GSA-GCN: Global Self-Attention Graph Convolutional Networks

Anonymous authors

Abstract

Applying self-attention over global features has achieved remarkable success on Convolutional Neural Networks (CNNs) in capturing global dependencies between input dimensions. In this paper, inspired by the similarity between CNNs and Message-passing Graph Networks (MPGNs), we extend the mechanism to Graph Convolutional Networks (GCNs) and propose Global Self-Attention Graph Convolutional Networks (GSA-GCNs). The novel method allows GCNs to leverage global information regardless of local edge geometry and explore the dependence between nodes. Besides the above intuition, we provide another insight for the proposed GSA-GCN from the perspectives of over-fitting and over-smoothing. We discuss the advantage of applying self-attention to prevent overfitting and prove the mechanism can alleviate oversmoothing based on a recent theorem. Experiments on multiple benchmark datasets illustrate favorable performances for the proposed GSA-GCN, which corroborate the intuitions and the theoretical results.

1 Introduction

The emerge of Graph Convolutional Network (GCN) framework [1] has prompted Graph Networks to be one of the most promising techniques in pursuing Artificial General Intelligence [2]. Inspired by the closely related field of Concolutional Neural Networks (CNNs), different attention and self-attention mechanisms have been proposed to improve the quality of information aggregation under the GCN framework (e.g. [3]). Existing self-attention mechanisms in GCNs usually consider the feature information between neighboring vertices, and assign connection weights to each vertex accordingly [3; 4]. This type of considers *local* geometry from the edge connections of the graph, and exclude possible scenarios when a vertex could have strong correlations and influences with another *without edge connection*. To date, as we can see from a comprehensive survey [5], there has now been any significant work considering such an issue.

In this paper, inspired by the Global Self-attention method in CNNs, we propose **G**lobal **S**elf-**A**ttention **G**raph **C**onvolutional **N**etwork (GSA-GCN) by extending the idea to the graph domain. The application of global self-attention mechanism in CNNs, as evidently illustrated in [6], considerably boosts performances by capturing long-range pixel dependencies. Consequently, introducing the method to graph networks seems promising for performance gains. Meanwhile, the analogy between CNNs and GCNs ([5]) paved the way for achieving such technique. Specifically, for graph networks, each node can be roughly viewed as a pixel of an image, and the local edge-based information aggregation can be deemed as an analogy of convolutional kernels. Based on the above intuition, the global self-attention mechanism in graph networks is designed as direct product between every pairs of nodes, regardless of edge connection. We utilize this mechanism as a separate layer to obtain a feature map, and interpolate with the information process by local graph aggregation. An intuitive explanation of the GSA-GCN model can be found in figure 1.

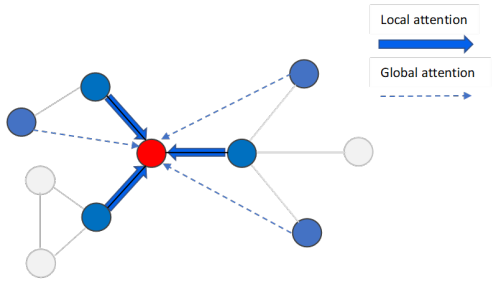

Figure 1: Local & Global Attention Mechanism

In addition to the above intuition, we notice that the feed-froward equation of GSA-GCN is roughly equivalent to interpolating a positive definite matrix to the original feature. This observation raises our interests in studying the properties concerning overfitting and over-smoothing for GSA-GCN. Most current GCN models incline to overfit such that testing accuracy decreases when training accuracy still appears to climb up. Another widely-discussed potential disadvantage of GCN models is over-smoothing, which indicates an exponential loss on both training and testing accuracies as more layers are stacked in GCN models. Consequently, in this paper, we analyzed the scenarios for GSA-GCN to tackle overfitting, and give a proof that the model can mitigate over-smoothing under reasonable assumptions based on recent theorems.

The rest of the paper is arranged as follows. Section 2

provides a brief review of related work and the context of this paper; Section 3 outlines the details of the method; The analysis and experimental results of GSA-GCN are discussed in sections 4 and 5; And finally, section 6 presents a general conclusion of our work.

2 Related Work

GSA-GCN is based on the framework of Message-passing Graph Networks (MPGNs), which was initiated by the idea of approximating graph spectral convolution [7; 8; 9] and popularized by the success of the GCN model [1]. To date, there have been fruitful outcomes on novel graph networks stemmed from the strategy [10; 11; 12]. The mechanism of aggregating information from neighboring nodes can be deemed as a spatial-based approach [13] which is similar to the procedures in Convolutional Neural Networks (CNNs) [5]. Consequently, the attention techniques in CNNs [14] have been widely examined from the perspective of GCNs [3; 15; 16; 17].

