

# Shoestring: Graph-Based Semi-Supervised Classification with Severely Limited Labeled Data

Wanyu Lin, Zhaolin Gao, Baochun Li  
University of Toronto

{wanyu.lin, zhaolin.gao}@mail.utoronto.ca bli@ece.toronto.edu

## Abstract

*Graph-based semi-supervised learning has been shown to be one of the most effective classification approaches, as it can exploit connectivity patterns between labeled and unlabeled samples to improve learning performance. However, we show that existing techniques perform poorly when labeled data are severely limited. To address the problem of semi-supervised learning in the presence of severely limited labeled samples, we propose a new framework, called Shoestring<sup>1</sup>, that incorporates metric learning into the paradigm of graph-based semi-supervised learning. In particular, our base model consists of a graph embedding network, followed by a metric learning network that learns a semantic metric space to represent the semantic similarity between the sparsely labeled and large numbers of unlabeled samples. Then the classification can be performed by clustering the unlabeled samples according to the learned semantic space. We empirically demonstrate Shoestring's superiority over many baselines, including graph convolutional networks, label propagation and their recent label-efficient variations (IGCN and GLP). We show that our framework achieves state-of-the-art performance for node classification in the low-data regime. In addition, we demonstrate the effectiveness of our framework on image classification tasks in the few-shot learning regime, with significant gains on miniImageNet (2.57% ~ 3.59%) and tieredImageNet (1.05% ~ 2.70%).*

## 1. Introduction

The availability of large quantities of labeled samples has made it possible for deep learning to achieve remarkable performance breakthroughs in speech recognition, natural language processing, and computer vision [6, 18]. How-

ever, the reliance on large amounts of labeled samples increases the burden of data collection, making it difficult to apply deep learning to the low-data regime where labeled samples are extremely rare and are difficult to collect.

With semi-supervised learning (SSL), small amounts of labeled samples are used with a relatively large number of unlabeled samples for classification. Among existing semi-supervised learning models, graph-based methods, such as graph convolutional networks and label propagation, have been demonstrated as one of the most effective approaches for semi-supervised classification, as they are capable of exploiting the connectivity patterns between labeled and unlabeled samples to improve classification performance. Given their advantages, in previous work on few-shot image classification, quick knowledge from a few samples is acquired by considering relationships between instances and representing the data into a graph [4, 7, 8, 13].

Yet, even with such graph-based semi-supervised learning, model learning performance degrades quickly with a diminishing number of labeled samples per class [11]. The performance degradation can be explained as follows. In general, labels work as “anchors,” and are used to force the learning models to fit these labeled samples with certain confidence, so that the information extracted from them can be reliably propagated to unlabeled samples. However, when the labeled samples are severely limited, there is a good chance they will exhibit a large testing error even though their training error is small—i.e., overfitting these limited labeled data. Taking graph convolutional networks as an example, they indeed lead to state-of-the-art accuracies on node classification tasks with two convolutional layers in the presence of a sufficient amount of labeled samples. However, when only a few labeled samples are given, it would not be able to effectively propagate the labels to the entire data graph [11].

Nevertheless, humans are exceptional learners capable of generalizing their learned knowledge to novel concepts, and capable of learning from very few examples. In this paper, we aim to tackle the problem of graph-based semi-supervised learning where labeled data are *severely limited*.

<sup>1</sup>Code will be made publicly available.

This research was supported in part by the NSERC Discovery Research Program.

There has been a major push in recent research, particularly on the image classification task, towards generalizing deep learning models to learn tasks in a data-efficient way through few-shot learning.

Among the best-performing methods (e.g., gradient-based [3], metric-learning based [16, 19] and model-based [15]) for few-shot learning, metric-learning approaches have been demonstrated as one of the simplest and most efficient methods in the few-shot setting. Metric-learning methods aim to optimize the transferable embeddings by learning a distance-based prediction rule over the embeddings. Motivated by this finding, in addition to exploiting the connectivity patterns between labeled and unlabeled samples, we seek to transfer as much knowledge as possible from limited labeled samples to a large number of unlabeled samples in the embedding space.

The main contribution in our proposed framework, called *Shoestring*, is that it is the first to incorporate a metric learning network into the settings of graph-based semi-supervised learning. It is simple yet effective that can be applied to boost the learning performance of typical graph-based semi-supervised learning methods. In essence, our framework is proposed based on the idea that in the low-dimensional semantic space, there exists an embedding in which points cluster around a single prototype representation for each class. More specifically, *Shoestring* jointly learns a non-linear mapping of each instance into a semantic space using a graph embedding network, and learns a metric space with a metric learning network to represent the semantic similarity between the labeled and unlabeled samples. Classification, for an embedded unlabeled sample, is then performed by finding its nearest class prototype based on the learned semantic metric.

Highlights of our original contributions are as follows. *First*, to verify the effectiveness of our framework, we revisited several graph-based semi-supervised learning models, such as graph convolutional networks, label propagation and their recent label-efficient variations proposed from the perspective of graph filters (IGCN and GLP), and empirically demonstrate the superiority of our framework over these baselines. We show that our framework leads to state-of-the-art node classification performance in the low-data regime, by incorporating these graph learning models as the base models. *Second*, we empirically analyze the underlying distance functions used in the metric learning network, such as cosine similarity and squared Euclidean distance. We find that the choice of a similarity metric is critical, as the performance of different metrics varies from different datasets as well as various label rates. *Finally*, we demonstrate the effectiveness of *Shoestring* on image classification tasks in the few-shot learning regime, and achieve state-of-the-art results on miniImageNet and tieredImageNet.

## 2. Problem Setup

We consider the task of semi-supervised node classification on graphs. Formally, a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{X})$  is given with  $n = |\mathcal{V}|$  vertices, where  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$  is the set of vertices,  $\mathcal{A} \in \{0, 1\}^{n \times n}$  is the adjacency matrix representing the connections, and  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}^T \in \mathcal{R}^{n \times m}$  is the feature matrix of vertices, and  $x_i \in \mathcal{R}^m$  is the  $m$ -dimensional feature vector of vertex  $v_i$ .

