# Learn to Propagate Reliably on Noisy Affinity Graphs

Lei Yang[1 [0000−0002−0571−5924]], Qingqiu Huang[1 [0000−0002−6467−1634]], Huaiyi Huang[1 [0000−0003−1548−2498]], Linning Xu[2 [0000−0003−1026−2410]], and Dahua Lin[1 [0000−0002−8865−7896]]

[1] The Chinese University of Hong Kong
[2] The Chinese University of Hong Kong, Shenzhen
{yl016,hq016,hh016,dhlin}@ie.cuhk.edu.hk,linningxu@link.cuhk.edu.cn

**Abstract.** Recent works have shown that exploiting unlabeled data through label propagation can substantially reduce the labeling cost, which has been a critical issue in developing visual recognition models. Yet, how to propagate labels reliably, especially on a dataset with unknown outliers, remains an open question. Conventional methods such as linear diffusion lack the capability of handling complex graph structures and may perform poorly when the seeds are sparse. Latest methods based on graph neural networks would face difficulties on performance drop as they scale out to noisy graphs. To overcome these difficulties, we propose a new framework that allows labels to be propagated reliably on large-scale real-world data. This framework incorporates (1) a local graph neural network to predict accurately on varying local structures while maintaining high scalability, and (2) a confidence-based path scheduler that identifies outliers and moves forward the propagation frontier in a prudent way. Both components are learnable and closely coupled. Experiments on both ImageNet and Ms-Celeb-1M show that our confidence guided framework can significantly improve the overall accuracies of the propagated labels, especially when the graph is very noisy.

## 1 Introduction

The remarkable advances in visual recognition are built on top of large-scale annotated training data [6, 34, 33, 11, 12, 46, 7, 41, 13, 17, 15, 42, 29, 41, 16, 28, 40]. However, the ever increasing demand on annotated data has resulted in prohibitive annotation cost. Transductive learning, which aims to propagate labeled information to unlabeled samples, is a promising way to tackle this issue. Recent studies [50, 26, 21, 38, 25, 18] show that transductive methods with an appropriate design can infer unknown labels accurately while dramatically reducing the annotation efforts.

Many transductive methods adopt graph-based propagation [49, 26, 21, 38] as a core component. Generally, these methods construct a graph among all samples, propagating labels or other relevant information from labeled samples to unlabeled ones. Early methods [49, 47, 1] often resort to a linear diffusion

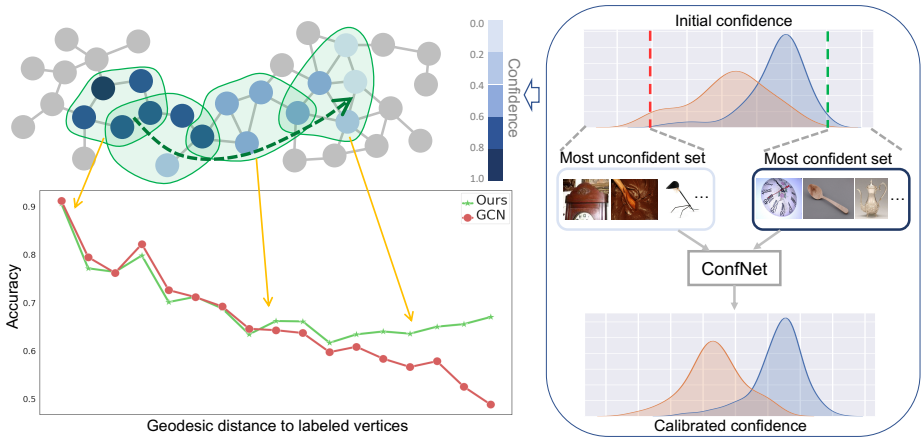


**Fig. 1:** In this paper, we propose a framework for transductive learning on noisy graphs, which contain a large number of outliers, *e.g.* out-of-class samples. The framework consists of a local predictor and a confidence-based path scheduler. The predictor updates local patches sequentially following a path driven by the estimated confidences. The path scheduler leverages both the confident and unconfident samples from the predictor to further calibrate the estimated confidence. The unconfident samples are usually images with low quality(*e.g.* the leftmost image is a clock with only top part), hard examples(*e.g.* the middle image is a spoon mixed with the background) or *out-of-class* samples(*e.g.* the rightmost image is a lamp but none of the labeled samples belong to this class). The lower left figure experimentally shows that the proposed method improves the reliability of propagation. When the distance from unlabeled samples to labeled ones increases, our method surpasses state-of-the-art by a significant margin

paradigm, where the class probabilities for each unlabeled sample are predicted as a linear combination of those for its neighbors. Relying on simplistic assumptions restricts their capability of dealing with complicated graph structures in real-world datasets. Recently, graph convolutional networks [21, 38, 39] have revealed its strong expressive power to process complex graph structures. Despite obtaining encouraging results, these GCN-based methods remain limited in an important aspect, namely the capability of coping with outliers in the graph. In real-world applications, unlabeled samples do not necessarily share the same classes with the labeled ones, leading to a large portion of *out-of-class* samples, which becomes the main source of outliers. Existing methods ignore the fact that the confidences of predictions on different samples can vary significantly, which may adversely influence the reliability of the predictions.

