Graph Convolutional Neural Networks via Motif-based Attention

Hao Peng† Jianxin Li† Qiran Gong† Yuanxin Ning† Lihong Wang‡
†Department of Computer Science & Engineering, Beihang University, Beijing, China
‡National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
{penghao,lijx,gongqr,ningyx}@act.buaa.edu.cn wlh@isc.org.cn

Abstract
Many real-world problems can be represented as graph-based learning problems. In this paper, we propose a novel framework for learning spatial and attentional convolution neural networks on arbitrary graphs. Different from previous convolutional neural networks on graphs, we first design a motif-matching guided subgraph normalization method to capture neighborhood information. Then we implement self-attentional layers to learn different importances from different subgraphs to solve graph classification problems. Analogous to image-based attentional convolution networks that operate on locally connected and weighted regions of the input, we also extend graph normalization from one-dimensional node sequence to two-dimensional node grid by leveraging motif-matching, and design self-attentional layers without requiring any kinds of cost depending on prior knowledge of the graph structure. Our results on both bioinformatics and social network datasets show that we can significantly improve graph classification benchmarks over traditional graph kernel and existing deep models.

1 Introduction
Graphs or networks are natural representations in many real-world applications, such as proteomics (Baldi and Pollastri 2004), molecular chemistry (Benkö, Flamm, and Stadler 2003), social network (Barabási et al. 2002), biomedical image analysis (So and Chung 2009), nature language processing (Gao et al. 2005), etc. Deep learning approaches had been successful in object recognition and image classification. So some recent works have focused on generalizing convolutional neural networks (CNNs) to structures beyond grids, i.e., from 2D/3D images to arbitrary graphs or networks (Bruna et al. 2014; Niepert, Ahmed, and Kutzkov 2016; Defferrard, Bresson, and Vandergheynst 2016; Velickovic et al. 2018; Duvenaud et al. 2015; Kipf and Welling 2017; Verma and Zhang 2018). These convolutional neural networks on graphs are now commonly known as Graph Convolutional Networks (GCNs), and are robust in feature extraction by representing arbitrary graphs as fix-sized feature maps with effective rich structured semantics information.

The principal idea behind GCNs has been derived from the graph signal processing (Shuman et al. 2012), which has been extended in different ways (Bruna et al. 2014; Duvenaud et al. 2015; Defferrard, Bresson, and Vandergheynst 2016; Kondor et al. 2018; Verma and Zhang 2018). Despite prosperities and effectiveness of GCNs models, there are still following four major limitations of the GCNs commonly used in deep learning approaches on graphs, especially when applied to graph classification problems.

- Oversimplification of graph convolution operations in local neighborhood. As aggregation of node values in a local neighborhood, there is a potential loss of information associated with the basic graph convolution operations, as discussed in (Hinton, Krizhevsky, and Wang 2011).
- Incapability of isomorphic graph classification. GCNs models can’t be applied directly because they are equivalent with respect to the node order in a graph, which means they cannot provide any guarantee that the outputs of any two isomorphic graphs are always the same (Verma and Zhang 2018).
- Limited ability in exploiting global information. The graph convolutional filters are local in nature and provide an “average/aggregated view” of the local data. This shortcoming poses a serious difficulty in handling graphs where node labels are not present.
- Shortcoming in Max Pooling. The Max Pooling cannot assure that the isomorphic subgraphs share invariant feature vector owing to the comparison done with respect to node ordering, since the Max Pooling can mess up the whole comparison order of subgraphs. And, the Max Pooling operator is a popular choice after aggregation for achieving invariance.

There have been several attempts in the literature to extend neural networks to arbitrarily structured graphs. On the one hand, recent works (Benson, Gleich, and Leskovec 2016; Yin et al. 2017; Yin, Benson, and Leskovec 2018) using motif-matching based technologies to generate higher-order connectivity patterns have proved the effectiveness of network motifs’ feature extraction. Particularly, Motif-CNN (Sankar, Zhang, and Chang 2017), MotifNet (Monti and Bronstein 2018), NEST (Carl et al. 2018) and Deep motif dashboard (Lanchantin et al. 2016), have made important progress towards generalizing motif neural operations in a computational graph structure. On the other hand, the generalized graph attention network (Velickovic et al. 2018) has been proven efficient in learning different importances to different nodes in a graph for discriminating node features. However, despite the potential presented in various kinds of Graph Convolution Networks, their accuracy on graph

Graph Convolutional Neural Networks via Motif-based Attention

Hao Peng† Jianxin Li† Qiran Gong† Yuanxin Ning† Lihong Wang‡
†Department of Computer Science & Engineering, Beihang University, Beijing, China
‡National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China
{penghao,lijx,gongqr,ningyx}@act.buaa.edu.cn wlh@isc.org.cn

Abstract
Many real-world problems can be represented as graph-based learning problems. In this paper, we propose a novel framework for learning spatial and attentional convolution neural networks on arbitrary graphs. Different from previous convolutional neural networks on graphs, we first design a motif-matching guided subgraph normalization method to capture neighborhood information. Then we implement self-attentional layers to learn different importances from different subgraphs to solve graph classification problems. Analogous to image-based attentional convolution networks that operate on locally connected and weighted regions of the input, we also extend graph normalization from one-dimensional node sequence to two-dimensional node grid by leveraging motif-matching, and design self-attentional layers without requiring any kinds of cost depending on prior knowledge of the graph structure. Our results on both bioinformatics and social network datasets show that we can significantly improve graph classification benchmarks over traditional graph kernel and existing deep models.

