# CONFIDENCE-BASED GRAPH CONVOLUTIONAL NETWORKS FOR SEMI-SUPERVISED LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Predicting properties of nodes in a graph is an important problem with applications in a variety of domains. Graph-based Semi Supervised Learning (SSL) methods aim to address this problem by labeling a small subset of the nodes as seeds, and then utilizing the graph structure to predict label scores for the rest of the nodes in the graph. Recently, Graph Convolutional Networks (GCNs) have achieved impressive performance on the graph-based SSL task. In addition to label scores, it is also desirable to have a confidence score associated with them. Unfortunately, confidence estimation in the context of GCN has not been previously explored. We fill this important gap in this paper and propose ConfGCN, which estimates labels scores along with their confidences jointly in GCN-based setting. ConfGCN uses these estimated confidences to determine the influence of one node on another during neighborhood aggregation, thereby acquiring *anisotropic*[1] capabilities. Through extensive analysis and experiments on standard benchmarks, we find that ConfGCN is able to significantly outperform state-of-the-art baselines. We have made ConfGCN's source code available to encourage reproducible research.

## 1 INTRODUCTION

Graphs are all around us, ranging from citation and social networks to knowledge graphs. Predicting properties of nodes in such graphs is often desirable. For example, given a citation network, we may want to predict the research area of an author. Making such predictions, especially in the semi-supervised setting, has been the focus of graph-based semi-supervised learning (SSL) (Subramanya & Talukdar, 2014). In graph-based SSL, a small set of nodes are initially labeled. Starting with such supervision and while utilizing the rest of the graph structure, the initially unlabeled nodes are labeled. Conventionally, the graph structure has been incorporated as an explicit regularizer which enforces a smoothness constraint on the labels estimated on nodes (Zhu et al., 2003a; Belkin et al., 2006; Weston et al., 2008). Recently proposed Graph Convolutional Networks (GCN) (Defferrard et al., 2016; Kipf & Welling, 2016) provide a framework to apply deep neural networks to graph-structured data. GCNs have been employed successfully for improving performance on tasks such as semantic role labeling (Marcheggiani & Titov, 2017), machine translation (Bastings et al., 2017), relation extraction (Vashishth et al., 2018; Zhang et al., 2018), event extraction (Nguyen & Grishman, 2018), shape segmentation (Yi et al., 2016), and action recognition (Huang et al., 2017). GCN formulations for graph-based SSL have also attained state-of-the-art performance (Kipf & Welling, 2016; Liao et al., 2018; Veličković et al., 2018). In this paper, we also focus on the task of graph-based SSL using GCNs.

GCN iteratively estimates embedding of nodes in the graph by aggregating embeddings of neighborhood nodes, while backpropagating errors from a target loss function. Finally, the learned node embeddings are used to estimate label scores on the nodes. In addition to the label scores, it is desirable to also have confidence estimates associated with them. Such confidence scores may be used to determine how much to trust the label scores estimated on a given node. While methods to estimate label score confidence in non-deep graph-based SSL has been previously proposed (Orbach & Crammer, 2012), confidence-based GCN is still unexplored.

[1] anisotropic (adjective): varying in magnitude according to the direction of measurement (Oxford English Dictionary)

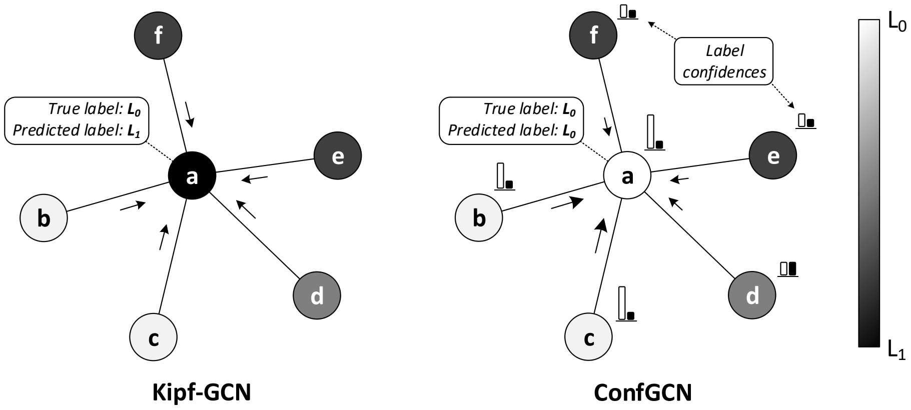


Figure 1: Label prediction on node $a$ by Kipf-GCN and ConfGCN (this paper). $L_0$ is $a$'s true label. Shade intensity of a node reflects the estimated score of label $L_1$ assigned to that node. Since Kipf-GCN is not capable of estimating influence of one node on another, it is misled by the dominant label $L_1$ in node $a$'s neighborhood and thereby making the wrong assignment. ConfGCN, on the other hand, estimates confidences (shown by bars) over the label scores, and uses them to increase influence of nodes $b$ and $c$ to estimate the right label on $a$. Please see Section 1 for details.