We follow the standard semi-supervised classification setting, which is commonly employed in various literature [1, 9]. Given a set of labeled nodes  $\mathcal{V}_l \subset \mathcal{V}$ , with class labels from  $\mathcal{Y} = \{y_1, y_2, y_3, \dots, y_K\}$  and a set of unlabeled nodes  $\mathcal{V}_u \subset \mathcal{V} \setminus \mathcal{V}_l$ , the goal of node classification is to map each node  $v \in \mathcal{V}$  to one class in  $\mathcal{Y}$ . We assume that the data domain is sparsely labeled so that the number of node-label pairs is much smaller than the number of unlabeled nodes,  $|\mathcal{V}_l| \ll |\mathcal{V}_u|$ . In particular, we are especially interested in cases where  $|\mathcal{V}_l|$  is severely limited, e.g., 1 or 2 labeled samples per class which may arise in situations where obtaining an unlabeled sample is cheap and easy for novel classes, while labeling the sample is expensive or difficult. Our ultimate goal is to produce an effective classifier for semi-supervised node classification on graphs, for which only very few labeled samples are available.

## 3. Revisiting Graph-based Semi-Supervised Learning

We do not attempt to provide a comprehensive literature review on graph-based semi-supervised learning. Instead, we selectively provide the baseline methods adopted by top performers on node classification tasks, such as graph convolutional networks and label propagation, either in terms of their simplicity or expressiveness. Furthermore, we think that these methods are of great value, not the least because they lead to state-of-the-art node classification with small numbers of labeled data in the literature and can readily be applied to image classification tasks in the few-shot learning regime [4, 13]. As prototypical examples, let us consider semi-supervised classification with graph convolutional networks [9] and label propagation methods [1, 21, 22].

*Graph convolutional networks:* Graph convolutional neural networks (GCNs) is a generalization of traditional convolutional neural networks to the graph domain. In [9], the GCN model applied for semi-supervised classification is a two-layer GCN followed by a softmax classifier on the output features:

$$Z = \mathbf{softmax}(\hat{\mathcal{A}}\mathbf{ReLU}(\hat{\mathcal{A}}\mathcal{X}\Theta^{(0)})\Theta^{(1)}) \quad (1)$$

where  $\tilde{\mathcal{A}} = \mathcal{A} + I$ ,  $\tilde{\mathcal{D}}_{ii} = \sum_j \tilde{\mathcal{A}}_{ij}$ ,  $\hat{\mathcal{A}} = \tilde{\mathcal{D}}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{\mathcal{D}}^{-\frac{1}{2}}$ ,  $\mathbf{softmax}(x_i) = \frac{1}{Z} \exp(x_i)$  with  $Z = \sum_i \exp(x_i)$ . The

optimization loss function is defined as the cross-entropy error over all labeled samples:

$$\mathcal{L}_{ce} = - \sum_{i \in \mathcal{V}_l} \sum_{k=1}^K Y_{ik} \ln Z_{ik} \quad (2)$$

where  $\mathcal{V}_l$  is the set of node indices that have labels, and  $K$  is the number of classes/labels.

**Label propagation:** Label propagation is a simple and effective principle of using the graph structure to spread labels from labeled samples to the entire data set. Starting with nodes with their known labels, each node starts to propagate its label to its neighbors, and the process is repeated until convergence. Due to its simplicity and effectiveness, there are several variations in the literature [1, 21, 22] and have been widely used in many scientific research fields and numerous industrial applications. An alternative method originating from smoothness considerations yields algorithms based on graph regularization, which naturally leads to a regularization term based on the graph Laplacian. Formally, the objective is to find an embedding matrix  $Z$  that agrees with the label matrix  $Y$  while being smooth on the graph such that nearby vertices have similar embeddings.

$$Z = \operatorname{argmin}\{\|Z - Y\|_2^2 + \alpha \operatorname{Tr}(Z^T LZ)\} \quad (3)$$

where  $L = D - \mathcal{A}$  is the graph Laplacian,  $D$  is the degree matrix, and  $\alpha$  is a parameter controlling the degree of Laplacian regularization. Then a closed-form solution can be obtained by taking the derivative of the objective function and setting it to zero.

**Analysis** In essence, for semi-supervised learning to work, a certain assumption, called the *smoothness assumption*, has to hold. It implies that if two inputs  $x_1, x_2$  in a high-density region are close, then so should be the corresponding outputs  $y_1, y_2$ . Semi-supervised GCN and label propagation methods have been proved to perform very well on many classification tasks.

These can be explained as follows. For GCN, graph convolution is a special form of Laplacian smoothing, which computes the new representation of a vertex by averaging over itself and its neighbors. Regarding label propagation, the second term of its objective function is a regularization term motivated by the smoothness assumption. When the number of labeled samples is large enough, both GCN and label propagation can effectively learn the shape of the manifolds near which the data concentrate in the embedding space, leading to superior performance on node classification tasks.

**Why do these methods fail?** Graph convolutional networks and label propagation essentially fall into the category of local learning algorithms in semi-supervised learning, relying on a neighborhood graph to approximate manifolds near which the data density is assumed to concentrate.

Table 1: Classification accuracy on Cora and CiteSeer, with one labeled sample per class (%). The performance of graph filtering-based variations are insignificant in the severely low-data regime.

DATASET	GCN	IGCN <sub>(RNM)</sub>	IGCN <sub>(AR)</sub>	LP	GLP <sub>(RNM)</sub>	GLP <sub>(AR)</sub>
CORA	39.5	41.5	42.3	43.6	38.4	37.7
CITeseer	34.1	33.1	33.0	30.6	37.0	37.4

When there are only a few labeled samples, one cannot generalize properly and the model performance degrades very quickly.

**Graph filtering-based variations of GCN and LP with severely limited labeled samples.** [12] aims to address the problem of label efficient semi-supervised learning from the perspective of graph filtering. They proposed a framework that draws graph structure into data features by taking them as signals on the graph and applying a low-pass graph filter to extract data representations for downstream classification tasks. Indeed, it can achieve label efficiency, to some extent, by adjusting the strength of the graph filter. Under this framework, generalized label propagation (GLP) and improved graph convolutional networks (IGCN) were proposed with two types of variations respectively, either relying on the renormalization (RNM) filter or the auto-regressive (AR) filter.