1 Introduction
Graphs or networks are natural representations in many real-world applications, such as proteomics (Baldi and Pollastri 2004), molecular chemistry (Benkö, Flamm, and Stadler 2003), social network (Barabási et al. 2002), biomedical image analysis (So and Chung 2009), nature language processing (Gao et al. 2005), etc. Deep learning approaches had been successful in object recognition and image classification. So some recent works have focused on generalizing convolutional neural networks (CNNs) to structures beyond grids, i.e., from 2D/3D images to arbitrary graphs or networks (Bruna et al. 2014; Niepert, Ahmed, and Kutzkov 2016; Defferrard, Bresson, and Vandergheynst 2016; Velickovic et al. 2018; Duvenaud et al. 2015; Kipf and Welling 2017; Verma and Zhang 2018). These convolutional neural networks on graphs are now commonly known as Graph Convolutional Networks (GCNs), and are robust in feature extraction by representing arbitrary graphs as fix-sized feature maps with effective rich structured semantics information.

The principal idea behind GCNs has been derived from the graph signal processing (Shuman et al. 2012), which has been extended in different ways (Bruna et al. 2014; Duvenaud et al. 2015; Defferrard, Bresson, and Vandergheynst 2016; Kondor et al. 2018; Verma and Zhang 2018). Despite prosperities and effectiveness of GCNs models, there are still following four major limitations of the GCNs commonly used in deep learning approaches on graphs, especially when applied to graph classification problems.

- Oversimplification of graph convolution operations in local neighborhood. As aggregation of node values in a local neighborhood, there is a potential loss of information associated with the basic graph convolution operations, as discussed in (Hinton, Krizhevsky, and Wang 2011).
- Incapability of isomorphic graph classification. GCNs models can’t be applied directly because they are equivalent with respect to the node order in a graph, which means they cannot provide any guarantee that the outputs of any two isomorphic graphs are always the same (Verma and Zhang 2018).
- Limited ability in exploiting global information. The graph convolutional filters are local in nature and provide an “average/aggregated view” of the local data. This shortcoming poses a serious difficulty in handling graphs where node labels are not present.
- Shortcoming in Max Pooling. The Max Pooling cannot assure that the isomorphic subgraphs share invariant feature vector owing to the comparison done with respect to node ordering, since the Max Pooling can mess up the whole comparison order of subgraphs. And, the Max Pooling operator is a popular choice after aggregation for achieving invariance.

There have been several attempts in the literature to extend neural networks to arbitrarily structured graphs. On the one hand, recent works (Benson, Gleich, and Leskovec 2016; Yin et al. 2017; Yin, Benson, and Leskovec 2018) using motif-matching based technologies to generate higher-order connectivity patterns have proved the effectiveness of network motifs’ feature extraction. Particularly, Motif-CNN (Sankar, Zhang, and Chang 2017), MotifNet (Monti and Bronstein 2018), NEST (Carl et al. 2018) and Deep motif dashboard (Lanchantin et al. 2016), have made important progress towards generalizing motif neural operations in a computational graph structure. On the other hand, the generalized graph attention network (Velickovic et al. 2018) has been proven efficient in learning different importances to different nodes in a graph for discriminating node features. However, despite the potential presented in various kinds of Graph Convolution Networks, their accuracy on graph
classification tasks has not been considerably higher than that of their convolutional counterparts, and these GCNs lack intuitive analytical capabilities.

In this paper, we propose a novel framework, which we call Motif-matching Normalized and Attentional Graph Convolutional Neural Networks, including Motif-GCNN and Motif-GCNN-Attention models, to overcome the limitations of graph convolutional operations and capture neighbor features. We carefully design a suitable network motif for bioinformatics and social network datasets, and perform motif-matching guided graph normalization to rasterize selected small graphs. Our models take the impact of different edges on node labeling generation into consideration. To illustrate, there exist different structures of bond between two elements in molecules, such as single bond, double bond, triple bond, etc. Then we implement deep convolutional neural networks to extract discriminating graph features. In particular, in order to avoid the isomorphic subgraph problem and adapt computable self-attention precondition, we not only abandon pooling operations in both directions, but also ensure that each dimension on the feature map represents the characteristics of a subgraph by setting horizontal strides. As a basic model, we adopt the popular Graph CNNs (Niepert, Ahmed, and Kutzkov 2016) due to its simplicity and comparable performance to other GCNs models (Bruna et al. 2014, Henaff, Bruna, and LeCun 2015, Dettferrard, Bresson, and Vandergheynst 2016, Duvenaud et al. 2015).

We conduct extensive experiments on 12 benchmark datasets for graph classification tasks. Compared with state-of-the-art methods, including traditional graph kernel based algorithms and existing deep learning approaches, our Motif-GCNN and Motif-GCNN-Attention models obtain significant performance gains for most of the tasks.