In order to fill this important gap, we propose ConfGCN, a GCN framework for graph-based SSL. ConfGCN jointly estimates label scores on nodes, along with confidences over them. One of the added benefits of confidence over node's label scores is that they may be used to subdue irrelevant nodes in a node's neighborhood, thereby controlling the number of effective neighbors for each node. In other words, this enables *anisotropic* behavior in GCNs. Let us explain this through the example shown in Figure 1. In this figure, while node $a$ has true label $L_0$ (white), it is incorrectly classified as $L_1$ (black) by Kipf-GCN (Kipf & Welling, 2016)[2]. This is because Kipf-GCN suffers from limitations of its neighborhood aggregation scheme (Xu et al., 2018). For example, Kipf-GCN has no constraints on the number of nodes that can influence the representation of a given target node. In a $k$-layer Kipf-GCN model, each node is influenced by all the nodes in its $k$-hop neighborhood. However, in real world graphs, nodes are often present in *heterogeneous* neighborhoods, i.e., a node is often surrounded by nodes of other labels. For example, in Figure 1, node $a$ is surrounded by three nodes ($d$, $e$, and $f$) which are predominantly labeled $L_1$, while two nodes ($b$ and $c$) are labeled $L_0$. Please note that all of these are estimated label scores during GCN learning. In this case, it is desirable that node $a$ is more influenced by nodes $b$ and $c$ than the other three nodes. However, since Kipf-GCN doesn't discriminate among the neighboring nodes, it is swayed by the majority and thereby estimating the wrong label $L_1$ for node $a$.

ConfGCN is able to overcome this problem by estimating confidences on each node's label scores. In Figure 1, such estimated confidences are shown by bars, with white and black bars denoting confidences in scores of labels $L_0$ and $L_1$, respectively. ConfGCN uses these label confidences to subdue nodes $d, e, f$ since they have low confidence for their label $L_1$ (shorter black bars), whereas nodes $b$ and $c$ are highly confident about their labels being $L_0$ (taller white bars). This leads to higher influence of $b$ and $c$ during aggregation, and thereby ConfGCN correctly predicting the true label of node $a$ as $L_0$ with high confidence. This clearly demonstrates the benefit of label confidences and their utility in estimating node influences. Graph Attention Networks (GAT) (Veličković et al., 2018), a recently proposed method also provides a mechanism to estimate influences by allowing nodes to attend to their neighborhood. However, as we shall see in Section 6, ConfGCN, through its use of label confidences, is significantly more effective.

Our contributions in this paper are as follows.

- We propose ConfGCN, a Graph Convolutional Network (GCN) framework for semi-supervised learning which models label distribution and their confidences for each node

[2] In this paper, unless otherwise stated, we refer to Kipf-GCN whenever we mention GCN.

in the graph. To the best of our knowledge, this is the first confidence-enabled formulation of GCNs.

- ConfGCN utilize label confidences to estimate influence of one node on another in a label-specific manner during neighborhood aggregation of GCN learning.
- Through extensive evaluation on multiple real-world datasets, we demonstrate ConfGCN effectiveness over state-of-the-art baselines.

ConfGCN's source code and datasets used in the paper are made publicly available[3] to foster reproducible research.

## 2 Related Work

**Semi-Supervised learning (SSL) on graphs:** SSL on graphs is the problem of classifying nodes in a graph, where labels are available only for a small fraction of nodes. Conventionally, the graph structure is imposed by adding an explicit graph-based regularization term in the loss function (Zhu et al., 2003a; Weston et al., 2008; Belkin et al., 2006). Recently, implicit graph regularization via learned node representation has proven to be more effective. This can be done either sequentially or in an end to end fashion. Methods like DeepWalk (Perozzi et al., 2014), node2vec (Grover & Leskovec, 2016), and LINE (Tang et al., 2015) first learn graph representations via sampled random walk on the graph or breadth first search traversal and then use the learned representation for node classification. On the contrary, Planetoid (Yang et al., 2016) learns node embedding by jointly predicting the class labels and the neighborhood context in the graph. Recently, Kipf & Welling (2016) employs Graph Convolutional Networks (GCNs) to learn node representations.