We evaluated the task of document classification with different semi-supervised learning methods on Cora [14] and CiteSeer [5] respectively, each of which has one labeled sample per class. The results are shown in Table 1. We observed that with severely limited labeled samples, the performance of graph filtering-based variations are non-significant. More specifically, IGCN performs worse than GCN on CiteSeer, while GLP leads to degraded performance on Cora. In this paper, we are interested to exploit the intrinsic structure of the data to boost classification accuracy with further gains when the number of labeled samples is severely limited.

## 4. Proposed Framework

In this section, we introduce our framework, called *Shoestring*, to address the problem of graph-based semi-supervised learning in the presence of severely limited labeled samples. The architecture of Shoestring is illustrated in Fig. 1, which composed of two modules: a typical graph-based semi-supervised learning module/a graph embedding network to learn a non-linear mapping of each instance into an embedding vector (e.g., a two-layer GCN or label propagation module), and a metric learning module that learns the semantic similarity between each node representation

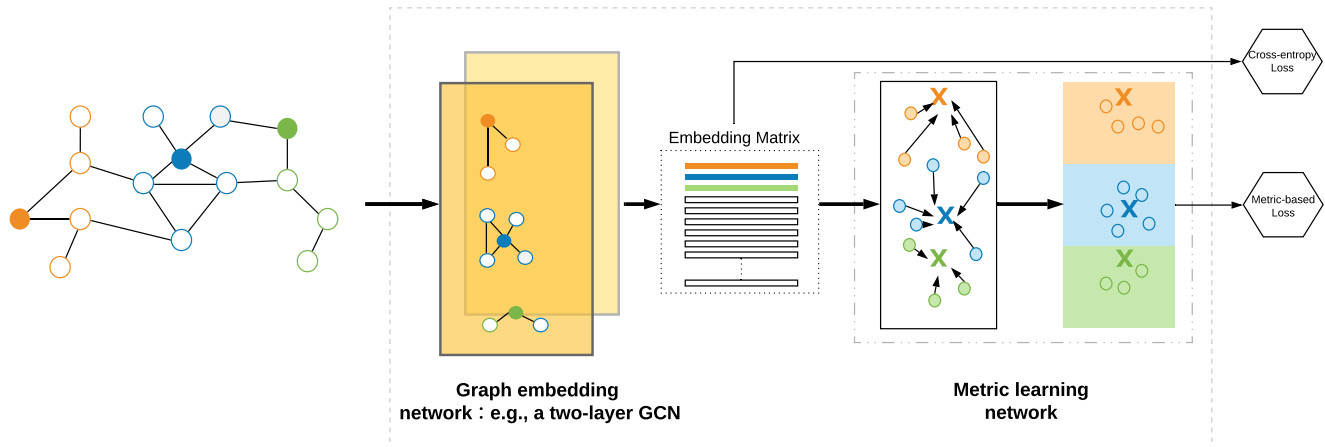


Figure 1: Illustration of the *Shoestring* framework: As an example, the input graph contains three types of node labels in three different colors, where the solid nodes are labeled and the rest are unlabeled. (1) A graph embedding network (a typical graph-based semi-supervised learning module, e.g., GCN) to learn a non-linear mapping of each node into an embedding vector; (2) A metric learning network to learn pair-wise similarity between each node representation and the centroid representation of each class in the low-dimensional semantic space. To optimize *Shoestring*, the full pipeline in our architecture is used.

and the centroid representation of each class in the low-dimensional semantic space. For simplicity, we first take semi-supervised GCNs as our prototypical model to illustrate our framework. We shall also discuss how to fit other semi-supervised learning methods into our framework, such as recent variations with graph filtering, IGCN and GLP.

Before we present our proposed framework, we first introduce the *manifold assumption* and *cluster assumption* [2], which are different from the smoothness assumption, but form the basis of our construction. The *manifold assumption* forms the basis of several semi-supervised learning methods in the literature, which indicates that the high-dimensional data lie on a low-dimensional manifold. The *cluster assumption* is one of the earliest forms of semi-supervised learning, which implies if data points/nodes are in the same cluster, they are likely to be of the same class.

As we discussed previously, the design basis of graph convolutional networks is the smoothness assumption (Laplacian smoothing). Its superior performance on semi-supervised classification tasks with sufficient labeled samples can also be interpreted as follows. The two-layer convolutional transformation tends to encourage the graph representations to lie on a low-dimensional manifold, such that the nodes can be classified distinctly in the embedding space. Motivated by this intuition, we seek to exploit the intrinsic structure of the data distribution in the embedding space, while the semi-supervised classification task is performed under fairly limited numbers of labeled samples.

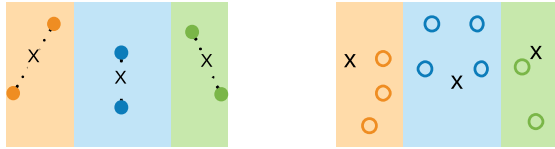
More specifically, the first component of *Shoestring* is a classical graph-based semi-supervised learning module,

a two-layer GCN in our prototypical example, which is able to inject the graph structure into data representations by convolutional operations. With this transformation, the graph representations of the data are encouraged to lie on a low-dimensional manifold. In addition, we exploit a metric learning network that is able to learn a semantic metric space to represent the semantic similarity between the sparsely labeled and large numbers of unlabeled samples. Label assignments, for the unlabeled samples, are performed through transferring the semantic knowledge of the labeled samples.

Our metric learning network includes a similarity network to learn the semantic similarity between each node representation and the centroid representation of each class (the colored cross sign in Fig. 1). In particular, the per-class centroid is the element-wise mean of its labeled samples in the embedding space (the output of the graph embedding network), shown in Fig. 2a:

$$c_{y_k} = \frac{1}{|\mathcal{V}_k|} \sum_{(x_l, y_l) \in \mathcal{V}_k} Z_{\Theta, x_l} \quad (4)$$

where  $\mathcal{V}_k \subset \mathcal{V}_l$  and  $Z_{\Theta, x_l}$  is the embedding vector of node  $x_l$ . Therefore, the output of the metric learning module contains the similarity values of each node to each class. Followed with a softmax (the output layer), the label of each unlabeled sample can be assigned to the class with the highest similarity value (its nearest class centroid), shown in Fig. 2b:



(a) Per class centroid.