Despite the similarity between the two models, the attention methods in GCNs are mostly applied to local geometry with neighborhood connections, while CNN-based attention techniques are often applied to the global features. For instance, [6] provides a self-attention CNN as the generator of Generative Adversarial Networks (GANs) and shows its celebrated capabilities in capturing long-range feature relations. The remarkable success indicates potential advance by applying the global attention mechanism to GCNs to get similar results. Moreover, [18] outlines one significant weakness of Message-passing Graph Networks is the lack of 'the ability to capture long-range dependencies'. Consequently, it can be reasonably assumed that introducing global attention mechanism to GCNs will provide positive outcomes.

From a more theoretical perspective, the global self-attention mechanism on GCNs is also related to the over-smoothing problem. It has been long noticed that GCNs cannot be stacked as deep as CNNs without invoking negative effects [19; 20]. [21] provides an insight of this by showing linear GCN is a special form of Laplacian smoothing and will converge to an feature-invariant point as the network goes deeper. Sub-sequentially, [22] proves GCNs with Relu activation will converge to a feature-invariant space with a rate exponential to the maximum singular value of the convolutional filter. Inspired by the work, we establish a connection between over-smoothing and the global self-attention method by showing the maximum singular value will increase with the mechanism under reasonable assumptions.

3 Method

3.1 Graph Convolution Networks

Graph Convolutional Network (GCN) was first introduced in [1] as a scalable graph approach based on the approximation of spectrum. Given a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, a Graph Convolutional Networks takes as input a set of features $\boldsymbol{X}_i$ for every note $V_i \in \boldsymbol{V}$, resulting in a node-feature matrix representation $\boldsymbol{X}$ of a graph $n \times d$ where n is number of nodes and d is number of features. Additionally, a Graph Convolutional Network input will include a graph adjacency matrix $\boldsymbol{A}$ with the size of $n \times n$ to represent the edge connections. The network layers are connected using a convolutional projection of input graph $\boldsymbol{X}$ with adjacency matrix $\boldsymbol{A}$. In practice, we add self-loop on each node to get $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ with $\boldsymbol{I}$ as the identity matrix. Therefore, the feed-forward layer of a GCN can be expressed as:

$$\begin{aligned} \boldsymbol{H}^{(l+1)} &= f_{\boldsymbol{W}}(\boldsymbol{H}^{(l)}, \boldsymbol{A}) \\ &= \sigma(\boldsymbol{D}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}) \\ &= \sigma(\tilde{\boldsymbol{A}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}) \end{aligned} \tag{1}$$

where $\boldsymbol{W}^{(l)}$ is the weight matrix for convolution of the l-th layer and $\sigma(\cdot)$ is the activation function. Matrix $\boldsymbol{D}$ serves as the normalization matrix of $\hat{A}$:

$$\boldsymbol{D} = \texttt{diag}\left(\sum_{j=1}^{N} \hat{\boldsymbol{A}}_{\cdot,j}\right)$$

where $\texttt{diag}(\cdot)$ means the operation of expanding an n-length vector to a $n \times n$ matrix with the main diagonal filled by the elements of the vector. Thus, $\tilde{\boldsymbol{A}}$ is essentially the graph Laplacian of $\tilde{\boldsymbol{G}}$, which is the original graph with self-loops on each node.

3.2 Global Self-attention Mechanism

The self-attention mechanism on image data with Convolutional Neural Networks is discussed thoroughly in [6]. In this section, we discuss the self-attention mechanism in GCN following the method in [6] with modification to the graph structure. On each layer, the self-attention layer takes the output of the previous layer $\boldsymbol{H}^{(l)}$ and calculate the influence of node j on node i with the following equation:

$$\begin{aligned} \beta_{i,j} &= \texttt{softmax}_{j\in\{1,2,\cdots,n\}}[s_{i,j}] \\ \text{where } s_{i,j} &= (\hat{\boldsymbol{H}}_i^{(l)}\boldsymbol{W_l})(\hat{\boldsymbol{H}}_j^{(l)}\boldsymbol{W_r})^T \end{aligned} \tag{2}$$

where the calculation of $s_{i,j}$ is essentially a pair-wise production by summing over all the channels/features of the node. Matrices $\boldsymbol{W_l}$ and $\boldsymbol{W_r}$ serve the purpose of dimension-reduction to reduce the computational load and to provide additional flexibility to the trainable variables. We denote the result of the attention importance map as matrix $\boldsymbol{B}$, and it will be a $n \times n$ matrix. With the attention importance mask, the attention feature can be calculated with:

$$\boldsymbol{o}_i^{(l)} = (\sum_{j=1}^{N} \beta_{i,j}\boldsymbol{H}_j^{(l)}\boldsymbol{W}_h)\boldsymbol{W}_g \tag{3}$$

where $\boldsymbol{W}_h$ is the matrix to transform the input $\boldsymbol{H}^{(l)}$ to a lower dimension and $\boldsymbol{W}_h$ is the matrix to project the feature size back to the original. The operations of the above equation can be efficiently paralleled by re-writing it in the matrix production formula.