**Graph Convolutional Networks (GCNs):** The generalization of Convolutional Neural Networks to non-euclidean domains is proposed by Bruna et al. (2013) which formulates the spectral and spatial construction of GCNs. This is later improved through an efficient localized filter approximation (Defferrard et al., 2016). Kipf & Welling (2016) provide a first-order formulation of GCNs and show its effectiveness for SSL on graphs. Marcheggiani & Titov (2017) propose GCNs for directed graphs and provide a mechanism for edge-wise gating to discard noisy edges during aggregation. This is further improved by Veličković et al. (2018) which allows nodes to attend to their neighboring nodes, implicitly providing different weights to different nodes. Liao et al. (2018) propose Graph Partition Neural Network (GPNN), an extension of GCNs to learn node representations on large graphs. GPNN first partitions the graph into subgraphs and then alternates between locally and globally propagating information across subgraphs. An extensive survey of GCNs and their applications can be found in Bronstein et al. (2017).

**Confidence Based Methods:** The natural idea of incorporating confidence in predictions has been explored by Li & Sethi (2006) for the task of active learning. Lei (2014) proposes a confidence based framework for classification problems, where the classifier consists of two regions in the predictor space, one for confident classifications and other for ambiguous ones. In representation learning, uncertainty (inverse of confidence) is first utilized for word embeddings by Vilnis & McCallum (2014). Athiwaratkun & Wilson (2018) further extend this idea to learn hierarchical word representation through encapsulation of probability distributions. Orbach & Crammer (2012) propose TACO (Transduction Algorithm with COnfidence), the first graph based method which learns label distribution along with its uncertainty for semi-supervised node classification. Bojchevski & Gnnemann (2018a) embeds graph nodes as Gaussian distribution using ranking based framework which allows to capture uncertainty of representation. They update node embeddings to maintain neighborhood ordering, i.e. 1-hop neighbors are more similar to 2-hop neighbors and so on. Gaussian embeddings have been used for collaborative filtering (Dos Santos et al., 2017) and topic modelling (Das et al., 2015) as well.

## 3 Notation & Problem Statement

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ be an undirected graph, where $\mathcal{V} = \mathcal{V}_l \cup \mathcal{V}_u$ is the union of labeled ($\mathcal{V}_l$) and unlabeled ($\mathcal{V}_u$) nodes in the graph with cardinalities $n_l$ and $n_u$, $\mathcal{E}$ is the set of edges and $\mathcal{X} \in$

[3] ConfGCN's source code: https://goo.gl/qdED2X

$\mathbb{R}^{(n_l+n_u)\times d}$ is the input node features. The actual label of a node $v$ is denoted by a one-hot vector $Y_v \in \mathbb{R}^m$, where $m$ is the number of classes. Given $\mathcal{G}$ and seed labels $Y \in \mathbb{R}^{n_l \times m}$, the goal is to predict the labels of the unlabeled nodes. To incorporate confidence, we additionally estimate label distribution $\boldsymbol{\mu}_v \in \mathbb{R}^m$ and a diagonal co-variance matrix $\boldsymbol{\Sigma}_v \in \mathbb{R}^{m\times m}$, $\forall v \in \mathcal{V}$. Here, $\boldsymbol{\mu}_{v,i}$ denotes the score of label $i$ on node $v$, while $(\boldsymbol{\Sigma}_v)_{ii}$ denotes the variance in the estimation of $\boldsymbol{\mu}_{v,i}$. In other words, $(\boldsymbol{\Sigma}_v^{-1})_{ii}$ is ConfGCN's confidence in $\boldsymbol{\mu}_{v,i}$.

## 4 Background: Graph Convolutional Networks

In this section, we give a brief overview of Graph Convolutional Networks (GCNs) for undirected graphs as proposed by Kipf & Welling (2016). Given a graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ as defined Section 3, the node representation after a single layer of GCN can be defined as

$$\boldsymbol{H} = f((\tilde{\boldsymbol{D}}^{-\frac{1}{2}}(\boldsymbol{A}+\boldsymbol{I})\tilde{\boldsymbol{D}}^{-\frac{1}{2}})\mathcal{X}\boldsymbol{W}) \tag{1}$$

where, $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ denotes the model parameters, $\boldsymbol{A}$ is the adjacency matrix and $\tilde{\boldsymbol{D}}_{ii} = \sum_j (\boldsymbol{A} + \boldsymbol{I})_{ij}$. $f$ is any activation function, we have used ReLU, $f(x) = \max(0, x)$ in this paper. Equation 1 can also be written as

$$\boldsymbol{h}_v = f\left(\sum_{u\in\mathcal{N}_{\mathcal{G}}(v)} \boldsymbol{W}\boldsymbol{h}_u + \boldsymbol{b}\right), \quad \forall v \in \mathcal{V}. \tag{2}$$

Here, $\boldsymbol{b} \in \mathbb{R}^d$ denotes bias, $\mathcal{N}(v) = \{u : \{u, v\} \in \mathcal{E}\}$ corresponds to immediate neighbors of $v$ in graph $\mathcal{G}$ and $\boldsymbol{h}_v$ is the obtained representation of node $v$.

