph convolution operations with regularizers. For conclusion, we rewrite a general form of the unified framework as follow.

Definition 1' Unified Graph Convolution Framework. *Convolution-based graph neural networks can be reformulated (after approximation) as a particular instance of the optimal solution of the following optimization problem:*

$$\min_{\bar{\boldsymbol{X}}} \ \alpha_0 \sum_{i \in \mathbb{V}} (\|\bar{\boldsymbol{x}}(i) - x(\boldsymbol{i})\|_{\tilde{\boldsymbol{D}}}^2 - \underbrace{\epsilon \|\bar{\boldsymbol{x}}(i)\Theta\|_{\tilde{\boldsymbol{D}}}^2}_{\mathrm{Non-Conv}}) + \underbrace{\sum_{k=1}^{t} \alpha_k \sum_{i,j \in \mathbb{V}} \underbrace{\mathrm{att}_{ij}^{(k)} \|\bar{\boldsymbol{x}}(i)\Theta^{(k)} - \bar{\boldsymbol{x}}(j)\Theta^{(k)}\|_2^2}_{\mathrm{Attention-based}}}_{\mathrm{Topology-based}} + \lambda \mathcal{L}_{\mathrm{reg}}, \tag{18}$$

where $\sum_{k=0}^{t} \alpha_k = 1$, $\alpha_k \geq 0$ and $\sum_{j \in \mathbb{V}} \mathrm{att}_{ij} = \tilde{\boldsymbol{D}}_{ii}, \forall i \in \mathbb{V}$.

If we let d be the feature dimension of $\boldsymbol{X}$, then $\Theta, \Theta^{(k)} \in \mathbb{R}^{d \times d}$ are the corresponding learning weights. $\mathcal{L}_{\mathrm{reg}}$ corresponds to the personalized regularizer based on the framework, which can be effective if carefully designed as we will show in Section 3.

By establishing the unified framework, we interpret various convolution filters as carefully designed regularizers in the graph Fourier domain, which provides new insights on understanding graph learning modules from the graph signal perspective. Several graph learning modules are reformulated as smoothing regularizers over the graph structure with different intentions. While vanilla GCNs focus on minimizing the variation over the local graph structure, attention-based and topology-based GCNs take a step forward and concentrate on the differences between connected edges and graph structure with larger receptive field. This novel perspective enables a better understanding of the similarities and differences among many widely used GCNs, and may inspire new approaches for designing better models.

3 Tackling Oversmoothing under the Unified Framework

Despite the great success that GCNs have achieved, there are still limitations that need to be addressed. Instead of seeking solutions in the feature learning space, the proposed unified framework focuses on tackling common problems of current graph neural networks in the graph Fourier domain. As a showcase, we present a novel regularization technique under the framework to tackle the oversmoothing problem. We hope this can inspire a new line of work towards designing better graph learning models.

Several researches have shared insights on understanding and tackling oversmoothing. It is shown in Li et al. (2018) that the graph convolution of GCN is a special form of Laplacian smoothing and the authors try to compensate the long-range dependencies by co-training GCN with a random walk model. JKNet (Xu et al., 2018b) proved that the influence score between nodes converges to a fixed distribution when layer stacks, therefore losing local information. As a remedy, they proposed to concatenate layer-wise representations to perform mixed structural information. More recently, Oono & Suzuki (2020) theoretically demonstrated that graph neural networks lose expressive power exponentially due to oversmoothing.

In this paper, we adopt the definition of *feature-wise* oversmoothing in Zhao & Akoglu (2019). Due to multiple layers of Laplacian smoothing, all features fall into the same subspace spanned by the dominated eigenvectors of the normalized adjacency matrix, which also corresponds to the similar situation described in Klicpera et al. (2018). To tackle this problem, we propose to penalize the features when they are close to each other. Specifically, we consider the pairwise distance between normalized features, which is summarized as:

$$\delta(\boldsymbol{X}) = \frac{1}{d^2} \sum_{i,j \in d} \|\boldsymbol{x}_{\cdot i}/\|\boldsymbol{x}_{\cdot i}\| - \boldsymbol{x}_{\cdot j}/\|\boldsymbol{x}_{\cdot j}\|\|_2^2, \tag{19}$$

where d is the feature dimension and $\boldsymbol{x}_{\cdot i} \in \mathbb{R}^n$ represents the i-th dimension for all nodes. Therefore, Equation 19 can be interpreted as a feature variance regularizer, representing the distance between features after normalization. By adding this regularizer to the unified framework, the proposed filter should have the property to drive different features away.