3.3 Global Self-Attention Graph Convolutional Network

The global self-attention layer correctly captures the feature information on a global level. Nevertheless, as we have discussed before, for graph inputs and networks, local geometry

denoted by edge connections is also crucial and it is the very characteristic that makes a graph network. Thus, we perform an interpolation similar first introduce by [6]. The resulting Global Self-attention GCN layer goes as follows:

$$\boldsymbol{H}^{(l+1)} = \sigma((\tilde{\boldsymbol{A}}\boldsymbol{H}^{(l)} + \gamma\boldsymbol{O}^{(l)})\boldsymbol{W}^{(l)}) \tag{4}$$

where $\boldsymbol{O}^{(l)}$ is the output of the global self-attention and γ is a non-negative trainable parameter ($\gamma > 0$) with initial value 0 and $\boldsymbol{W}^{(l)}$ is the convolution/filter matrix of layer l. Notice that the attention feature will *not* be processed by graph information aggregation in the GSA-GCN, as we do not mix the local and global information. Furthermore, the necessity of $\boldsymbol{W}^{(l)}$ is question in [23] and one can drop this matrix if the computational resource is limited. Nevertheless, we keep the matrix here to serve a general purpose of application and analysis.

The proposed GSA-GCN is a general framework that can be applied to various tasks in the regime of graph machine learning, such as graph node classification, graph classification, and graph embedding. In addition, as a mechanism independent from the network structure, the global self-attention layer of GSA-GCN can be applied to multiple graph neural networks.

4 Tackling Over-fitting and Over-smoothing with GSA-GCN

4.1 Over-fitting

In this section, we discuss the intuition for GSA-GCN to remedy the overfitting problem in transductive node and graph classification problems. For a transductive graph with static nodes and features, there are two major sources of overfitting: the noise in node features and the local graph connection geometries. More specifically, if the output graph embedding of a node is over-fitted, there are two likely reasons: 1. the graph network fails to capture the node pattern since it overly converges to node features in the training set; and 2. the node is adversely affected by the features from its neighbors and becomes dependent on the geometry.

The global self-attention mechanism in the GSA-GCN model can mitigate both of the above issues by interpolating the attention-augmented features. Particularly, for the first type of overfitting, GSA-GCN can 'adjust' the node features on each layer by taking the features over the graph into account. Notice that for a transductive graph, since all the node features are accessible in the training phase, the GSA-GCN can include information from distant nodes that will not be available without the global self-attention method. For the latter problem discussed before, we remark that the GSA-GCN model can leverage a geometry-invariant attention layer, which will, by intuition, alleviate the overfitting introduced by local geometries.

One can also understand the impact of the Global Self-attention mechanism on GCNs from the perspective of node subsampling [10] and edge dropout [24]. The two types of methods can be deemed as extensions of the Dropout technique on graph networks, and both have been shown to alleviate overfitting. We argue that equation 4 of GSA-GCN can be drawn equivalence with both node subsampling and edge dropout under specific conditions. To see the first equivalence, consider the case when the attention feature is a row-wise scaling of the current feature. In that case, the attention feature will selectively augment and down-weight some node, which essentially functions as 'node sampling.' For the second equivalence, consider the scenario that the attention feature will cancel the effect of a neighboring node for some nodes. In this case, the global self-attention layer will serve as a DropEdge process.

4.2 Over-smoothing

We theoretically show that with close approximations of matrix $\tilde{\boldsymbol{A}}$ and $\boldsymbol{B}$, and a reasonable simplification of the attention transformation matrices, the proposed GSA-GCN can guarantee to mitigate the over-smoothing problem in message passing-based Graph Networks. Our result is based on the theory in [22], which analyzes the over-smoothing problem as the convergence distance to an invariant subspace. Under the ReLu activation, the distance between the layer output and the space is upper-bounded by $O((s\lambda)^L)$, where s is the maximum singular value of the weight matrix, λ is a parameter related to the Graph Laplacian, and L is the number of layers. We demonstrate that applying the Global Self-attention mechanism is equivalent to performing convolution with a substituting weight matrix with a larger maximum singular value. We remark that our proof follows an idea similar to [24]. The difference is that the proof in [24] established a notion of ϵ-smoothing layer, while our proof is underpinned by the explicit increment of s.