For capturing multi-hop dependencies between nodes, multiple GCN layers can be stacked on top of one another. The representation of node $v$ after $k^{th}$ layer of GCN is given as

$$\boldsymbol{h}_v^{k+1} = f\left(\sum_{u\in\mathcal{N}(v)} \left(\boldsymbol{W}^k\boldsymbol{h}_u^k + \boldsymbol{b}^k\right)\right), \forall v \in \mathcal{V}. \tag{3}$$

where, $\boldsymbol{W}^k, \boldsymbol{b}^k$ denote the layer specific parameters of GCN.

## 5 Confidence Based Graph Convolution (ConfGCN)

Following (Orbach & Crammer, 2012), ConfGCN uses co-variance matrix based symmetric Mahalanobis distance for defining distance between two nodes in the graph. Formally, for any two given nodes $u$ and $v$, with label distributions $\boldsymbol{\mu}_u$ and $\boldsymbol{\mu}_v$ and co-variance matrices $\boldsymbol{\Sigma}_u$ and $\boldsymbol{\Sigma}_v$, distance between them is defined as follows.

$$d_M((\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u), (\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)) = (\boldsymbol{\mu}_u - \boldsymbol{\mu}_v)^T(\boldsymbol{\Sigma}_u^{-1} + \boldsymbol{\Sigma}_v^{-1})(\boldsymbol{\mu}_u - \boldsymbol{\mu}_v).$$

Characteristic of the above distance metric is that if either of $\boldsymbol{\Sigma}_u$ or $\boldsymbol{\Sigma}_v$ has large eigenvalues, then the distance will be low irrespective of the closeness of $\boldsymbol{\mu}_u$ and $\boldsymbol{\mu}_v$. On the other hand, if $\boldsymbol{\Sigma}_u$ and $\boldsymbol{\Sigma}_v$ both have low eigenvalues, then it requires $\boldsymbol{\mu}_u$ and $\boldsymbol{\mu}_v$ to be close for their distance to be low. Given the above properties, we define $r_{uv}$, the influence score of node $u$ on its neighboring node $v$ during GCN aggregation, as follows.

$$r_{uv} = \frac{1}{d_M((\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u), (\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v))}.$$

This influence score gives more relevance to neighboring nodes with highly confident similar label, while reducing importance of nodes with low confident label scores. This results in ConfGCN acquiring anisotropic capability during neighborhood aggregation. For a node $v$, ConfGCN's equation for updating embedding at the $k$-th layer is thus defined as follows.

$$\boldsymbol{h}_v^{k+1} = f\left(\sum_{u\in\mathcal{N}(v)} r_{uv} \times \left(\boldsymbol{W}^k\boldsymbol{h}_u^k + \boldsymbol{b}^k\right)\right), \forall v \in \mathcal{V}. \tag{4}$$

The final node representation obtained from ConfGCN is used for predicting labels of the nodes in the graph as follows.

$$\hat{\boldsymbol{Y}}_v = \mathrm{softmax}(\boldsymbol{W}^K \boldsymbol{h}_v^K + \boldsymbol{b}^K), \; \forall v \in \mathcal{V}$$

where, $K$ denotes the number of ConfGCN's layers. Finally, in order to learn label scores $\{\boldsymbol{\mu}_v\}$ and co-variance matrices $\{\boldsymbol{\Sigma}_v\}$ jointly with other parameters $\{\boldsymbol{W}^k, \boldsymbol{b}^k\}$, following Orbach & Crammer (2012), we include the following three terms in ConfGCN's objective function.

For enforcing neighboring nodes to be close to each other, we include $L_{\text{smooth}}$ defined as

$$L_{\text{smooth}} = \sum_{(u,v)\in\mathcal{E}} (\boldsymbol{\mu}_u - \boldsymbol{\mu}_v)^T (\boldsymbol{\Sigma}_u^{-1} + \boldsymbol{\Sigma}_v^{-1})(\boldsymbol{\mu}_u - \boldsymbol{\mu}_v).$$

To impose the desirable property that the label distribution of nodes in $\mathcal{V}_l$ should be close to their input label distribution, we incorporate $L_{\text{label}}$ defined as

$$L_{\text{label}} = \sum_{v\in\mathcal{V}_l} (\boldsymbol{\mu}_v - \boldsymbol{Y}_v)^T (\boldsymbol{\Sigma}_v^{-1} + \frac{1}{\gamma}\boldsymbol{I})(\boldsymbol{\mu}_v - \boldsymbol{Y}_v).$$