In this paper, we aim to explore a new framework that can propagate labels over noisy unlabeled data *reliably*. This framework is designed based on three principles: 1) *Local update:* each updating step can be carried out within a local part of the graph, such that the algorithm can be easily scaled out to a large-scale graph with millions of vertices. 2) *Learnable:* the graph structures over a real-world dataset are complex, and thus it is difficult to prescribe a rule that works

well for all cases, especially for various unknown outliers. Hence, it is desirable to have a core operator with strong expressive power that can be learned from real data. 3) *Reliable path:* graph-based propagation is sensitive to noises – a noisy prediction can mislead other predictions downstream. To propagate reliably, it is crucial to choose a path such that most inferences are based on reliable sources.

Specifically, we propose a framework comprised of two learnable components, namely, a local predictor and a path scheduler. The local predictor is a light-weight graph neural network operating on local sub-graphs, which we refer to as *graph patches*, to predict the labels of unknown vertices. The path scheduler is driven by confidence estimates, ensuring that labels are gradually propagated from highly confident parts to the rest. The key challenge in designing the path scheduler is how to estimate the confidences effectively. We tackle this problem via a two-stage design. First, we adopt a *multi-view* strategy by exploiting the fact that a vertex is usually covered by multiple *graph patches*, where each patch may project a different prediction on it. Then the confidence can be evaluated on how consistent and certain the predictions are. Second, with the estimated confidence, we construct a candidate set by selecting the most confident samples and the most unconfident ones. As illustrated in Fig. 1, we devise a *ConfNet* to learn from the candidate set and calibrate the confidence estimated from the first stage. Highly confident samples are assumed to be labeled and used in later propagation, while highly unconfident samples are assumed to be outliers and excluded in later propagation. Both components work closely together to drive the propagation process. On one hand, the local predictor follows the scheduled path to update predictions; on the other hand, the path scheduler estimates confidences based on local predictions. Note that the training algorithm also follows the same coupled procedure, where the parameters of the local predictor and confidence estimator are learned end-to-end.

Our main contributions lie in three aspects: (1) A learnable framework that involves a local predictor and a path scheduler to drive propagation reliably on noisy large-scale graphs. (2) A novel scheme of exploiting both confident and unconfident samples for confidence estimation. (3) Experiments on ImageNet [6] and Ms-Celeb-1M [9] show that our proposed approach outperforms previous algorithms, especially when the graphs are noisy and the initial seeds are sparse.

## 2 Related Work

In this paper, we focus on graph-based transductive learning [49, 26, 21, 38, 14], which constructs a graph among all samples and propagates information from labeled samples to unlabeled ones. We summarize existing methods into three categories and briefly introduce other relevant techniques.

**Early Methods.** Conventional graph-based transductive learning [49, 47, 1] is mainly originated from smoothness assumption, which is formulated as a graph Laplacian regularization. They share the same paradigm to aggregate neighbors' information through linear combination. While relying on the simple assumption,

these methods are limited by their capability of coping with complex graph structures in large-scale real-world datasets.

**GCN-based Methods.** Graph Convolutional Network (GCN) and its variants [21, 38, 32, 39] apply filters over the entire graph and achieve impressive performance on several tasks [48]. To extend the power of GCN to large-scale graphs, GraphSAGE [10] proposes to sample a fixed number of neighbors and apply aggregation functions thereon. FastGCN [4] further reduces memory demand by sampling vertices rather than neighbors in each graph convolution layer. However, they propagate labels in parallel across all vertices, regardless of the confidence difference among predictions. This ignorance on prediction confidence may adversely influence the reliability of propagation.

**Confidence-based Methods.** Previous approaches either model the node label as a distribution along with uncertainty [31, 2] or re-scale the weight of links by introducing attention to vertices [38, 37]. Unlike these methods, which mainly focus on making use of confident samples, our approach learns from both confident and unconfident data for confidence estimation.

**Inductive Learning.** Inductive learning is closely related to transductive learning [36]. The key difference lies in that the former aims to learn a better model with unlabeled data [35, 25, 45, 44, 43], while the latter focuses on predicting labels for unlabeled data [49, 50]. One important line of inductive learning is leveraging the predicted labels of unlabeled data in a supervised manner [23, 19]. In this paper, we focus on transductive learning and show that the obtained pseudo labels can be applied to inductive learning as a downstream task.