(b) Label assignment.

Figure 2: Our metric learning network for semantic similarity learning and label assignments: (a) Three types of solid nodes are labeled samples in three classes, respectively. The class centroid of each class is calculated by the element-wise mean of the labeled samples in each class, as shown in cross signs. (b) Each unlabeled sample is assigned to the label of its nearest class centroid in the learned metric space.

$$p_{\Theta}(i = k|x_l) = \frac{\exp[\mathbf{sim}(z_{\Theta, x_l}, c_{y_k})]}{\sum_{i=1}^K \exp[\mathbf{sim}(z_{\Theta, x_l}, c_{y_i})]} \quad (5)$$

where  $\mathbf{sim}$  is the distance function for similarity measurement in the low-dimensional embedding space. The underlying design intuition is that in the embedding space, the graph representations tend to lie on a low-dimensional manifold, in which closely clustered node representations tend to be assigned similar labels (the ‘‘cluster assumption’’). The underlying similarity function can be cosine similarity, or negative square Euclidean distance, etc. We will show the choice of a similarity metric is critical, where the performance varies from different datasets as well as various label rates.

**Objective function of *Shoestring*.** To optimize *Shoestring*, the full pipeline in our architecture is used. There are two components in our objective function. (1) The typical graph-based semi-supervised learning loss. (2) The metric-based learning loss. More specifically, in the GCN module, the first term is the cross-entropy loss as defined in Eq. (2), while the second term is the metric-based cross-entropy loss.

Assume we have  $|\mathcal{V}_k|$  labeled samples from each class in the target domain. We compute the centroid representation  $c_{y_k}$  for each class, by taking the element-wise mean of the  $|\mathcal{V}_k|$  labeled samples, in the embedding space (the output of the classical graph embedding network). Thus, we can have the similarity vector for each labeled sample, where the  $k$ th element is the similarity between this sample and the centroid of class  $c_{y_k}$ . Therefore, the metric-based loss<sup>2</sup>

<sup>2</sup>Here we use term ‘‘metric-based loss’’ to differentiate the metric-based cross-entropy loss from the cross-entropy defined in Eq. (2).

can be formulated as:

$$\mathcal{L}_{\text{me}} = - \sum_{(x_l, y_l) \in \mathcal{V}_l} \log \frac{\exp[\mathbf{sim}(z_{x_l}, c_{y_l})]}{\sum_{i=1}^K \exp[\mathbf{sim}(z_{x_l}, c_{y_i})]} \quad (6)$$

Formally, the objective function of *Shoestring* is defined as follows:

$$\mathcal{L}_{\text{Shoestring}} = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{me}} \quad (7)$$

where  $\lambda$  is to control the degree of metric-based learning loss. After the optimization, *Shoestring* uses the forward propagation through the graph embedding network and the metric learning network, followed by a softmax (the output layer) to obtain the final label assignment.

Our proposed framework is fairly general that can be used to further boost the classification performance of several graph-based semi-supervised learning methods, while the number of labeled samples are severely limited. In particular, to fit label propagation, the label-efficient variations with graph filtering (IGCN, GLP) into our framework, we can just simply replace the graph embedding network module with any of these methods. In the experimental section, we will show empirically that *Shoestring* can indeed dramatically improve the classification accuracy of these methods, especially when there are only a few labeled samples.

Table 2: Statistics description of citation networks.

DATASET	NODES	EDGES	CLASSES	FEATURES
CORA	2,708	5,429	7	1,433
CITeseer	3,327	4,732	6	3,703
PUBMED	19,717	44,338	3	500
LARGE CORA	11,881	64,898	10	3,780

## 5. Experiments

We evaluate and compare *Shoestring* with state-of-the-art methods on semi-supervised document classification in citation networks, as well as a few-shot learning task for image classification on two datasets, e.g., miniImageNet and tieredImageNet.

### 5.1. Performance Evaluation on Citation Networks

*Datasets.* Following [11, 12], we select four citation networks: Cora [14], CiteSeer [5], PubMed [17] and Large Cora. The statistics of these datasets are summarized in Table 2. More specifically, for each citation network, we test several scenarios, each of which the number of labeled samples per class varies from 1  $\sim$  5. In particular, we also test

Table 3: Document classification accuracy on citation networks (%).