Here, for input labels, we assume a fixed uncertainty of $\frac{1}{\gamma}\boldsymbol{I} \in \mathbb{R}^{L\times L}$, where $\gamma > 0$. We also include the following regularization term, $L_{\text{reg}}$ to constraint the co-variance matrix to be finite and positive.

$$L_{\text{reg}} = \sum_{v\in\mathcal{V}} \mathrm{Tr}\boldsymbol{\Sigma}_v - \eta \sum_{v\in\mathcal{V}} \log(\det\boldsymbol{\Sigma}_v),$$

for some $\eta > 0$. The first term increases monotonically with the eigenvalues of $\boldsymbol{\Sigma}$ and the second term prevents them from becoming zero. Additionally in ConfGCN, we include the $L_{\text{const}}$ in the objective, to push the label distribution ($\boldsymbol{\mu}$) close to the final model prediction ($\hat{\boldsymbol{Y}}$).

$$L_{\text{const}} = \sum_{v\in\mathcal{V}} (\boldsymbol{\mu}_v - \hat{\boldsymbol{Y}}_v)^T (\boldsymbol{\mu}_v - \hat{\boldsymbol{Y}}_v).$$

Finally, we include the standard cross-entropy loss for semi-supervised multi-class classification over all the labeled nodes ($\mathcal{V}_l$).

$$L_{\text{cross}} = -\sum_{v\in\mathcal{V}_l} \sum_{j=1}^{m} \boldsymbol{Y}_{vj} \log(\hat{\boldsymbol{Y}}_{vj}).$$

The final objective for optimization is the linear combination of the above defined terms.

$$L(\{\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v\}) = L_{\text{cross}} + \lambda_1 L_{\text{smooth}} + \lambda_2 L_{\text{label}} + \lambda_3 L_{\text{const}} + \lambda_4 L_{\text{reg}} \tag{5}$$

where, $\lambda_i \in \mathbb{R}$, are the weights of the terms in the objective. We optimize $L(\{\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v\})$ using stochastic gradient descent. We hypothesize that all the terms help in improving ConfGCN's performance and we validate this in Section 7.4.

# 6 Experiments

## 6.1 Datasets

For evaluating the effectiveness of ConfGCN, we evaluate on several semi-supervised classification benchmarks. Following the experimental setup of (Kipf & Welling, 2016; Liao et al., 2018), we evaluate on Cora, Citeseer, and Pubmed datasets (Sen et al., 2008). The dataset statistics is summarized in Table 1. Label mismatch denotes the fraction of edges between nodes with different labels in the training data. The benchmark datasets commonly used for semi-supervised classification task have substantially low label mismatch rate. In order to examine models on datasets with more heterogeneous neighborhoods, we also evaluate on Cora-ML dataset (Bojchevski & Gnnemann, 2018b).

All the four datasets are citation networks, where each document is represented using bag-of-words features in the graph with undirected citation links between documents. The goal is to classify documents into one of the predefined classes. We use the data splits used by (Yang et al., 2016) and follow similar setup for Cora-ML dataset. Following (Kipf & Welling, 2016), additional 500 labeled nodes are used for hyperparameter tuning.

| Dataset | Nodes | Edges | Classes | Features | Label Mismatch | $\frac{\lvert\mathcal{V}_l\rvert}{\lvert\mathcal{V}\rvert}$ |
|---|---|---|---|---|---|---|
| Cora | 2,708 | 5,429 | 7 | 1,433 | 0.002 | 0.052 |
| Cora-ML | 2,995 | 8,416 | 7 | 2,879 | 0.046 | 0.166 |
| Citeseer | 3,327 | 4,372 | 6 | 3,703 | 0.003 | 0.036 |
| Pubmed | 19,717 | 44,338 | 3 | 500 | 0.0 | 0.003 |

Table 1: Details of the datasets used in the paper. Please refer Section 6.1 for more details.