The rest of the paper is organized as follows. We start with related work on main approaches to solve the graph classification problem in Section 2. In Section 3, we introduce a motif-matching guided graph normalization to generate sequential and gridding representation. In Section 4, we design a generalization of attention graph convolutional neural network for arbitrary graphs to learn sequential subgraph weights. After that, we show the datasets, baselines, and experiments in Section 5. Finally, we conclude this paper in Section 7.

2 Related Work

In this section, we introduce the related work on graph kernel based methods, graph convolution neural networks, and motif-based convolution neural network.

Much of work has been devoted to finding which sub-graph or kernel function is the most suitable one for graph classification. Among the existing graph kernels, popular ones are graphlets (Shervashidze et al. 2009), random walk and shortest path kernels (Borgwardt and Kriegel 2005), Weisfeiler-Lehman subtree kernel (Shervashidze et al. 2011), deep graph kernels (Yanardag and Vishwanathan 2015), graph invariant kernels (Orsini, Frasconi, and Raedt 2015), and multiscale laplacian graph kernel (Kondor and Pan 2016). These kernels based models have been proposed to capture sub-structural similarity at different levels.

Graph convolution neural networks models are more recent and perhaps more promising when dealing with graph classification tasks. The original idea of defining graph convolution has been recognized as the problem of learning filter parameters that appear in the graph fourier transform in the form of a graph Laplacian (Bruna et al. 2014, Henaff, Bruna, and LeCun 2015, Kipf and Welling 2017, Duvenaud et al. 2015, Atwood and Towsley 2016) have proposed a self-loop graph adjacency matrix and a propagation rule to compute and update each neural network layer weights. An optimized GCNNs model is proposed in (Dettferrard, Bresson, and Vandergheynst 2016) by utilizing fast localized spectral filters and efficient pooling operations. Considering the weakness of traditional CNNs in spatial hierarchies and rotational invariance, a Graph Capsule networks model is proposed in (Verma and Zhang 2018). Other RNN autoencoders based graph representation method adopts random walks, breadth-first search, and shortest paths to generate node sequences to learn structural features (Aynaz and Tanya 2018). The most relevant GCNs in this paper is (Niepert, Ahmed, and Kutzkov 2016), as a standardized process from arbitrary graphs to convolutional neural networks, and the GCNs framework consists of node sequence selection, graph normalization, and convolutional architecture.

Motifs are high-order structures that are crucial in many domains such as bioinformatics, neuroscience, and social networks. Recent work has explored motifs in clustering (Benson, Gleich, and Leskovec 2016, Yin, Benson, and Leskovec 2018) and graph classification (Sankar, Zhang, and Chang 2017, Lanchantin et al. 2016, Monti and Bronstein 2018, Carl et al. 2018) tasks. However, above-mentioned techniques do not fully exploit motifs to capture local stationary and spatial structures’ features on graphs. In (Ivanov and Burnaev 2018), authors use distribution of anonymous walks as a network embedding, sampling walks in a graph to approximate actual distribution with a given confidence. There are some works focusing on applying structures of dataset to optimize neural network frameworks. Simonov and Komoda (2017) introduces a edge-conditioned convolution operation on graph signal performed in the spatial domain where filter weights are conditioned on edge labels and dynamically generated for each specific input sample. Ying et al. (2018) shows a differentiable graph pooling module to generate hierarchical representations of graphs. In (Zhang, Cui, and Chen 2018), authors propose a SortPooling layer which sorts graph vertices in a consistent order so that traditional neural networks can be trained on the graphs.

3 Motif-matching based Graph Normalization

In this section, we introduce graph processing for arbitrary graphs, including node sequencing and selection, neighborhood graph construction, and motif-matching guided graph normalization, as shown in Figure 7.

We first explain why we choose the two-hop paths motif. First, we will see in experimental section that the differences among nodes of bioinformatics datasets and social network datasets are relatively small, since the nodes represent chemi-
Supposing arbitrarily connected graph $G = (V, E)$ with $n$ vertices and $m$ edges ($|V| = n, |E| = m$), we use $d(v, u)$ to denote the length of a shortest-path between $v$ and $u$ and compute the closeness centrality $C_v = (n - 1)/\sum_{u \in V, u \neq v} d(v, u)$ for each node $v$ in parallel. Here, we treat the double, triple and higher bonds of the edges as two, three, and more single bonds in bioinformatics datasets, and ignore the directional and weighting information of the edge. We first sort the nodes according to their closeness centrality, and select the sequence of Top $N$ as central nodes, as shown in the step 1 in Figure 1. If the number of central nodes on graph is less than $N$, it is padded with zeros. Algorithm 1 presents one such procedure. Second, we extract sub-graph $G(v)$ for each selected central node $v$. We limit the number of nodes in the sub-graph $G(v)$ does not exceed $K$, and the sub-graph is extracted in the order of first, second, and third order using breadth first search algorithm, and their closeness centrality, as shown in the step 2 in Figure 1. Algorithm 2 is executed to construct a neighborhood field, until exactly $K$ nodes of sub-graph have been created.