Before diving into the analysis, let us first re-formulate a simplified version of the layer of GSA-GCN. Notice that the linear attention transformation matrices $\boldsymbol{W}_l, \boldsymbol{W}_r, \boldsymbol{W}_h, \boldsymbol{W}_g$ are employed mainly for the purpose of reducing computational complexity and accelerating training. Thus, we can simplify them to the identity matrix $\boldsymbol{I}$ in the analysis. Furthermore, we use simplified notations $\boldsymbol{H} := \boldsymbol{H}^{(l)}$ and $\boldsymbol{W} := \boldsymbol{W}^{(l)}$ to denote the output matrix at any layer. The convolutional output of GSA-GCN will become:

$$(\boldsymbol{H} + \gamma\tilde{\boldsymbol{A}}^{-1}\boldsymbol{B}\boldsymbol{H})\boldsymbol{W} \tag{5}$$

where $\tilde{\boldsymbol{A}}$ invertible as it is an approximation of the graph Laplacian. We assume the output matrices $\boldsymbol{H}$ satisfy the following property:

Assumption 1. $\boldsymbol{H}$ is of full row and column ranks. Notice that this implies $(\boldsymbol{H}^T\boldsymbol{H})$ and $(\boldsymbol{H}\boldsymbol{H}^T)$ are invertible, Hermitian and positive definite.

Also, we employ the following approximation of $\boldsymbol{B}$ and $\tilde{\boldsymbol{A}}$ in the analysis:

Assumption 2. We use $\hat{\boldsymbol{B}}$ as an Hermitian and positive definite approximation of $\boldsymbol{B}$. Moreover, we use $\tilde{\boldsymbol{A}}' = (1+\epsilon)\boldsymbol{I} + \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ ($\epsilon > 0$) as a variation of graph Laplacian to approximate $\tilde{\boldsymbol{A}}$. Notice that $\tilde{\boldsymbol{A}}'$ is also Hermitian and positive definite as it is strictly diagonally dominant.

Remark 4.1. *The approximations $\tilde{\boldsymbol{A}}'$ and $\hat{\boldsymbol{B}}$ are close to their real values for the most inputs and parameters. Notice*

that matrix $\boldsymbol{B}$ before the Softmax normalization is indeed Hermitian and the main diagonal of $\boldsymbol{B}$ is at least as large as any other element. Also, $\tilde{\boldsymbol{A}}'$ without the ϵ term will be Hermitian and positive semi-definite itself – and the input corresponds to the 0 eigenvalue is unique.

Denote the output of equation 5 as $\boldsymbol{H}\tilde{\boldsymbol{W}}$, with some simple algebraic manipulation, the expression of $\tilde{\boldsymbol{W}}$ will be:

$$\begin{aligned}\tilde{\boldsymbol{W}} &= (\boldsymbol{I} + \gamma(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\tilde{\boldsymbol{A}}^{-1}\boldsymbol{B}\boldsymbol{H})\boldsymbol{W} \\ &\approx (\boldsymbol{I} + \gamma(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\tilde{\boldsymbol{A}'}^{-1}\hat{\boldsymbol{B}}\boldsymbol{H})\boldsymbol{W}\end{aligned} \tag{6}$$

To simplify the notations, we define $\hat{\boldsymbol{C}} := \tilde{\boldsymbol{A}'}^{-1}\hat{\boldsymbol{B}}$. Notice that since both $\tilde{\boldsymbol{A}'}$ and $\hat{\boldsymbol{B}}$ are Hermitian and positive definite, $\hat{\boldsymbol{C}}$ should be Hermitian and positive definite. Finally, we also set $\gamma > 0$ in the analysis as it will be otherwise without the global self-attention mechanism.

Now we are ready to introduce the definitions and results in [22] as our preliminaries. The main concepts of [22] are the invariant subspace and its distance between a matrix. Formally, they are defined as follows:

Definition 4.1. ([22]) For $M, N \in \mathbb{N}^+$ and $M < N$, suppose there exists a subspace $U \subseteq \mathbb{R}^M$ for $\mathbb{R}^N$ such that:

- U has orthonormal basis $\boldsymbol{E}$ consisting of non-negative vectors;
- U is invariant under P;

then we define $\mathcal{M} := U \otimes R^C = \{\sum_{i=1}^M \boldsymbol{e}_i \otimes \boldsymbol{k}_i | \boldsymbol{k}_i \in \mathbb{R}^C\} = \{\boldsymbol{E}\boldsymbol{K} | \boldsymbol{K} \in \mathbb{R}^{M\times C}\}$ as the subspace of $\mathbb{R}^{N\times C}$. Furthermore, for any matrix $\boldsymbol{H} \in \mathbb{R}^{N\times C}$, we define the distance between $\boldsymbol{H}$ and $\mathcal{M}$ as $d_{\mathcal{M}}(\boldsymbol{H}) := \inf_{\boldsymbol{Y}\in\mathcal{M}}\{||\boldsymbol{H} - \boldsymbol{Y}||_F\}$.