Definition 7 Regularized Feature Variance. *Let $\otimes$ be the Kronecker product operator, $\mathrm{vec}(\boldsymbol{X}) \in \mathbb{R}^{n \times d}$ be the vectorized signal $\boldsymbol{X}$. Let $\boldsymbol{D}_X$ be a diagonal matrix whose value is defined by $\boldsymbol{D}_X(i,i) = \|\boldsymbol{x}_{\cdot i}\|_2$. A graph convolution filter with regularized feature variance:*

$$\mathrm{vec}(\bar{\boldsymbol{X}}) = (\boldsymbol{I}_n \otimes [(\alpha_1 + \alpha_2)\boldsymbol{I} - \alpha_2 \tilde{\mathcal{A}}_{\mathrm{rw}}] - \alpha_3[\boldsymbol{D}_x^{-1}(\boldsymbol{I} - \frac{1}{d}\boldsymbol{1}\boldsymbol{1}^T)\boldsymbol{D}_x^{-1}] \otimes \tilde{\boldsymbol{D}}^{-1})^{-1}\mathrm{vec}(\boldsymbol{X}) \tag{20}$$

***is equivalent to** the optimal solution of the following optimization:*

$$\min_{\bar{\boldsymbol{X}}} \alpha_1 \sum_{i \in \mathbb{V}} \|\bar{\boldsymbol{x}}(i) - \boldsymbol{x}(i)\|_{\tilde{\boldsymbol{D}}}^2 + \alpha_2 \sum_{i,j \in \mathbb{V}} a_{ij} \|\bar{\boldsymbol{x}}(i) - \bar{\boldsymbol{x}}(j)\|_2^2 - \alpha_3 \frac{1}{d} \sum_{i,j \in d} \|\bar{\boldsymbol{x}}_{\cdot i}/\|\boldsymbol{x}_{\cdot i}\| - \bar{\boldsymbol{x}}_{\cdot j}/\|\boldsymbol{x}_{\cdot j}\|\|_2^2, \tag{21}$$

where $\alpha_1 > 0$, $\alpha_2, \alpha_3 \geq 0$. For computation efficiency, we approximate $\|\bar{\boldsymbol{x}}_{\cdot i}\|$ with $\|\boldsymbol{x}_{\cdot i}\|$ as we assume that a single convolution filter provides little effect to the norm of features.

Calculating the Kronecker product and inverse operators are computationally expensive. Nevertheless, we can approximate Equation 20 via Taylor expansion with an iterative algorithm. If we let:

$$\boldsymbol{A} = (\alpha_1 + \alpha_2)\boldsymbol{I} - \alpha_2 \tilde{\mathcal{A}}_{\mathrm{rw}}, \qquad \boldsymbol{B} = \boldsymbol{I}_n, \tag{22}$$

$$\boldsymbol{C} = -\alpha_3 \tilde{\boldsymbol{D}}^{-1}, \qquad \boldsymbol{D} = \boldsymbol{D}_x^{-1}(1 - \frac{1}{d}\boldsymbol{1}\boldsymbol{1}^T)\boldsymbol{D}_x^{-1}. \tag{23}$$

Table 1: Test accuracy (%) on transductive learning datasets. We report mean values and standard deviations in 30 independent experiments. The best results are highlighted with **boldface**.

Method	Dataset	Citeseer	Cora	Pubmed
Vanilla	FastGCN (Chen et al., 2018)	68.8±0.6	79.8±0.3	76.8±0.6
	DGI (Veličković et al., 2019)	71.8±0.7	82.3±0.6	76.8±0.6
	GIN (Xu et al., 2018a)	66.1±0.9	77.6±1.1	77.0±1.2
	SGC (Wu et al., 2019a)	71.9±0.1	81.0±0.0	78.9±0.0
	GCN (Kipf & Welling, 2017)	70.3±0.4	81.5±0.5	79.0±0.4
	GCN+reg (ours)	**72.2±0.4**	**83.6±0.3**	**79.8±0.2**
Attention	AGNN (Thekumparampil et al., 2018)	71.6±0.5	82.7±0.4	78.9±0.4
	GatedGCN (Bresson & Laurent, 2017)	72.0±0.4	82.4±0.6	78.9±0.3
	MoNet (Monti et al., 2017)	-	81.7±0.5	78.8±0.4
	GAT (Veličković et al., 2018)	72.5±0.7	83.0±0.6	78.5±0.3
	GAT+reg (ours)	**73.3±0.4**	**83.9±0.6**	**80.3±0.3**
Topology	TAGCN (Du et al., 2017)	70.9	82.5	**81.1**
	MixHop (Kapoor et al., 2019)	71.4±0.8	81.9±0.4	80.8±0.6
	APPNP (Klicpera et al., 2018)	70.5±0.9	82.7±0.8	79.4±0.6
	APPNP+reg (ours)	**71.9±0.4**	**84.0±0.6**	80.2±0.3