**Algorithm 1** SelNodeSequence: Select Node Sequence

**Require:** graph $G = (V, E)$, vertex $v \in V$, receptive field size $K$, number of central nodes $N$

**Ensure:** matrix $M$

**function** SELNODESEQUENCE($G, N, K$)

1. Treat multiple bonds as multiple single bond in $G$
2. $V_{sort} = \text{top } N \text{ elements of } V \text{ according to closeness centrality}$
3. $i = 1$, $j = 1$
4. while $j < N$
5.   if $i \leq |V_{sort}|$
6.      $g = \text{NEIGHFIELD}(V_{sort}[i], K)$
7.   else
8.      $g = \text{ZERONEIGHFIELD}()$
9.   end if
10. $M' = \text{MONORMGRA}(g, V_{sort}[i], w_1, w_2, w_3)$
11. $j = j + 1$, $i = i + 1$
12. $M = \text{Concatenate}(M, M')$
13. end while
14. return $M$

**end function**

Then we compute the shortest distance from any node in subgraph to central node, and recalculate closeness centrality of each node within subgraph range. We sort the nodes in subgraph $G(v)$ by updated closeness centrality, shortest distance, and original closeness centrality. As shown in the step 3 in Figure 1, red node represents central node, and colors indicate distance to the central node. So, we label nodes by the distance to the central node.

**Algorithm 2** NeighField: Neighborhood Field

**Require:** central node $v \in V_{sort}$, receptive field size $K$

**Ensure:** subgraph $G$ of neighborhood for node $v$

**function** NEIGHFIELD($v, K$)

1. while $|G| < K$ and $|G| > 0$
2.   $\langle \bar{v}, \bar{c} \rangle = \text{EIGH}(v, 1)$
3.   $G = G \cup \langle \bar{v}, \bar{c} \rangle$
4.   $\langle \bar{v}, \bar{c} \rangle = \text{EIGH}(v, 2)$
5.   $G = G \cup \langle \bar{v}, \bar{c} \rangle$
6.   $\langle \bar{v}, \bar{c} \rangle = \text{EIGH}(\text{ClosenessCentrality}(v, 3))$
7.   $G = G \cup \langle \bar{v}, \bar{c} \rangle$
8. end while
9. return $G$

**end function**

---

*Figure 1:* An illustration of motif-matching guided graph processing. In real datasets, there are some edges with double or triple bonds, which we treat in the form of edge weights in steps.
tance and closeness centrality, such as \{a_1, a_2, \ldots, a_8\}. In order to preserve spatial structure information, we introduce a motif-matching based subgraph normalization method and design a novel matrix \( M(v) \) where the rows are sorted by different available orders away from central node. We divide the matrix \( M(v) \) into three blocks initializing with zeros, and fill row with matching two-hop paths\(^1\). The first block must contain central node. Similarly, the second or third blocks should contain 2 hop or 3 hop nodes from center node. As shown in Figure 1, different available hops contain different numbers of blocks, such as \{a_1, a_2, a_3\}, \{a_1, a_2, a_4\}, and \{a_1, a_3, a_5\} in the first block, \{a_2, a_4, a_6\}, \{a_2, a_4, a_7\}, etc. in the second block, and \{a_4, a_6, a_7\}, \{a_5, a_7, a_8\} in the third block. Note that the intrinsic order of the three nodes does not affect subsequent results of convolutional operations. Considering the problem of unified matrix representation of sub-graphs with different scales, we fix the row number of the first, second, and third blocks as \( w_1, w_2, \) and \( w_3 \) in different tasks. If the number of connected sub-graph set is smaller than corresponding size, it creates zero receptive fields for padding purposes. So, each sub-graph \( G(v) \) can be represented by a matrix \( M(v) \). Next, we concatenate the \( N \) matrices into a large combined matrix \( M(v) \) follow the sequence in step 1. Algorithm 3 lists the above graph normalization procedure.

Each element in the combined matrix refers to a node, and we adopt node labeling to represent the value. So, we can implement a common convolutional neural network to capture the rich spatial structure and node labeling informations. Note that the convolution network can have different strides in the horizontal and vertical directions, and no pooling operations to keep some original properties. We name the combine of motif-matching guide graph convolutional neural network and traditional fully-connected network as Motif-GCNN. We introduce more advanced and generalized attention graph convolutional neural networks in the next section.

4 Learning Attention Graph CNNs

We design an architecture with attentional graph convolutional neural networks, as shown in Figure 2. The size of the input concatenated matrix \( M = 3 \times N \times (w_1 + w_2 + w_3) \), where \( N \) is the number of selected sub-graphs, \( 3 \) is the number of nodes in the above two-hop paths motif, \( (w_1 + w_2 + w_3) \) is the total rows of first-block, second-block and third-block of matrices. To extract distinguishable structural features, the convolutional operations will follow some graph processing rules. The convolution kernel size is \( 3 \times 1 \) and slides in horizontal and vertical directions respectively. Compared with traditional convolution networks, the difference includes that the horizontal direction stride is \( 1 \) equaling the size of above two-hop paths motif, the vertical direction step is \( 1 \). For a convolution kernel, the output feature map size is \( N \times (w_1 + w_2 + w_3) \), where each vertical dimension characterizes the spatial structure information of the corresponding central node. We use \( K_1 \) convolution kernels to convolve the same way to generate \( K_1 \times N \times (w_1 + w_2 + w_3) \) features with multichannel representation. In the second layer, the convolution kernel size is \( 1 \times 3 \times K_1 \), where the horizontal stride is \( 1 \) and the vertical stride is \( 3 \). Note that in order to preserve spatial structure information, we did not use pooling operations. We denote \( T = (w_1 + w_2 + w_3) / 3 \). The second layer output is \( N \times K_2 \times T \) with \( K_2 \) convolution kernels. The feature map’s size can be changed with variety of convolutional strides. We ensure that each part of the \( N \) sections represents the characteristics of one normalized subgraph.