Following the notions in the literature, we denote the maximum singular value of $\boldsymbol{W}^l$ as s_l, and let $s := \sup_{l\in\mathbb{N}^+} s_l$. We recall theorem 2 and corollary 2 of [22] for a multi-layer GCN to form the following lemma:

Lemma 4.2. *([22]) For any initial value $\boldsymbol{H}^{(0)}$ and the output of the l-th layer as $\boldsymbol{H}^{(l)}$, the convergence rate to $\mathcal{M}$ is $d_{\mathcal{M}}(\boldsymbol{H}^{(l)}) = O((s\lambda)^l)$, where λ is the second-largest eigenvalue of the normalized graph Laplacian. Furthermore, $\boldsymbol{H}^{(l)}$ satisfies $d_{\mathcal{M}}(\boldsymbol{H}^{(l)}) \leq (s\lambda)^l d_{\mathcal{M}}(\boldsymbol{H}^{(0)})$.*

In the above lemma we assume $s\lambda < 1$, which means $d_{\mathcal{M}}(\boldsymbol{H}^{(l)})$ will exponentially converge to 0. We are now ready to demonstrate our main theorem for over-smoothing:

Theorem 4.3. *At any level l, let $\boldsymbol{H}^{(l)}$ and $\hat{\boldsymbol{H}}^{(l)}$ denote the output of plain and GSA- GCN. The convergence rate to $\mathcal{M}$ for GSA-GCN is asymptotically slower than the plain GCN i.e. $d_{\mathcal{M}}(\hat{\boldsymbol{H}}^{(l)}) = \omega(d_{\mathcal{M}}(\boldsymbol{H}^{(l)}))$. Furthermore, there exists $\boldsymbol{H}^{(l)}$ on layer l and choice of γ such that $d_{\mathcal{M}}(\hat{\boldsymbol{H}}^{(l+1)}) > d_{\mathcal{M}}(\boldsymbol{H}^{(l+1)})$.*

The proof of theorem 4.3 crucially relies on the fact that for each layer, the maximum singular value of $\tilde{\boldsymbol{W}}$ is larger than its counterpart in $\boldsymbol{W}$. En route to proving theorem 4.3, we will show the following lemmas. We use $\lambda_{\min}(\cdot)$ and $\sigma_{\min}(\cdot)$ to denote the minimum eigenvalue and singular value.

Lemma 4.4. *The minimum eigenvalue and singular value of matrix $(\boldsymbol{I} + \gamma(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H})$ in equation 6 are greater than 1. i.e. let $\boldsymbol{P} := (\boldsymbol{I} + \gamma(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H})$, then $\lambda_{\min}(\boldsymbol{P}) > 1$ and $\sigma_{\min}(\boldsymbol{P}) > 1$.*

Proof. We first prove the matrix $\boldsymbol{Q} := \gamma(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H}$ is Hermitian and positive definite. According to assumption 1, since $\boldsymbol{H}^T\boldsymbol{H}$ is Hermitian and positive definite, $(\boldsymbol{H}^T\boldsymbol{H})^{-1}$ is Hermitian and positive definite; and since $\hat{\boldsymbol{C}}$ is Hermitian and positive definite, $\boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H}$ is Hermitian and positive definite. Thus, $\boldsymbol{Q}$ is positive definite by multiplication. We now show $\boldsymbol{Q}$ is Hermitian by showing $\bar{\boldsymbol{Q}}$ without γ is Hermitian:

$$\begin{aligned}\bar{\boldsymbol{Q}} &= (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H} \\ &= (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)(\boldsymbol{H}\boldsymbol{H}^T)^{-1}\hat{\boldsymbol{C}}\boldsymbol{H} \\ &= \boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1}\hat{\boldsymbol{C}}\boldsymbol{H} \\ &= \boldsymbol{H}^T\hat{\boldsymbol{C}}\hat{\boldsymbol{C}}^{-1}(\boldsymbol{H}\boldsymbol{H}^T)^{-1}\hat{\boldsymbol{C}}\boldsymbol{H} \\ &= \boldsymbol{H}^T\hat{\boldsymbol{C}}(\boldsymbol{H}\boldsymbol{H}^T\hat{\boldsymbol{C}})^{-1}\hat{\boldsymbol{C}}\boldsymbol{H} \\ &= \boldsymbol{H}^T\hat{\boldsymbol{C}}(\boldsymbol{H}\boldsymbol{H}^T\hat{\boldsymbol{C}})^{-1}\hat{\boldsymbol{C}}\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})(\boldsymbol{H}^T\boldsymbol{H})^{-1} \\ &= \boldsymbol{H}^T\hat{\boldsymbol{C}}(\boldsymbol{H}\boldsymbol{H}^T\hat{\boldsymbol{C}})^{-1}(\boldsymbol{H}\boldsymbol{H}^T\hat{\boldsymbol{C}})\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1} \\ &= \boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H}(\boldsymbol{H}^T\boldsymbol{H})^{-1} = \bar{\boldsymbol{Q}}^T\end{aligned}$$