| Method | Citeseer | Cora | Pubmed | Cora ML |
|---|---|---|---|---|
| LP (Zhu et al., 2003a) | 45.3 | 68.0 | 63.0 | - |
| ManiReg (Belkin et al., 2006) | 60.1 | 59.5 | 70.7 | - |
| SemiEmb (Weston et al., 2008) | 59.6 | 59.0 | 71.1 | - |
| Feat (Yang et al., 2016) | 57.2 | 57.4 | 69.8 | - |
| DeepWalk (Perozzi et al., 2014) | 43.2 | 67.2 | 65.3 | - |
| GGNN (Li et al., 2015) | 68.1 | 77.9 | 77.2 | - |
| Planetoid (Yang et al., 2016) | 64.9 | 75.7 | 75.7 | - |
| Kipf-GCN (Kipf & Welling, 2016) | 70.3 | 81.5 | 79.0 | 51.6 |
| G-GCN (Marcheggiani & Titov, 2017) | 71.1 | 82.0 | 77.3 | 50.4 |
| GPNN (Liao et al., 2018) | 69.7 | 81.8 | 79.3 | 60.6 |
| GAT (Veličković et al., 2018) | 72.5 | 83.0 | 79.0 | 54.9 |
| ConfGCN (this paper) | **73.9** | **83.5** | **80.7** | **80.9** |

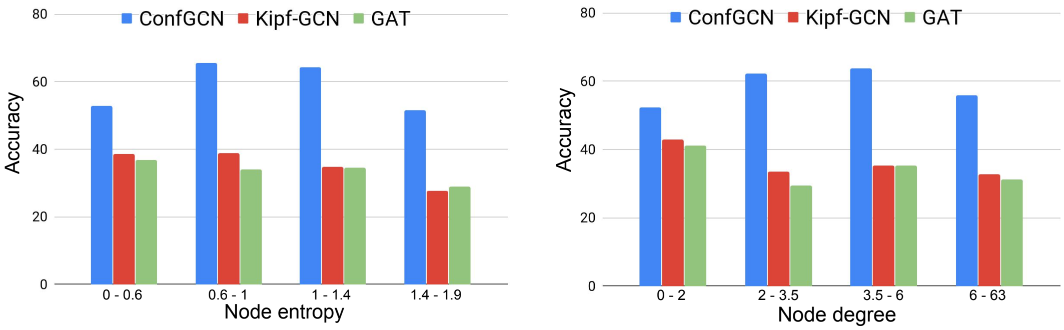
Table 2: Performance comparison of several methods for semi-supervised node classification on multiple benchmark datasets. ConfGCN performs consistently better across all the datasets. Baseline method performances on Citeseer, Cora and Pubmed datasets are taken from Liao et al. (2018); Veličković et al. (2018). We consider only the top performing baseline methods on these datasets for evaluation on the Cora-ML dataset. Please refer Section 7.1 for details.

**Hyperparameters:** We use the same data splits as described in (Yang et al., 2016), with a test set of 1000 labeled nodes for testing the prediction accuracy of ConfGCN and a validation set of 500 labeled nodes for optimizing the hyperparameters. The model is trained using Adam (Kingma & Ba, 2014) with a learning rate of 0.01. The weight matrices along with $\mu$ are initialized using Xavier initialization (Glorot & Bengio, 2010) and $\mathbf{\Sigma}$ matrix is initialized with identity.

### 6.2 Baselines

For evaluating ConfGCN, we compare against the following baselines:

- **Feat** (Yang et al., 2016) takes only node features as input and ignores the graph structure.
- **ManiReg** (Belkin et al., 2006) is a framework for providing data-dependent geometric regularization.
- **SemiEmb** (Weston et al., 2008) augments deep architectures with semi-supervised regularizers to improve their training.
- **LP** (Zhu et al., 2003a) is an iterative iterative label propagation algorithm which propagates a nodes labels to its neighboring unlabeled nodes according to their proximity.
- **DeepWalk** (Perozzi et al., 2014) learns node features by treating random walks in a graph as the equivalent of sentences.
- **Planetoid** (Yang et al., 2016) provides a transductive and inductive framework for jointly predicting class label and neighborhood context of a node in the graph.
- **GCN** (Kipf & Welling, 2016) is a variant of convolutional neural networks used for semi-supervised learning on graph-structured data.
- **G-GCN** (Marcheggiani & Titov, 2017) is a variant of GCN with edge-wise gating to discard noisy edges during aggregation.
- **GGNN** (Li et al., 2015) is a generalization of RNN framework which can be used for graph-structured data.



(a)

(b)

Figure 2: Plots of node classification accuracy vs. (a) node entropy and (b) node degree. On $x$-axis we have quartiles of (a) node entropy and (b) degree, i.e., each bin has 25% of the samples in sorted order. Overall, we observe that the performance of Kipf-GCN and GAT degrades with the increase in node entropy and degree. In contrast, ConfGCN is able to avoid such degradation due to its estimation and use of confidence scores. Refer Section 7.2 for details.

- **GPNN** (Liao et al., 2018) is a graph partition based algorithm which propagates information after partitioning large graphs into smaller subgraphs.
- **GAT** (Veličković et al., 2018) is a graph attention based method which provides different weights to different nodes by allowing nodes to attend to their neighborhood.