Different from traditional fully connected layers and dropout operation, we implement a masked self-attentional layers to address mutual influences among subgraph features, as shown in Figure 2. We name this framework as Motif-GCNN-Attention. To represent subgraphs’ features, we concatenate different channel feature maps into a \( K_2 \times T \times N \) size of flat matrix representation. So, for any subgraph \( G(v) \), the subgraph feature can be represented as a \( F = K_2 \times T \) size of vector \( \hat{h}_v \). For any central nodes \( i \) and \( j \) in \( V_{sort} \), the corresponding feature vectors of the subgraphs are \( \hat{h}_i \) and \( \hat{h}_j \). Then, we add a \( W \in R^{F \times F} \) weight matrix and \( \alpha \) weight vector to learn mutual influences. More precisely, the attentional layer contains \( F \) hidden neurons, and the attention mechanism is a feedforward neural network. When applying the LeakyReLU non-linearity (Xu et al. 2015) (with negative input slope \( \alpha = 0.2 \)), the coefficients of \( \hat{h}_j \) on \( \hat{h}_i \) computed by the attention mechanism can be formalized

---

\(^1\)Because the number of nodes in sub-graph is small, we ignore more fast subgraph matching algorithms (Sun et al. 2012).
as:

\[
\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\tilde{a}^T[W\tilde{h}_i \parallel W\tilde{h}_j]))}{\sum_{k \in N} \exp(\text{LeakyReLU}(\tilde{a}^T[W\tilde{h}_i \parallel W\tilde{h}_k]))},
\]

where \( \cdot^T \) represents transposition and \( \parallel \) is concatenation operation. Following the attention mechanism, we perform \( S \) independent attention computations, and employ averaging strategy to evaluate influences. The final nonlinearity uses a softmax or sigmoid for classification problem:

\[
\tilde{h}^\prime_i = \sigma\left(\frac{1}{S} \sum_{s=1}^{S} \sum_{j \in N} \alpha_{ij} s W^s \tilde{h}_j, \right),
\]

where \( \sigma \) is a sigmoid or softmax function, and \( \tilde{h}^\prime_i \) is the output feature of subgraph \( G(i) \). Finally, we sum the output features according to the catalog, as shown in Figure 2. The maximum value represents the category it belongs to.

The dominant reason for such approach based on the assumption that each subgraph could partially represent the whole graph. Through attentional layers, we could obtain classification on each subgraph. Then we can form an ensemble according to these presumably classified results by applying sum operation and a softmax layer.

5 EXPERIMENTAL SETTINGS

In the experiments, we compare our proposed algorithms with state-of-the-art traditional graph kernels based classification models as well as recently developed deep learning architectures on twelve datasets.

Datasets. We evaluate performances on two types of real-world graphs, i.e., bioinformatics and social network datasets. Table 1 summarizes the statistics of these datasets. Bioinformatics datasets contain six categories, namely MUTAG, PTC, PROTEINS, D&D, NC11, and NC1109. MUTAG is a dataset of 188 mutagenic aromatic and heteroaromatic nitro compounds (Debnath, RL, and Debnath 1991) with 7 discrete node labels, namely C, N, O, F, I, Cl, Br, and 4 discrete edge labels, including aromatic, single, double, and triple bonds. The classes indicate whether the compound has a mutagenic effect on a bacterium. PTC (Toivonen, Srinivasan, and Helma 2003) is a dataset of 344 organic molecules marked according to their carcinogenicity on male and female mice and rats, and it has 19 discrete labels in nodes. PROTEINS is a graph collection obtained from (Borgwardt, Cheng, and Vishwanathan 2005) where nodes are secondary structure elements and edges indicate neighborhood in the amino-acid sequence or in 3D space with 61 discrete labels. The graphs are classified as enzyme or non-enzyme. D&D is a dataset of 1178 protein structures (Dobson and Doig 2003) with 82 discrete labels, and is also classified into enzymes and non-enzymes. NC11 and NC1109 datasets are chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell lines (Wale and Karypis 2008), and contain 4100 and 4127, respectively. In order to test the effectiveness of our algorithms on unlabeled graphs, we derive several social network datasets with different tasks. We use node degree as attribute in these datasets, and even it can easily incorporate continuous features. COLLAB is scientific collaboration dataset of 5000 graphs, derived from 3 public collaboration datasets. IMDB-BINARY and IMDB-MULTI are movie collaboration datasets, and contain 1000 and 1500 graphs, respectively. The task is then simply to identify which genre an ego-network graph belongs to. REDDIT-BINARY, REDDIT-MULTI-5K, and REDDIT-MULTI-12K are balanced datasets where each graph corresponds to an online discussion thread and their nodes correspond to users. There is an edge between two nodes if at least one of them
Weisfeiler-Lehman Sub-tree kernel (WL) (Shervashidze et al. works namely: PATCHY-SAN (PSCN) (Niepert, Ahmed, and Vishwanathan 2015) as kernel baselines. For deep learning based graph classification, Network Structural Convolutional Neural Network (DGCNN) (Zhang, Cui, and Chen 2018), Graph Capsule CNN (GCAPS-CNN) (Verma and Zhang 2018), Anony-
Table 3: Classification accuracy on bioinformatics datasets.