Since $\gamma > 0$, we can conclude that $\boldsymbol{Q}$ is Hermitian, and its minimum eigenvalue $\lambda_{\min}(\boldsymbol{Q}) > 0$. Now we introduce a theory in [25], which says that if $\boldsymbol{A}$ and $\boldsymbol{B}$ are Hermitian matrices with eigenvalues denoted in descending order. i.e. $\lambda_1(\boldsymbol{A}) > \lambda_2(\boldsymbol{A}) > \cdots > \lambda_n(\boldsymbol{A})$ and $\lambda_1(\boldsymbol{B}) > \lambda_2(\boldsymbol{B}) > \cdots > \lambda_n(\boldsymbol{B})$, and if $\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{C}$ with the eigenvalue of $\boldsymbol{C}$ denoted in descending order. i.e. $\lambda_1(\boldsymbol{C}) > \lambda_2(\boldsymbol{C}) > \cdots > \lambda_n(\boldsymbol{C})$, then there is inequality:

$$\lambda_{n-i-j}(\boldsymbol{C}) \geq \lambda_{n-i}(\boldsymbol{A}) + \lambda_{n-j}(\boldsymbol{B})$$

If we set $i = 0, j = 0$ for the above inequality, and consider the condition that identity matrix is Hermitian and with eigenvalue 1, then we will have:

$$\lambda_{\min}(\boldsymbol{P}) \geq \lambda_{\min}(\boldsymbol{I}) + \lambda_{\min}(\boldsymbol{Q}) > 1$$

and since $\boldsymbol{P}$ is obviously Hermitian, which means its smallest singular value should be the absolute value of its smallest eigenvalue. Therefore, we have $\sigma_{\min}(\boldsymbol{P}) > 1$. □

Now we give the following proposition to help compare the singular values:

Proposition 1. *Let $||\cdot||$ be a 2-norm, and given two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ where $\boldsymbol{A}$ is a square matrix, then the inequalities $\sigma_{\min}(\boldsymbol{A})||\boldsymbol{B}|| \leq ||\boldsymbol{A}\boldsymbol{B}|| \leq \sigma_{\max}(\boldsymbol{A})||\boldsymbol{B}||$ hold.*

The proof of proposition 1 can be found in advanced linear algebra textbooks. Now we can demonstrate the relation between the singular values of $\tilde{\boldsymbol{W}}$ and $\boldsymbol{W}$.

Lemma 4.5. *At any layer l, let $s^{(l)}$ and $\tilde{s}^{(l)}$ be the maximum singular values of $\boldsymbol{W}^{(l)}$ and $\tilde{\boldsymbol{W}}^{(l)}$, respectively. Then there is $\tilde{s}^{(l)} > s^{(l)}$.*

Proof. The proof follows a simple combination of the results of lemma 4.4 and proposition 1. Let $||\cdot||$ be a 2-norm, then one will have:

$$\begin{aligned}\tilde{s}^{(l)} &= ||\tilde{\boldsymbol{W}}^{(l)}|| \\ &= ||(\boldsymbol{I}+\gamma(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H})\boldsymbol{W}^{(l)}|| \\ &\geq \sigma_{\min}(\boldsymbol{I}+\gamma(\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\hat{\boldsymbol{C}}\boldsymbol{H})||\boldsymbol{W}^{(l)}|| \qquad \text{(proposition 1)} \\ &> ||\boldsymbol{W}^{(l)}|| = s^{(l)} \qquad \text{(lemma 4.4)}\end{aligned}$$

□

Proof of Theorem 4.3

The asymptotic relation on convergence rate can be shown straightforwardly by taking the limit of the rate in lemma 4.2. Let $s = \sup_{l\in\mathbb{N}^+} s_l$ and $\hat{s} = \sup_{l\in\mathbb{N}^+} \hat{s}_l$ be the largest singular values among all layers, one can get:

$$\lim_{l\to\infty}\frac{d_{\mathcal{M}}(\hat{\boldsymbol{H}}^{(l)})}{d_{\mathcal{M}}(\boldsymbol{H}^{(l)})} = \lim_{l\to\infty}(\frac{\hat{s}}{s})^l = \infty \qquad (\tfrac{\hat{s}}{s} > 1)$$