# 7 RESULTS

In this section, we attempt to answer the following questions:

Q1. How does ConfGCN compare against the existing methods semi-supervised node classification task? (Section 7.1)

Q2. How does the performance of methods vary with increasing node degree and label mismatch? (Section 7.2)

Q3. What is the effect of ablating different terms in ConfGCN's loss function? (Section 7.4)

Q4. How does increasing the number of layers effects ConfGCN's performance? (Section 7.3)

## 7.1 NODE CLASSIFICATION

The evaluation results for semi-supervised node classification are summarized in Table 2. Results of all other baseline methods on Cora, Citeseer and Pubmed datasets are taken from (Liao et al., 2018; Veličković et al., 2018) directly. Overall, we find that ConfGCN outperforms all existing approaches consistently across all the datasets. We observe that on the more noisy and challenging Cora-ML dataset, ConfGCN performs considerably better, giving nearly 20% absolute increase in accuracy compared to the previous state-of-the-art method. This can be attributed to ConfGCN's ability to model nodes' label distribution along with the confidence scores which subdues the effect of noisy nodes during neighborhood aggregation. The lower performance of G-GCN compared to Kipf-GCN on Cora-ML shows that calculating edge-wise gating scores using the hidden representation of nodes is not much helpful in suppressing noisy neighborhood nodes as the representations lack label information or are over averaged or unstable. Similar reasoning holds for GAT for its poor performance on Cora-ML.

## 7.2 EFFECT OF NODE ENTROPY AND DEGREE ON PERFORMANCE

In this section, we provide an analysis of the performance of Kipf-GCN, GAT and ConfGCN for node classification on Cora-ML dataset which has higher label mismatch rate. We use neighborhood

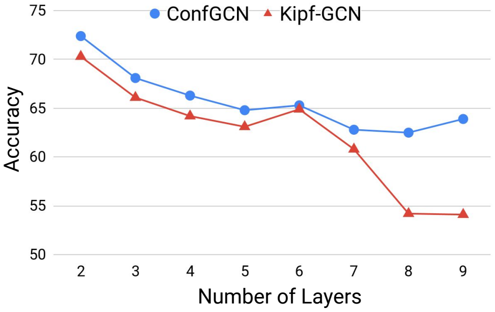


Figure 3: Evaluation of Kipf-GCN and ConfGCN on the citeseer dataset with increasing number of GCN layers. Overall, ConfGCN outperforms Kipf-GCN, and while both methods' performance degrade with increasing layers, ConfGCN's degradation is more gradual than Kipf-GCN's abrupt drop. Please see Section 7.3 for details.

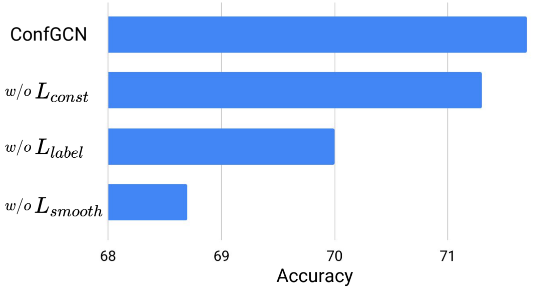


Figure 4: Performance comparison of different ablated version of ConfGCN on citeseer dataset. These results justify inclusion of the different terms in ConfGCN's loss function. Please see Section 7.4 for details.

label entropy to quantify label mismatch, which for a node $u$ is defined as follows.

$$\mathrm{NeighborLabelEnttropy}(u) = -\sum_{l=1}^{L} p_{ul} \log p_{ul}; \text{ where, } p_{ul} = \frac{|\{v \in \mathcal{N}(u) \mid \mathrm{label}(v) = l\}|}{|\mathcal{N}(u)|}.$$

Here, $\mathrm{label}(v)$ is the true label of node $v$. The neighborhood label entropy of a node increases with label mismatch amongst its neighbors. The problem of node classification becomes difficult with increase in node degree, therefore, we also evaluate the performance of methods with increasing node degree. The results are summarized in Figures 2a and 2b. We find that the performance of both Kipf-GCN and GAT decreases with increase in node entropy and degree. On the contrary, ConfGCN's performance remains consistent and does not degrade with increase in entropy or degree. This shows that ConfGCN is able to use the label distributions and confidence effectively to subdue irrelevant nodes during aggregation.