Then, a t-order approximated formulation is summarized as:

$$\bar{\boldsymbol{X}}^{(0)} = \boldsymbol{X}, \tag{24}$$

$$\bar{\boldsymbol{X}}^{(k+1)} = \boldsymbol{X} + \bar{\boldsymbol{X}}^{(k)} - \boldsymbol{A}\bar{\boldsymbol{X}}^{(k)}\boldsymbol{B} - \boldsymbol{C}\bar{\boldsymbol{X}}^{(k)}\boldsymbol{D}, \quad k = 0, 1, \ldots, t-1. \tag{25}$$

Through approximation, computation overhead is greatly reduced. See details in the Appendix **??**.

As far as we are concerned, the advantages of utilizing feature variance regularization are threefold. First, the regularizer measures the difference between features, therefore explicitly preventing all features from falling into the same subspace. Second, the modified convolution filter does not require additional training parameters, avoiding the risk of overfitting. Third, the regularizer is designed based on the proposed unified framework, which means it can be easily implemented on other convolution-based networks as a plug-in module.

4 Experiment

In this section, we empirically validate the proposed regularization method on several widely used semi-supervised node classification benchmarks, including transductive and inductive settings. As we have stated in Section 3, our regularizer can be implemented on various convolution-based approaches under the unified graph convolution framework. Therefore, we consider three different versions by implementing the regularizer on vanilla-GCNs, attention-based GCNs and topology-based GCNs. We achieve state-of-the-art results on almost all of the settings and show the effectiveness of tackling oversmoothing on graph-structured data.

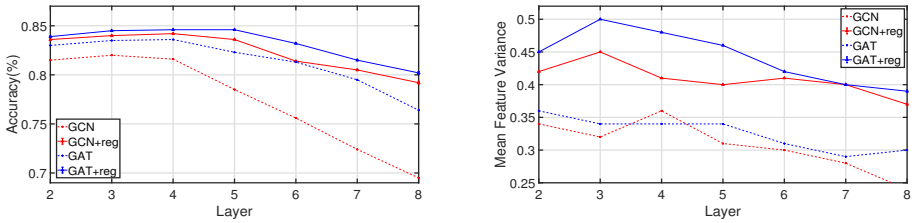
4.1 Dataset and Experimental Setup

We conduct experiments on four real-world graph datasets. For transductive learning, we evaluate our method on the Cora, Citeseer, Pubmed datasets, following the experimental setup in Sen et al. (2008). PPI (Zitnik & Leskovec, 2017) is adopted for inductive learning. Dataset statistics and more experimental setups are presented in Appendix **??**.

For comparison, we categorize state-of-the-art convolution-based graph neural networks into three specific classes, corresponding to the three versions of our proposed method. The first category is based on the vanilla-GCN proposed by Kipf & Welling (2017), including GCN, FastGCN (Chen et al., 2018), SGC (Wu et al., 2019a), GIN (Xu et al., 2018a), and DGI (Veličković et al., 2019). Since GIN is not initially evaluated on citation networks, we implement GIN following the setting in (Xu et al., 2018a). The second category corresponds to the attention-based approaches, including GAT (Veličković et al., 2018), AGNN (Thekumparampil et al., 2018), MoNet (Monti et al., 2017) and GatedGCN (Bresson & Laurent, 2017). The last category of approaches is topology-based GCNs which utilizes the structural information in the multi-hop neighborhood. We consider APPNP (Klicpera et al., 2018), TAGCN (Du et al., 2017) and MixHop (Kapoor et al., 2019) as the baselines.