Label Rate Dataset	1 label per class				2 labels per class				5 labels per class			
	Cora	CiteSeer	PubMed	Large Cora	Cora	CiteSeer	PubMed	Large Cora	Cora	CiteSeer	PubMed	Large Cora
LP	43.6 <sub>(0.1)</sub>	30.6 <sub>(0.1)</sub>	49.8 <sub>(0.2)</sub>	24.3 <sub>(0.3)</sub>	53.1 <sub>(0.1)</sub>	33.0 <sub>(0.1)</sub>	56.1 <sub>(0.2)</sub>	37.2 <sub>(0.3)</sub>	60.6 <sub>(0.1)</sub>	41.5 <sub>(0.1)</sub>	64.5 <sub>(0.2)</sub>	42.1 <sub>(0.3)</sub>
GCN	39.5 <sub>(0.6)</sub>	34.1 <sub>(0.9)</sub>	50.8 <sub>(4.8)</sub>	28.1 <sub>(3.8)</sub>	51.7 <sub>(0.7)</sub>	45.5 <sub>(1.0)</sub>	59.9 <sub>(5.0)</sub>	39.6 <sub>(3.9)</sub>	68.7 <sub>(0.6)</sub>	57.0 <sub>(0.9)</sub>	69.6 <sub>(4.8)</sub>	51.8 <sub>(3.8)</sub>
ST-CT	54.7 <sub>(5.3)</sub>	48.5 <sub>(8.4)</sub>	59.3 <sub>(51)</sub>	31.8 <sub>(36)</sub>	62.7 <sub>(5.5)</sub>	51.3 <sub>(8.4)</sub>	67.3 <sub>(51)</sub>	41.6 <sub>(35)</sub>	73.1 <sub>(5.6)</sub>	63.5 <sub>(8.8)</sub>	71.0 <sub>(53)</sub>	53.4 <sub>(36)</sub>
IGCN <sub>(RNM)</sub>	41.5 <sub>(0.6)</sub>	33.1 <sub>(1.0)</sub>	51.4 <sub>(4.9)</sub>	30.9 <sub>(4.2)</sub>	62.6 <sub>(0.7)</sub>	44.5 <sub>(1.0)</sub>	60.4 <sub>(5.3)</sub>	44.9 <sub>(4.5)</sub>	71.2 <sub>(0.6)</sub>	57.6 <sub>(0.9)</sub>	70.5 <sub>(4.9)</sub>	55.4 <sub>(4.2)</sub>
IGCN <sub>(AR)</sub>	42.3 <sub>(1.0)</sub>	33.0 <sub>(1.3)</sub>	52.1 <sub>(5.7)</sub>	31.6 <sub>(8.8)</sub>	62.7 <sub>(1.7)</sub>	44.9 <sub>(1.9)</sub>	61.6 <sub>(8.1)</sub>	45.3 <sub>(9.4)</sub>	72.1 <sub>(1.0)</sub>	58.1 <sub>(1.2)</sub>	71.1 <sub>(5.7)</sub>	55.7 <sub>(8.8)</sub>
GLP <sub>(RNM)</sub>	38.4 <sub>(0.4)</sub>	37.0 <sub>(0.7)</sub>	54.7 <sub>(0.8)</sub>	30.2 <sub>(2.2)</sub>	59.6 <sub>(0.4)</sub>	46.0 <sub>(0.6)</sub>	60.6 <sub>(0.6)</sub>	45.2 <sub>(2.0)</sub>	72.2 <sub>(0.4)</sub>	59.2 <sub>(0.7)</sub>	69.9 <sub>(0.8)</sub>	55.4 <sub>(1.5)</sub>
GLP <sub>(AR)</sub>	37.7 <sub>(4.0)</sub>	37.4 <sub>(19)</sub>	55.8 <sub>(9.1)</sub>	27.8 <sub>(26)</sub>	57.7 <sub>(3.4)</sub>	46.1 <sub>(16)</sub>	61.7 <sub>(7.6)</sub>	44.8 <sub>(26)</sub>	71.1 <sub>(3.9)</sub>	59.4 <sub>(19)</sub>	71.2 <sub>(9.0)</sub>	55.7 <sub>(13)</sub>
GCN	60.2 <sub>(0.9)</sub>	52.2 <sub>(1.3)</sub>	60.3 <sub>(6.1)</sub>	48.0 <sub>(4.0)</sub>	68.3 <sub>(0.9)</sub>	57.7 <sub>(1.3)</sub>	63.5 <sub>(5.7)</sub>	52.8 <sub>(4.1)</sub>	73.0 <sub>(1.2)</sub>	64.2 <sub>(1.5)</sub>	68.6 <sub>(6.3)</sub>	58.9 <sub>(4.5)</sub>
IGCN <sub>(RNM)</sub>	69.1 <sub>(1.0)</sub>	<b>57.9</b> <sub>(1.4)</sub>	63.3 <sub>(6.2)</sub>	<b>54.6</b> <sub>(4.4)</sub>	73.0 <sub>(1.0)</sub>	<b>61.7</b> <sub>(1.4)</sub>	64.9 <sub>(6.2)</sub>	57.3 <sub>(4.5)</sub>	76.4 <sub>(1.3)</sub>	<b>65.8</b> <sub>(1.6)</sub>	69.0 <sub>(7.1)</sub>	61.4 <sub>(5.1)</sub>
IGCN <sub>(AR)</sub>	<b>70.1</b> <sub>(2.4)</sub>	<b>58.3</b> <sub>(2.7)</sub>	64.7 <sub>(11)</sub>	<b>56.0</b> <sub>(8.3)</sub>	<b>73.3</b> <sub>(2.4)</sub>	<b>61.9</b> <sub>(2.7)</sub>	66.4 <sub>(11)</sub>	<b>58.1</b> <sub>(8.5)</sub>	<b>76.5</b> <sub>(3.0)</sub>	<b>65.9</b> <sub>(3.4)</sub>	70.0 <sub>(13)</sub>	61.6 <sub>(9.5)</sub>
GLP <sub>(RNM)</sub>	69.3 <sub>(0.6)</sub>	57.6 <sub>(0.8)</sub>	63.3 <sub>(0.8)</sub>	54.2 <sub>(2.2)</sub>	72.8 <sub>(0.6)</sub>	61.3 <sub>(0.8)</sub>	65.0 <sub>(0.8)</sub>	56.4 <sub>(2.7)</sub>	75.7 <sub>(0.8)</sub>	65.0 <sub>(1.1)</sub>	67.9 <sub>(1.3)</sub>	59.9 <sub>(3.3)</sub>
GLP <sub>(AR)</sub>	<b>69.8</b> <sub>(3.7)</sub>	<b>58.1</b> <sub>(17)</sub>	<b>65.2</b> <sub>(7.7)</sub>	<b>55.5</b> <sub>(26)</sub>	<b>73.5</b> <sub>(3.7)</sub>	<b>61.7</b> <sub>(17)</sub>	66.2 <sub>(7.6)</sub>	<b>57.7</b> <sub>(26)</sub>	76.3 <sub>(4.9)</sub>	<b>65.4</b> <sub>(21)</sub>	69.7 <sub>(11)</sub>	61.5 <sub>(32)</sub>
GCN	60.7 <sub>(1.3)</sub>	51.0 <sub>(1.5)</sub>	62.1 <sub>(6.1)</sub>	46.5 <sub>(4.7)</sub>	67.4 <sub>(1.2)</sub>	55.5 <sub>(1.5)</sub>	64.6 <sub>(6.1)</sub>	53.9 <sub>(4.7)</sub>	74.2 <sub>(1.3)</sub>	62.2 <sub>(1.5)</sub>	71.4 <sub>(6.0)</sub>	<b>62.0</b> <sub>(4.7)</sub>
IGCN <sub>(RNM)</sub>	69.6 <sub>(1.4)</sub>	54.5 <sub>(1.7)</sub>	64.4 <sub>(6.7)</sub>	53.3 <sub>(5.2)</sub>	73.1 <sub>(1.4)</sub>	58.6 <sub>(1.7)</sub>	<b>67.1</b> <sub>(6.7)</sub>	<b>57.7</b> <sub>(5.1)</sub>	76.4 <sub>(1.4)</sub>	63.8 <sub>(1.7)</sub>	71.7 <sub>(6.8)</sub>	<b>62.0</b> <sub>(5.1)</sub>
IGCN <sub>(AR)</sub>	<b>70.1</b> <sub>(2.8)</sub>	54.9 <sub>(3.2)</sub>	<b>66.4</b> <sub>(12)</sub>	53.2 <sub>(9.2)</sub>	<b>73.4</b> <sub>(2.8)</sub>	59.3 <sub>(3.2)</sub>	<b>67.3</b> <sub>(12)</sub>	57.6 <sub>(9.0)</sub>	<b>76.7</b> <sub>(2.8)</sub>	64.3 <sub>(3.2)</sub>	<b>73.1</b> <sub>(12)</sub>	61.7 <sub>(9.1)</sub>
GLP <sub>(RNM)</sub>	68.1 <sub>(0.9)</sub>	52.3 <sub>(1.1)</sub>	64.1 <sub>(1.2)</sub>	49.7 <sub>(2.8)</sub>	72.3 <sub>(0.9)</sub>	57.3 <sub>(1.1)</sub>	65.5 <sub>(1.2)</sub>	56.4 <sub>(2.8)</sub>	75.8 <sub>(0.9)</sub>	62.5 <sub>(1.1)</sub>	<b>72.1</b> <sub>(1.1)</sub>	61.3 <sub>(2.8)</sub>
GLP <sub>(AR)</sub>	68.0 <sub>(4.2)</sub>	53.5 <sub>(17)</sub>	<b>65.5</b> <sub>(8.3)</sub>	49.1 <sub>(27)</sub>	72.9 <sub>(4.1)</sub>	57.9 <sub>(17)</sub>	<b>66.5</b> <sub>(8.2)</sub>	56.9 <sub>(26)</sub>	<b>76.7</b> <sub>(4.1)</sub>	63.3 <sub>(17)</sub>	<b>74.0</b> <sub>(8.2)</sub>	<b>63.1</b> <sub>(26)</sub>