The proof of the second part of the theorem relies on the proposition 4 of [22], which stated that for some $\boldsymbol{H}$, if $s^{(l)}\lambda > 1$, we will have $d_{\mathcal{M}}(\boldsymbol{H}^{(l+1)}) > d_{\mathcal{M}}(\boldsymbol{H}^{(l)})$. Thus, for layer l, if we choose γ so that $s^{(l)}\lambda < 1$ and $\hat{s}^{(l)}\lambda > 1$, then there exists input $\boldsymbol{H}^{(l)} = \hat{\boldsymbol{H}}^{(l)}$ so that:

$$\begin{aligned}d_{\mathcal{M}}(\boldsymbol{H}^{(l+1)}) &\leq (s^{(l)}\lambda)d_{\mathcal{M}}(\boldsymbol{H}^{(l)}) \\ &< (\hat{s}^{(l)}\lambda)d_{\mathcal{M}}(\hat{\boldsymbol{H}}^{(l+1)}) \\ &< d_{\mathcal{M}}(\hat{\boldsymbol{H}}^{(l+1)})\end{aligned}$$

5 Experiments

We evaluate the performance of GSA-GCN with multiple benchmark datasets on two tasks: (I) node classification with citation network datasets: cora, citeseer and pubmed [26], aiming to classify academic papers into various subjects, (II) graph classification on the COIL-RAG dataset. We split the experiments into three parts to show the respective properties of the GSA-GCN. Firstly, we apply the GSA-GCN to both supervised and semi-supervised node classifications to demonstrate the competitive performance of GSA-GCN comparing with other state-of-the-art methods. The semi-supervised training follows the experimental setup in [1], and the full-supervised training follows the practice in [24; 27]. Secondly, to corroborate the theorem for the GSA-GCN to mitigate over-smoothing, we show the comparison between the training accuracy for GCNs and GSA-GCNs as the number of layers goes deep. And finally, to illustrate the preferable performances of GSA-GCN on graph classification, we compare the classification on the COIL-RAG dataset for GSA- and plain GCNs. The metadata of aforementioned datasets are illustrated in table 1.

Table 1: Datasets used in the Experiments

Dataset	Graphs	Nodes	Edges	Classes	Features
Cora	1	2708	5429	7	1433
Citeseer	1	3327	4732	6	3703
Pubmed	1	19717	44338	3	500
COIL-RAG	3900	3.01	3.02	100 (graph)	64

5.1 Semi- and Full-supervised Node Classification

We apply the same data splits described in [28; 1], and evaluate predicting accuracy on a test set with 1000 labeled examples. We compare the performance of GSA-GCN with several recently-proposed compelling models based on GCNs, and two-layer backbone is considered as baseline unless otherwise denoted. For semi-supervised learning, only a marginal portion of labeled instances are used in training process, thus it yields performance to full-supervised learning. Table 2 demonstrates test accuracy on three datasets by GCNs [1], Graph Attention Networks [3], Jumping Knowledge Networks [29] and DropEdge-GCN [24]. We also summarize previous state-of-the-art testing accuracy on three datasets with full-supervised learning in table 3. Noted that the accuracies of comparison models are either as reported in original paper or obtained by fine-tuning to the best performance as we know.

Table 2: Accuracy(%) comparisons with semi-supervised SOTA

Model	Cora	Citeseer	Pubmed
GCN	81.5	70.3	79.0
GAT	83.0	72.5	79.0
JK-Net (4)	80.2	68.7	78.0
DropEdge-GCN	82.8	72.3	79.6
GSA-GCN	**83.3**	**72.9**	**80.1**

Table 3: Accuracy(%) comparisons with full-supervised SOTA

Model	Cora	Citeseer	Pubmed
GCN	86.1	75.9	90.2
GAT	86.4	76.6	OOM
JK-Net	86.9	78.3	90.5
DropEdge-GCN	86.5	78.7	**91.2**
GSA-GCN	**88.2**	**79.1**	89.4

As illustrated in table 2 and table 3, GSA-GCN outperforms other state-of-the-art methods for multiple tasks. The method performs epecially well on semi-supervised tasks, which also confirms its capacities in mitigating overfitting.