| Methods     | MUTAG      | PTC        | PROTEINS   | D&D        | NCI1       | NCI109     |
|-------------|------------|------------|------------|------------|------------|------------|
| SP          | 85.79±0.25 | 58.24±2.44 | 75.07±0.54 | -          | 73.00±0.24 | 73.00±0.21 |
| WL          | 83.78±0.46 | 57.97±0.49 | 74.68±0.49 | 79.78±0.36 | 84.55±0.36 | 83.53±0.30 |
| GK          | 81.58±2.11 | 57.26±1.41 | 71.67±0.55 | 78.45±1.11 | 62.28±0.29 | 62.60±0.19 |
| DGK         | 87.44±2.72 | 60.08±2.55 | 75.68±0.54 | 73.50±1.01 | 80.31±0.46 | 80.32±0.33 |

| FSCN        | 92.63±4.21 | 62.29±3.68 | 77.12±2.41 | 76.34±1.68 | 76.30±2.31 | 74.13±3.27 |
| ECC         | 89.44      | 72.65      | 74.10      | 83.80      | 82.14      |
| DGCCN       | 85.83±1.66 | 58.59±2.47 | 75.54±0.94 | 79.37±0.94 | 74.44±0.47 | 75.03±1.72 |
| GCAPS-CNN   | 82.41±2.31 | 66.01±5.91 | 76.40±4.17 | 72.62±4.99 | 82.72±2.38 | 81.12±1.28 |
| AWE         | 81.58±2.11 | -          | 71.51±4.02 | -          | 83.72±0.4  | 83.64±0.3  |
| S2S-N2N-PP  | 89.86±1.11 | 64.54±1.11 | 76.61±0.5  | -          | 81.59±0.46 | 81.72±0.41 |
| NEST        | 91.85±1.57 | 67.42±1.83 | 76.54±0.26 | 78.11±0.36 | 81.59±0.46 | 81.72±0.41 |
| DIFFPOOL    | -          | -          | 78.10      | 81.15      | -          | -          |

| Motif-GCNN   | 92.78±3.56 | 70.30±3.59 | 78.19±1.93 | 81.48±1.11 | 80.91±2.17 | 79.69±1.95 |
| Motif-GCNN-Attention | 93.91±2.95 | 71.77±5.13 | 79.35±1.74 | 81.37±1.43 | 81.77±2.36 | 80.33±2.19 |

| Gain        | 1.28       | 4.35       | 1.25       | 0.33       | -          | -          |

Table 4: Classification accuracy on social network datasets.

| Methods     | COLLAB     | IMDB-B     | IMDB-M     | RE-BINARY  | RE-MULTU-5K | RE-MULTU-12K |
|-------------|------------|------------|------------|------------|-------------|--------------|
| WL          | 79.02±1.77 | 73.41±4.63 | 49.33±4.75 | 81.10±1.90 | 49.44±2.36  | 38.18±1.30  |
| GK          | 72.84±0.28 | 65.87±0.98 | 43.89±0.38 | 77.34±0.18 | 41.01±0.17  | 31.82±0.08  |
| DGK         | 73.09±0.25 | 66.96±0.56 | 44.55±0.52 | 78.04±0.39 | 41.27±0.18  | 32.22±0.10  |
| PSCN        | 72.60±2.15 | 71.00±2.29 | 45.23±2.84 | 86.30±1.58 | 49.10±0.70  | 41.32±0.32  |
| DGCCN       | 73.76±0.49 | 70.03±0.86 | 47.83±0.85 | 76.02±1.73 | 48.70±4.54  | 41.82       |
| GCAPS-CNN   | 77.71±2.51 | 71.69±3.40 | 48.50±4.10 | 87.61±2.51 | 50.10±1.72  | -           |
| AWE         | 73.93±1.94 | 74.45±5.83 | 51.54±3.61 | 87.89±2.53 | 54.74±2.93  | 41.51±1.98  |
| S2S-N2N-PP  | 81.75±0.8  | 73.8±0.7   | 51.19±0.5  | 86.50±0.8  | 52.28±0.5   | 42.47±0.1   |
| NEST        | 74.55±0.60 | 73.26±7.26 | 53.08±3.31 | 88.52±0.64 | 48.61±0.46  | 42.80±0.28  |
| DIFFPOOL    | 82.13      | -          | -          | -          | -           | 47.04       |

| Motif-GCNN   | 74.31±2.05 | 75.10±3.14 | 52.19±2.66 | 88.06±1.29 | 55.62±2.19  | 47.35±1.31  |
| Motif-GCNN-Attention | 74.93±1.38 | 77.20±2.96 | 53.77±3.11 | 89.44±1.18 | 56.18±1.48  | 48.14±1.93  |

| Gain        | -          | 2.75       | 0.69       | 0.92       | 1.44        | 1.10        |

NEST model by an average of 4.35%. The average of node of PROTEINS and D&D datasets is larger than the MUTAG and PTC datasets, so we expend hyperparameters in $N,K,w_1,w_2,w_3$. Here, we were able to achieve up to 1.25% and 0.33% average accuracy gain, and smaller standard deviations on PROTEINS and D&D datasets. For NCI1 and NCI109 datasets, our Motif-GCNN-Attention model achieves state-of-the-art result on 6 out of 10 benchmarks.