Shoestring-COS

Shoestring-L2

our framework under 20 labeled samples per class to evaluate the performance of *Shoestring* with sufficient labeled samples.

*Baselines.* As *Shoestring* aims for boosting the learning performance of graph-based semi-supervised learning methods, we implemented several existing models as the base models of *Shoestring* and compared with their original implementations. These methods are GCN [9], IGCN<sub>(RNM)</sub>, IGCN<sub>(AR)</sub>, GLP<sub>(RNM)</sub>, and GLP<sub>(AR)</sub>. In addition, we also compared with the methods that training GCN with self-training and co-training [11] (For simplicity, we call this set of methods *ST-CT*). Experimental results are averaged over 20 runs to ensure statistical significance.

It is worth mentioning that IGCN and GLP [12] are the state-of-the-art methods for semi-supervised learning under limited labeled samples. They are variations of GCN and LP [21] from the perspective of graph filtering. More specifically, IGCN<sub>(RNM)</sub> and IGCN<sub>(AR)</sub> change the renormalization of the adjacency matrix of the original GCN to Auto-Regressive filter (AR) and renormalization filter (RNM), respectively. GLP<sub>(RNM)</sub> and GLP<sub>(AR)</sub> propagate node features through the graph instead of propagating labels in LP. The input node features are filtered using Auto-Regressive filter (AR) or renormalization filter (RNM) for GLP<sub>(AR)</sub> and GLP<sub>(RNM)</sub> respectively. A classifier is trained on propagated features to generate the labels.

For ST-CT [11], there are four different proposals, including co-training, self-training, union, and intersection to train GCN so as to improve the learning performance. More specifically, co-training is a GCN with a random walk

model that can add the nearest neighbors of the labeled nodes to expand the labeled set iteratively. Self-training is an iterative process, where a classifier assigns the labels for the unlabeled samples which have been classified with confidence in the previous step. Union expands the training set with both random walk and GCN. Intersection, similar to union, also uses two methods but only uses the predictions that are in common. Due to the space limitation, we reported the best accuracy among these four methods.

*Similarity metrics.* In our similarity network, we used three types of similarity metrics: distance-based similarity according to L1 and L2 respectively (negative distance value as the similarity), and cosine similarity. More specifically, L1 calculates the distance between two nodes by adding the absolute differences of their feature embeddings, while L2 adds the squares differences of the feature embeddings. Cosine similarity (COS), on the other hand, is a similarity measurement between two non-zero vectors of an inner product space.

All the experiments were performed on a machine with Intel Core i7-9700K 8-core 3.6GHz CPU, 32GB RAM, 500GB SSD, and GeForce GTX 1660 Ti GPU.

*Parameter settings.* For LP, GCN, IGCN, and GLP, we use the same setting as in [12]: 0.01 learning rate, 0.5 dropout rate,  $5 \times 10^{-4}$  weight decay, 200 epochs, 16 hidden units for Cora, CiteSeer, PubMed and 64 hidden units for Large Cora. The weight of the metric-based cross-entropy loss is tuned amongst {0.001, 0.01, 0.05, 0.1} and is set to 0.01, 0.05, 0.001 for COS, L1 and L2 similarity metrics respectively.



*Results analysis.* The results for 1, 2, and 5 labeled samples are reported in Table 3. We highlighted the top-3 classification accuracies in bold. Due to the space limitation, we reported the results for 3 and 4 labeled samples and all of the results using L1 distance metric in the Appendix. A first conclusion that we can draw from these experiments is that no similarity metric is uniformly better than the others. We can also observe that  $IGCN_{(AR)}$  and  $GLP_{(AR)}$  under *Shoestring* with cosine similarity perform the best in overall cases. In particular, for 5 labeled samples per class, there is a  $\sim 5\%$  improvement with our proposed framework as compared to the original implementation. As the label rates get smaller, the improvement increases significantly, up to 32.1% performance gain on Cora for  $GLP_{(AR)}$  with 1 labeled sample per class, which shows the label-efficiency of our methods.