5.2 Over-smoothing

To provide empirical evidences for theorem 4.3, we show the changes of training accuracy with an increasing number of layers on plain GCN and GSA-GCN. Two fully-supervised node classification datasets (Cora and Citeseer) are employed in this part of the experiment. Notice that since over-smoothing will obstruct the accuracy optimization on both training and

testing sets, demonstrating the ability for the network to fit the *training* accuracy is enough to show the robustness against over-smoothing.

The results can be summarized in figure 2 and 3. From the figures, it can be observed that GSA-GCN can consistently achieve a higher training accuracy than its plain counterpart. Moreover, as the graph network deepens to a further extent, the gap becomes more significant. We observe from the training dynamic that the training accuracy of plain GCN for deep models typically stuck at a low level after few epochs, while the curves in GSA-GCN illustrate a fluctuating pattern even it fails to converge to a high value.

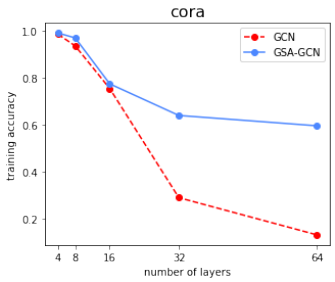

Figure 2: Over-smoothing with model depth on cora

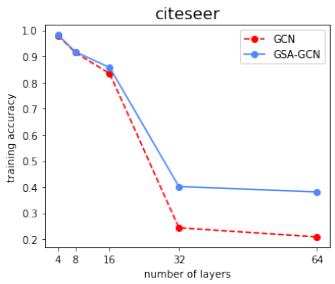

Figure 3: Over-smoothing with model depth on citeseer

5.3 Graph Classification

Limited by space, for the graph classification task, we only illustrate the results on the COIL-RAG datasets. We tested the results on plain and GSA-augmented GCNs, and referred an external result from [30]. Our read-out layer for graph classification is based on summation pooling, and the weight regularization is set to $1e-4$. The results are summarized in table 4.

Table 4: Graph Classification Performance of GCNs and GSA-GCNs

Model	Training Accuracy	Test Accuracy
GCN	95.35	85.15
SPI-GCN [30]	N.A.	75.72
GSA-GCN (ours)	**97.34**	**88.28**

From the table, it can be observed that the proposed GSA-GCN model can provide the optimal performance, and the overfitting problem is alleviated in the sence that the higher training accuracy does not bring a downfall on the test-set performance. To further illustrate the insights of GSA-GCN, we plot the training and testing accuracy curves with respect to the number of epochs in figure 4. From the figure, it can be observed that the curves of both the training and testing accuracy of GSA-GCN stand to a higher level than their plain GCN counterpart. The disadvantage of GSA-GCN is that its curves are more fluctuating, indicating a more sensitive nature to the model parameters and learning rates.

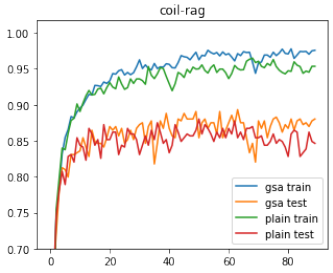

Figure 4: Graph Classification Accuracy for Plain and GSA GCNs

6 Conclusion

In this paper, motivated by recent cutting-edge developments on Graph Convolution Networks and global self-attention mechanism, we propose Global Self-attention Graph Convolution Networks (GSA-GCN). To the best of the authors' knowledge, GSA-GCN is first-of-its-kind in graph machine learning to leverage global self-attention mechanism for improved information aggregation. Moreover, we demonstrate its advantageous characteristics in tackling overfitting and over-smoothing through analysis and a rigorous proof. Experimental results on two classical tasks illustrate compelling performance for GSA-augmented GCNs over existing models, and its robustness against over-smoothing with deeper layers verifies our theoretical results.

References

[1] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[4] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on*

Knowledge Discovery & Data Mining, pages 1666–1674. ACM, 2018.

[5] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[6] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[7] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[9] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

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

[11] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[15] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In *UAI*, 2018.

[16] Chen Wang, Chengyuan Deng, and Vladimir Ivanov. Sag-vae: End-to-end joint inference of data representations and feature relations. *arXiv preprint arXiv:1911.11984*, 2019.

[17] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *ArXiv*, abs/2001.02908, 2020.

[18] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

[19] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. The truly deep graph convolutional networks for node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[20] Guohao Li, Matthias Müller, Ali K. Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[21] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

[23] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019.

[24] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020.

[25] Allen Knutson and Terence Tao. Honeycombs and sums of hermitian matrices. 2001.

[26] 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.

[27] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[28] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

[29] 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*, 2018.

[30] Asma Atamna, Nataliya Sokolovska, and Jean-Claude crivello. SPI-GCN: A Simple Permutation-Invariant Graph Convolutional Network. Apr 2019. preprint, HAL hal-02093451.