**Social Network Graph Classification.** Table 4 compares the performance of Motif-GCNN and Motif-GCNN-Attention models to these state-of-the-art graph classification baselines for social network datasets. We adopt the normalized node degree as node attribute in social network datasets. The nodes of the social dataset are richer in attributes and have continuity characteristics than the bioinformatics datasets. We also mark the Top-2 score in bold. As we can see, our Motif-GCNN and Motif-GCNN-Attention achieve highly competitive results with the compared deep learning and kernel based models, including achieving the highest accuracies on IMDB-B, IMDB-M, REDDIT-BINARY, REDDIT-MULTU-5K (RE-MULTU-5K) and REDDIT-MULTU-12K (RE-MULTU-12K) datasets. In COLLAB dataset, our Motif-GCNN-Attention model achieves state-of-the-art result on 6 out of 10 benchmarks. In IMDB-B and IMDB-M datasets, our Motif-GCNN-Attention model achieves the highest accuracies, 77.20% and 53.77%, respectively. Even, our Motif-GCNN-Attention model improves upon the base the AWE model by an average of 2.75% and smaller standard deviations. In REDDIT-BINARY, REDDIT-MULTU-5K, and REDDIT-MULTU-12K datasets, we have adopted a larger hyper-parameters setting in $N,K,w_1,w_2,w_3$ and batch size. Our Motif-GCNN-Attention model improves upon the base NEST, AWE and DIFFPOOL models by an average of 0.92%, 1.44%, and 1.10%, and smaller standard deviation.

Altogether, our Motif-GCNN and Motif-GCNN-Attention models show very promising results against both the current state-of-art deep learning and graph kernels methods. The highest prediction accuracies can be obtained from 20 to 500 epochs.

### 7 Conclusion and Discussion

In this paper, we proposed a novel Motif-matching Normalized and Attentional Graph Convolutional Neural Networks that are able to extract the complex spatial informations and train impacts of different spatial subgraphs in real-world graphs. By using the proposed motif-matching guided subgraph normalization and masked self-attentional layers, we
achieved new state-of-the-art performances on several graph classification benchmarks. Interesting future directions include designing semantically rich meta-path/meta-schema to replace the isomorphic motif, taking advantage of fast subgraph matching algorithms, and applying motif-matching and self-attentional methods to other tasks that require modeling of the entire graph structure and perform a thorough analysis on the model interpretability.

8 Acknowledgments
Hao Peng and Qiran Gong contribute equally! This work is supported by NSFC program (No.61872022,61772151,61421003) and partly by the Beijing Advanced Innovation Center for Big Data and Brain Computing.