To further investigate the performance of our model on datasets with a larger portion of labeled samples, we test the scenario with 20 labeled samples per class. The results are shown in Table 4 with the best accuracy highlighted in bold. One of the interesting results of our framework is when the labeled samples are sufficiently large enough. Indeed, it has been shown that semi-supervised learning methods under our *Shoestring* can be very useful and the results from Table 4 exhibit better classification performance over the baseline methods. There could be a possible explanation on this fact that these semi-supervised learning models have already been effective and reliable to generate smooth and representative features for subsequent classification, when the number of labeled samples is significantly large. Augmented with a metric learning network, which is designed on the basis of the manifold assumption and cluster assumption in the embedding space, it can achieve a further performance gain, up to 1.7% on Large Cora.

The reason for high performance even with severely limited labeled samples is that, *Shoestring* can locate the centroid for each class and generate labels based on the cluster assumption and manifold assumption, which enables transferring as much knowledge as possible from sparsely labeled samples to a large number of unlabeled samples in the embedding space. To clearly visualize the improvement, Fig. 3 shows the raw features of Cora, its feature embeddings learned with one labeled sample per class based on the original GCN [9], and the feature embeddings learned based on *Shoestring*-COS and *Shoestring*-L2, respectively. The results show that GCN performs poorly with one labeled sample per class, while our proposed framework can cluster more compactly, as shown in Fig. 3c and Fig. 3d. The feature embeddings learned with 5 labels and 20 labels with *Shoestring*-COS are also shown in Fig. 3e and Fig. 3f, respectively.

**Computation cost.** The time needed for computing centroid and relative distance for similarity measurement in

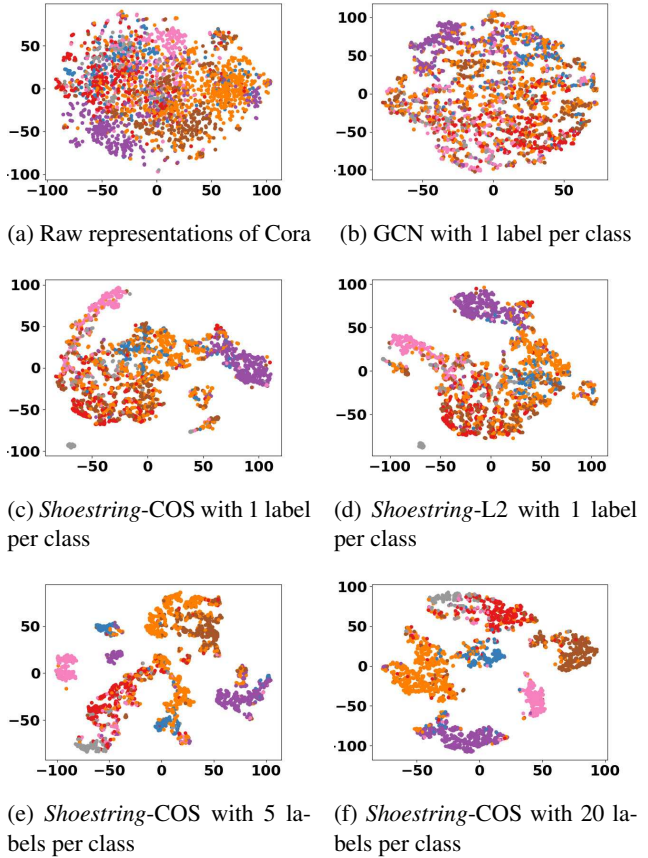


Figure 3: Visualization of Cora features.

each iteration is corresponding with the number of classes. As the benchmarking datasets do not have a significant amount of classes, the time efficiency of *Shoestring* is comparable with the original implementations. As reported in Table 3, the numbers in brackets are the computation time of each model to perform classification. For 1-labeled sample per class, there is only a 0.5 second increase in time on average with a 20% performance gains on average.

## 5.2. Performance Evaluation on Few-Shot Image Classification

Our proposed framework can also be used for few-shot image classification. Few-shot learning [3] is to learn a classifier that generalizes well even when trained with a limited number of training instances per class. An episodic meta-learning strategy [20], due to its generalization performance, has been adopted by many works on few-shot learning. To achieve larger improvements with limited numbers of training instances, several previous works proposed to consider the relationships between instances and representing the data into a graph [4, 13]. In particular, TPN [13] pro-

Table 4: Document classification accuracy on citation networks with 20 labeled samples (%).

Label Rate Dataset	20 labels per class			
	Cora	CiteSeer	PubMed	Large Cora
LP	67.8 <sub>(0.1)</sub>	47.7 <sub>(0.1)</sub>	73.3 <sub>(0.2)</sub>	52.5 <sub>(0.3)</sub>
GCN	79.8 <sub>(0.7)</sub>	68.1 <sub>(0.9)</sub>	78.0 <sub>(5.4)</sub>	67.4 <sub>(3.8)</sub>
ST-CT	80.1 <sub>(5.8)</sub>	<b>70.1</b> <sub>(9.1)</sub>	77.6 <sub>(5.4)</sub>	66.0 <sub>(18)</sub>
IGCN <sub>(RNM)</sub>	80.9 <sub>(0.7)</sub>	68.4 <sub>(1.0)</sub>	77.6 <sub>(5.5)</sub>	68.5 <sub>(3.9)</sub>
IGCN <sub>(AR)</sub>	81.3 <sub>(1.4)</sub>	68.6 <sub>(1.7)</sub>	78.5 <sub>(8.5)</sub>	68.8 <sub>(6.4)</sub>
GLP <sub>(RNM)</sub>	80.7 <sub>(0.3)</sub>	67.7 <sub>(0.4)</sub>	77.7 <sub>(0.4)</sub>	68.1 <sub>(1.5)</sub>
GLP <sub>(AR)</sub>	81.2 <sub>(1.8)</sub>	68.4 <sub>(8.6)</sub>	78.8 <sub>(3.9)</sub>	68.7 <sub>(13)</sub>
Ours	<b>81.9</b> <sub>(2.1)</sub>	69.5 <sub>(2.4)</sub>	<b>79.7</b> <sub>(4.4)</sub>	<b>70.5</b> <sub>(4.7)</sub>

posed to propagate labels between data instances for unseen classes via episodic meta-learning. Here, we replace the label propagation module with *Shoestring* in each episode training of TPN and test its performance on the few-shot image classification task.