### 7.3 Effect of Increasing Convolutional Layers

Recently, Xu et al. (2018) highlighted an unusual behavior of Kipf-GCN where its performance degraded significantly with increasing number of layers. For comparison, we evaluate the performance of Kipf-GCN and ConfGCN on citeseer dataset with increasing number of convolutional layers. The results are summarized in Figure 3. We observe that Kipf-GCN's performance degrades drastically with increasing number of layers, whereas ConfGCN's decrease in performance is more gradual. We also note that ConfGCN outperforms Kipf-GCN at all layer levels.

### 7.4 Ablation Results

In this section, we evaluate the different ablated version of ConfGCN by cumulatively eliminating terms from its objective function as defined in Section 5. The results on citeseer dataset are summarized in Figure 4. Overall, we find that ConfGCN performs best when all the terms in its loss function (Equation 5) are included.

## 8 Conclusion

In this paper we present ConfGCN, a confidence based Graph Convolutional Network which estimates label scores along with their confidences jointly in GCN-based setting. In ConfGCN, the influence of one node on another during aggregation is determined using the estimated confidences and label scores, thus inducing anisotropic behavior to GCN. We demonstrate the effectiveness of ConfGCN against recent methods for semi-supervised node classification task and analyze its performance in different settings. We make ConfGCN's source code available.

## References

Ben Athiwaratkun and Andrew Gordon Wilson. On modeling hierarchical data via probabilistic order embeddings. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJCXZQbAZ.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1957–1967. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1209.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, December 2006. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1248547.1248632.

Aleksandar Bojchevski and Stephan Gnnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=r1ZdKJ-0W.

Aleksandar Bojchevski and Stephan Gnnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=r1ZdKJ-0W.

M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. ISSN 1053-5888. doi: 10.1109/MSP.2017.2693418.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013. URL http://arxiv.org/abs/1312.6203.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 795–804. Association for Computational Linguistics, 2015. doi: 10.3115/v1/P15-1077. URL http://www.aclweb.org/anthology/P15-1077.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR*, abs/1606.09375, 2016. URL http://arxiv.org/abs/1606.09375.

Ludovic Dos Santos, Benjamin Piwowarski, and Patrick Gallinari. Gaussian embeddings for collaborative filtering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pp. 1065–1068, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080722. URL http://doi.acm.org/10.1145/3077136.3080722.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1243–1252. IEEE computer Society, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Jing Lei. Classification with confidence. *Biometrika*, 101(4):755–769, 2014. doi: 10.1093/biomet/asu038. URL http://dx.doi.org/10.1093/biomet/asu038.

Mingkun Li and Ishwar K. Sethi. Confidence-based active learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1251–1261, August 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.156. URL https://doi.org/10.1109/TPAMI.2006.156.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

Renjie Liao, Marc Brockschmidt, Daniel Tarlow, Alexander Gaunt, Raquel Urtasun, and Richard S. Zemel. Graph partition neural networks for semi-supervised classification, 2018. URL https://openreview.net/forum?id=rk4Fz2e0b.

Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1506–1515. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1159.

Thien Nguyen and Ralph Grishman. Graph convolutional networks with argument-aware pooling for event detection, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16329.

Matan Orbach and Koby Crammer. Graph-based transduction with confidence. In *ECML/PKDD*, 2012.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 701–710, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9. doi: 10.1145/2623330.2623732. URL http://doi.acm.org/10.1145/2623330.2623732.

Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008. URL http://www.cs.iit.edu/~ml/pdfs/sen-aimag08.pdf.

Amarnag Subramanya and Jeff Bilmes. Semi-supervised learning with measure propagation. *J. Mach. Learn. Res.*, 12:3311–3370, November 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1953048.2078212.

Amarnag Subramanya and Partha Pratim Talukdar. Graph-based semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(4):1–125, 2014.

Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ECML PKDD '09, pp. 442–457, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-04173-0. doi: 10.1007/978-3-642-04174-7_29. URL http://dx.doi.org/10.1007/978-3-642-04174-7_29.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*. ACM, 2015.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. Reside: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '18)*, Brussels, Belgium, Oct 31-Nov 4 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ. accepted as poster.

Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *CoRR*, abs/1412.6623, 2014. URL http://arxiv.org/abs/1412.6623.

Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 1168–1175, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390303. URL http://doi.acm.org/10.1145/1390156.1390303.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/xu18c.html.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 40–48. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045396.

Li Yi, Hao Su, Xingwen Guo, and Leonidas Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *arXiv preprint arXiv:1612.00606*, 2016.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '18)*, Brussels, Belgium, Oct 31-Nov 4 2018.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 912–919. AAAI Press, 2003a. ISBN 1-57735-189-4. URL http://dl.acm.org/citation.cfm?id=3041838.3041953.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 912–919. AAAI Press, 2003b. ISBN 1-57735-189-4. URL http://dl.acm.org/citation.cfm?id=3041838.3041953.