4.2 Transductive Learning

Table 1 presents the performance of our method and several state-of-the-art graph neural networks on transductive learning datasets. For three classes of convolution-based approaches, we implement



(a) Accuracy (b) Mean Feature Variance

Figure 1: Accuracy and mean feature variance on Cora. Use GCN and GAT for comparison.

our regularizer with GCN, GAT and APPNP as comparisons with other baselines, respectively. For a fair comparison, we adopt the same network structure, hyperparameters and training configurations as baseline models. It is shown that the proposed model achieves state-of-the-art results on all three settings. On all of the datasets, we can observe a 0.5$\sim$1.0% higher performance after adopting the proposed regularizer. Notably, the proposed model achieves the highest improvement on the vanilla GCNs as this simplest version suffers most from the oversmoothing problem. Meanwhile, when combining with GAT, the model achieves the highest results comparing with almost all the baselines. Considering that attention mechanism and the regularization on oversmoothing focus on the local and global properties respectively, this can be an ideal combination for graph representation learning. We also conduct experiments on three citation networks with random splits and present the result in Appendix **??**.

4.3 Inductive Learning

For the inductive learning task, we implement our method on the vanilla GCN and GAT, and adopt the same experimental setup. Table 2 presents the comparison results on inductive learning dataset. It can be seen that our model compares favorably with all the competitive baselines. On the PPI dataset, out model achieves 0.5$\sim$1% higher on test Micro-F1 score, showing the effectiveness of applying our method under inductive settings.

Table 2: Test Micro-F1 Score on inductive learning dataset. We report mean values and standard deviations in 5 independent experiments.

Dataset	PPI
GAT (Veličković et al., 2018)	97.3
SGC (Wu et al., 2019a)	66.4
JKNet (Xu et al., 2018b)	97.6
GraphSAGE (Hamilton et al., 2017)	61.2
DGI (Veličković et al., 2019)	63.8
GCN+reg (ours)	97.69$\pm$0.32
GAT+reg (ours)	**98.23$\pm$0.08**

4.4 Analysis

As we have stated above, the proposed regularizer can be interpreted as the mean feature variance, which prevents different features falling into the same subspace. To testify the effect of our method, we compute the mean pairwise distance (Equation 19) of the last hidden layer of GCN and GAT, with and without regularizer on the Cora dataset. We show the result of models with 2-8 layers in Figure 1. As we can observe, the feature variance and the accuracies of models with regularization are comparably higher than vanilla models with obvious gaps. Therefore, after applying the regularizer, features are more separated from each other, and the oversmoothing problem is alleviated.

5 Conclusion

In this paper, we develop a unified graph convolution framework by establishing graph convolution filters with optimization problems in the graph Fourier space. We show that most convolution-based graph learning models are equivalent to adding carefully designed regularizers. Besides vanilla GCN, our framework is extended to formulating non-convolutional operations, attention-based GCNs and topology-based GCNs, which cover a large fraction of state-of-the-art graph learning models. On this basis, we propose a novel regularization on tackling the oversmoothing problem as a showcase, proving the effectiveness of designing new modules based on the framework.

Through the unified framework, we provide a general methodology for understanding and relating different graph learning modules, with new insights on tackling common problems and improving the generalization performance of current graph neural networks in the graph Fourier domain. Meanwhile, the unified framework can also serve as a once-for-all platform for handy modifications on convolution-based approaches. We hope our work can promote the recognitions towards graph convolutional networks and inspire more insights in this field.

REFERENCES

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *International Conference on Learning Representations*, 2018.

Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology adaptive graph convolutional networks. *CoRR*, abs/1710.10370, 2017.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

NT Hoang and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Amol Kapoor, Aram Galstyan, Bryan Perozzi, Greg Ver Steeg, Hrayr Harutyunyan, Kristina Lerman, Nazanin Alipourfard, and Sami Abu-El-Haija. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *International Conference on Machine Learning*, 2019.

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

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *International Conference on Learning Representations*, 2018.

Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning, 2019.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5115–5124, 2017.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3693–3702, 2017.

Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. Attention-based graph neural network for semi-supervised learning. *CoRR*, abs/1803.03735, 2018.

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

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *International Conference on Learning Representations*, 2019.

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *International Conference on Machine Learning*, 2019a.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 346–353, 2019b.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *CoRR*, abs/1810.00826, 2018a.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. *arXiv preprint arXiv:1806.03536*, 2018b.

Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7370–7377, 2019.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. *arXiv preprint arXiv:1909.12223*, 2019.

Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.