*Datasets.* For fair comparisons with previous works, we use two datasets, miniImageNet and tieredImageNet, and follow the data preprocessing and split from [13]. The miniImageNet dataset is a subset of ImageNet dataset and designed for few-shot classification. It has 100 classes with 64 classes for training, 16 for validation, 20 for test, and 600 examples per class. Similar to miniImageNet, tieredImageNet is also a subset of ImageNet dataset, which has 608 classes and the average number of examples for each class is 1, 281. It has a hierarchical structure with 34 categories which are separated to 20 for training, 6 for validation, and 8 for test.

*Baselines.* Except for TPN [13], we also compared with the state-of-the-art method, MetaOptNet [10]. This model adapted the meta-learning framework with different convex base learners for few-shot learning. In particular, the framework was incorporated with ridge regression and support vector machines, called MetaOptNet-RR and MetaOptNet-SVM, respectively. For fair comparisons, both TPN and MetaOptNet used the standard 4-layer convolutional network with 64 filters per layer as their feature embedding architecture.

*Parameter settings.* Our implementation followed the parameter settings in [13], where the hyper-parameter  $k$  of the  $k$ -nearest neighbor graph is set to 20, label propagation parameter  $\alpha$  is set to 0.99, the query number is 15, and the results are averaged over 600 randomly generated episodes from the test set. In addition, the learning rate is set to  $10^{-3}$  initially and then is halved every 10,000 episodes for miniImageNet and 25,000 episodes for tieredImageNet, respectively. The tests are conducted under the semi-supervised condition with 5-way 1-shot and 5-way 5-

Table 5: Classification accuracy (%) on few-shot image classification on miniImageNet and tieredImageNet (5-way).

Model	miniImageNet		tieredImageNet	
	1-shot	5-shot	1-shot	5-shot
TPN	52.78	66.42	55.74	71.01
MetaOptNet-RR	53.23	69.51	54.63	72.11
MetaOptNet-SVM	52.87	68.76	54.71	71.79
<i>Shoestring</i> -TPN <sub>(COS)</sub>	<b>55.35</b>	<b>70.01</b>	<b>56.79</b>	<b>73.71</b>

shot for both datasets.

*Results analysis.* The results are shown in Table 5 with the top accuracy of each category highlighted in bold. The results of benchmarking datasets are directly obtained from their papers. From experiments, we observe that the cosine similarity is best suited for image classification and, therefore, we only include results from this method. *Shoestring*-TPN<sub>(COS)</sub> outperformed all baseline methods. In particular, *Shoestring*-TPN<sub>(COS)</sub> achieved significant gains on miniImageNet (2.57% ~ 3.59%) and tieredImageNet (1.05% ~ 2.70%), respectively. In addition, TPN under *Shoestring* leads to state-of-the-art performance as compared to MetaOptNet, demonstrating the effectiveness of *Shoestring* on few-shot image classification tasks. We can observe that the improvement for 1-shot learning is even higher than that of 5-shot, 1.765% and 1.05% respectively, showing that *Shoestring* can provide with more superior performance in severely limited labeled samples.

## 6. Concluding Remarks

In this paper, we advanced the graph-based semi-supervised learning paradigm towards a scenario where labeled data are severely limited. We proposed a new framework, called *Shoestring*, which is designed on the basis of the manifold assumption and cluster assumption in the embedding space. The experiments for both document classification on citation networks and few-shot learning image classification show strong benefits of using *Shoestring*, resulting in new state-of-the-art results across overall cases. The key factor that determines the performance of our proposed framework is that, with the metric learning network, *Shoestring* can transfer the semantic knowledge of a limited number of labeled samples to a large number of unlabeled samples. Therefore, even with just a few labeled samples, *Shoestring* can outperform all of the baseline methods. We empirically show the choice of similarity metrics in our framework is critical. One strategy to fit different datasets with different similarity metrics is to learn an adaptive similarity function. We leave this as our future work.



## References

- [1] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 11 Label Propagation and Quadratic Criterion. 2006.
- [2] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-Supervised Learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.
- [3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proc. International Conference on Machine Learning*, 2017.
- [4] Victor Garcia and Joan Bruna. Few-Shot Learning with Graph Neural Networks. In *Proc. International Conference on Learning Representation*, 2018.
- [5] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. CiteSeer: An Automatic Citation Indexing System. In *ACM DL*, 1998.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.
- [7] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label Propagation for Deep Semi-Supervised Learning. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2019.
- [8] Bo Jiang, Ziyang Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-Supervised Learning With Graph Learning-Convolutional Networks. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2019.
- [9] Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. International Conference on Learning Representation*, 2017.
- [10] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-Learning with Differentiable Convex Optimization. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2019.
- [11] 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.
- [12] Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan. Label Efficient Semi-Supervised Learning via Graph Filtering. In *Conference on Computer Vision and Pattern Recognition*, 2019.
- [13] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to Propagate Labels: Transductive Propagation Network for Few-Shot Learning. In *Proc. International Conference on Learning Representation*, 2019.
- [14] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval*, 3(2):127–163, 2000.
- [15] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *Proc. International Conference on Learning Representation*, 2018.
- [16] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-Learning for Semi-Supervised Few-Shot Classification. In *Proc. International Conference on Learning Representation*, 2018.
- [17] 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.
- [18] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. International Conference on Learning Representation*, 2015.
- [19] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical Networks for Few-Shot learning. In *Advances in Neural Information Processing Systems*, 2017.
- [20] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, 2016.
- [21] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with Local and Global Consistency. In *Advances in Neural Information Processing Systems*, 2004.
- [22] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-Supervised Learning using Gaussian Fields and Harmonic Functions. In *Proc. International Conference on Machine Learning*, 2003.