References
[Atwood and Towsley 2016] Atwood, J., and Towsley, D. 2016. Diffusion-convolutional neural networks. In NIPS.
[Aynaz and Tanya 2018] Aynaz, T., and Tanya, B.-W. 2018. Learning graph representations with recurrent neural network autoencoders. In DLDay.
[Baldi and Pollastri 2004] Baldi, P., and Pollastri, G. 2004. The principled design of large-scale recursive neural network architectures--dag-rnns and the protein structure prediction problem.
[Barabási et al. 2002] Barabási, A.; Jeong, H.; Nát' da, Z.; Ravasz, E.; Schubert, A.; and Vicsek, T. 2002. Evolution of the social network of scientific collaborations. Physica A Statistical Mechanics andand Its Applications 311(3):590–614.
[Benkő, Flamm, and Stadler 2003] Benkő, G.; Flamm, C.; and Stadler, P. F. 2003. A graph-based toy model of chemistry. Journal of Chemical Information and Computer Sciences 43(4):1085–1093.
[Benson, Gleich, and Leskovec 2016] Benson, A. R.; Gleich, D. F.; and Leskovec, J. 2016. Higher-order organization of complex networks. Science.
[Borgwardt and Krieger 2005] Borgwardt, K. M., and Krieger, H. P. 2005. Shortest-path kernels on graphs. In IEEE International Conference on Data Mining, 74–81.
[Borgwardt, Cheng, and Vishwanathan 2005] Borgwardt, K. M.; Cheng, S.; and Vishwanathan. 2005. Protein function prediction via graph kernels. Bioinformatics.
[Bruna et al. 2014] Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral networks and locally connected networks on graphs. In ICLR.
[Carl et al. 2018] Carl, Y.; Mengxiong, L.; Vincent W., Z.; and Jiawei, H. 2018. Node, motif and subgraph: Leveraging network functional blocks through structural convolution. In ASONAM.
[Debnath, Ri, and Debnath 1991] Debnath, A. K.; Ri, L. D. C.; and Debnath. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds, correlation with molecular orbital energies and hydrophobicity. JMC.
[Defferrard, Bresson, and Vandergheynst 2016] Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16, 3844–3852. USA: Curran Associates Inc.
[Dobson and Doig 2003] Dobson, P. D., and Doig, A. J. 2003. Distinguishing enzyme structures from non-enzymes without alignments. JMB.
[Duvenaud et al. 2015] Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; and Adams, R. P. 2015. Convolutional networks on graphs for learning molecular fingerprints. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS’15, 2224–2232. Cambridge, MA, USA: MIT Press.
[Gao et al. 2005] Gao, B.; Liu, T. Y.; Feng, G.; Qin, T.; Cheng, Q. S.; and Ma, W. Y. 2005. Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph copartitioning. IEEE Transactions on Knowledge and Data Engineering 17(9):1263–1273.
[Henaff, Bruna, and LeCun 2015] Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep convolutional networks on graph-structured data. CoRR.
[Hinton, Krizhevsky, and Wang 2011] Hinton, G. E.; Krizhevsky, A.; and Wang, S. D. 2011. Transforming auto-encoders. In Honkela, T.; Duch, W.; Girolami, M.; and Kaski, S., eds., Artificial Neural Networks and Machine Learning – ICANN 2011, 44–51. Berlin, Heidelberg: Springer Berlin Heidelberg.
[Ivanov and Burnaev 2018] Ivanov, S., and Burnaev, E. 2018. Anonymous walk embeddings. In ICML.
[Kipf and Welling 2017] Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In ICLR.
[Kondor and Pan 2016] Kondor, R., and Pan, H. 2016. The multiscale laplacian graph kernel. In NIPS.
[Kondor et al. 2018] Kondor, R.; Son, H. T.; Pan, H.; Anderson, B.; and Trivedi, S. 2018. Covariant compositional networks for learning graphs. arXiv.
[Lanchantin et al. 2016] Lanchantin, J.; Singh, R.; Wang, B.; and Qi, Y. 2016. Deep motif dashboard: Visualizing and understanding genomic sequences using deep neural networks. PSB.
[Monti and Bronstein 2018] Monti, F., and Bronstein, M. M. 2018. Motifnet: a motif-based graph convolutional network for directed graphs. arXiv.
[Niepert, Ahmed, and Kutzkov 2016] Niepert, M.; Ahmed, M.; and Kutzkov, K. 2016. Learning convolutional neural networks for graphs. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML’16, 2014–2023. JMLR.org.
[Orsini, Frasconi, and Raedt 2015] Orsini, F.; Frasconi, P.; and Raedt, L. D. 2015. Graph invariant kernels. In International Conference on Artificial Intelligence, 3756–3762.
[Sankar, Zhang, and Chang 2017] Sankar, A.; Zhang, X.; and Chang, C. C. 2017. Motif-based convolutional neural network on graphs. arXiv.

[Shervashidze et al. 2009] Shervashidze, N.; Vishwanathan, S. V. N.; Petri, T. H.; Mehlhorn, K.; and Borgwardt, K. M. 2009. Efficient graphlet kernels for large graph comparison. AISTATS.

[Shervashidze et al. 2011] Shervashidze, N.; Schweitzer, P.; van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. J. Mach. Learn. Res. 12:2539–2561.

[Shuman et al. 2012] Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2012. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine 30(3):83–98.

[Simonov and Komoda 2017] Simonov, M., and Komoda, N. 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In CVPR.

[So and Chung 2009] So, R. W. K., and Chung, A. C. S. 2009. Multi-level non-rigid image registration using graph-cuts. In IEEE International Conference on Acoustics, Speech and Signal Processing, 397–400.

[Sun et al. 2012] Sun, Z.; Wang, H.; Wang, H.; Shao, B.; and Li, J. 2012. Efficient subgraph matching on billion node graphs. In VLDB.

[Toivonen, Srinivasan, and Helma 2003] Toivonen, H.; Srinivasan, A.; and Helma, C. 2003. Statistical evaluation of the predictive toxicology challenge 2000â–Y2001. Bioinformatics.

[Veličković et al. 2018] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In ICLR.

[Verma and Zhang 2018] Verma, S., and Zhang, Z. L. 2018. Graph capsule convolutional neural networks. arXiv.

[Wale and Karypis 2008] Wale, N., and Karypis, G. 2008. Comparison of descriptor spaces for chemical compound retrieval and classification. In ICDM.

[Xu et al. 2015] Xu, B.; Wang, N.; Chen, T.; and Li, M. 2015. Empirical evaluation of rectified activations in convolutional network. In ICML.

[Yanardag and Vishwanathan 2015] Yanardag, P., and Vishwanathan, S. V. N. 2015. Deep graph kernels. In Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 1365–1374.

[Yin, Benson, and Leskovec 2018] Yin, H.; Benson, A. R.; and Leskovec, J. 2018. Higher-order clustering in networks. arXiv.

[Yin et al. 2017] Yin, H.; Benson, A. R.; Leskovec, J.; and Gleich, D. F. 2017. Local higher-order graph clustering. In KDD.

[Ying et al. 2018] Ying, R.; You, J.; Morris, C.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. Hierarchical graph representation learning with differentiable pooling. In DL-Day.

[Zhang, Cui, and Chen 2018] Zhang, M.; Cui, Z.; and Chen, Y. 2018. An end-to-end deep learning architecture for graph classification. In AAAI.