**Outlier Detection.** Previous methods [3] either rely on crafted unsupervised rules [24] or employing a supervised method to learn from an extra labeled outlier dataset [5]. The unsupervised rules lack the capability of handling complex real-world dataset, while the supervised methods are easy to overfit to the labeled outliers and do not generalize well. In this paper, we *learn* to identify outliers from carefully selected confident and unconfident samples during propagation.

## 3 Propagation on Noisy Affinity Graphs

Our goal is to develop an effective method to propagate reliably over noisy affinity graphs, *e.g.* those containing lots of out-of-class samples, while maintaining reasonable runtime cost. This is challenging especially when the proportion of outliers are high and initial seeds are sparse. As real-world graphs often have complex and varying structures, noisy predictions can adversely affect these downstream along the propagation paths. We propose a novel framework for graph-based propagation, which copes with the complexity in local graph structures via a light-weight graph convolutional network while improving the reliability via a confidence-based scheduler that chooses propagation paths prudently.

### 3.1 Problem Statement

Consider a dataset with $N = N_l + N_u$ samples, where $N_l$ samples are labeled and $N_u$ are unlabeled, and $N_l \ll N_u$. We denote the set of labeled samples as

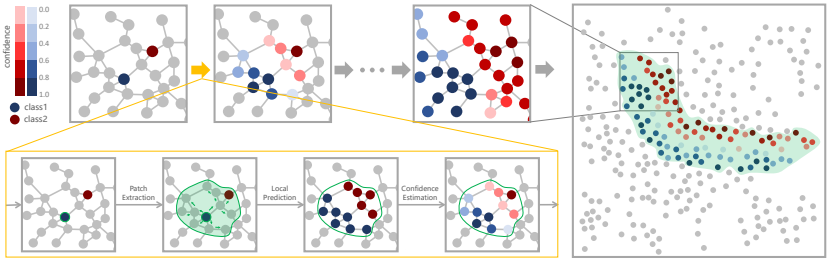


**Fig. 2:** Overview of our framework (better viewed in color). At each iteration, our approach consists of three steps: (1) Starting from the selected confident vertex, the patch extractor generates a graph patch. (2) Given the graph patch, the learned local predictor updates the predictions of all unlabeled vertices on the patch. (3) Given the updated predictions, the path scheduler estimates confidence for all unlabeled vertices. Over many iterations, labeled information are gradually propagated from highly confident parts to the rest

$\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_l}$, and that of unlabeled ones as $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=N_l+1}^{N_l+N_u}$. Here, $\mathbf{x}_i \in \mathbb{R}^d$ is the feature for the $i$-th sample, which is often derived from a deep network in vision tasks, and $y_i \in \mathcal{Y}$ is its label, where $\mathcal{Y} = \{1, ..., m\}$. In our setting, $\mathcal{D}_u$ consists of two parts, namely in-class samples and out-of-class samples. For out-of-class data, their labels do not belong to $\mathcal{Y}$. The labeled set $\mathcal{D}_l$ only contains in-class labeled samples. The goal is to assign a label $\hat{y} \in \mathcal{Y} \bigcup \{-1\}$ to each unlabeled sample in $\mathcal{D}_u$, where $\hat{y} = -1$ indicates an unlabeled sample is identified as an outlier.

To construct an affinity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ on this dataset, we treat each sample as a vertex and connect it with its $K$ nearest neighbors. The graph $\mathcal{G}$ can be expressed by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where $a_{i,j} = \mathbf{A}(i, j)$ is the cosine similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ if $(i, j) \in \mathcal{E}$, and otherwise 0.

For label propagation, we associate each vertex with a probability vector $\mathbf{p}_i$, where $p_{ik} = \mathbf{p}_i(k)$ indicates the probability of the sample $\mathbf{x}_i$ belonging to the $k$-th class, and a confidence score $c_i \in [0, 1]$. For labeled samples, $\mathbf{p}_i$ is fixed to be a one-hot vector with $p_{ik} = 1$ for $k = y_i$. For unlabeled samples, $\mathbf{p}_i$ is initialized to be a uniform distribution over all classes and will be gradually updated as the propagation proceeds. We set a threshold $o_\tau$ to determine whether a sample is an outlier. After the propagation is completed, for those with confidence smaller than $o_\tau$, the predicted label for each unlabeled sample $\mathbf{x}_i$ is set to $\hat{y}_i = -1$. For those with confidence larger than $o_\tau$. the predicted label for each unlabeled sample $\mathbf{x}_i$ is set to be the class with highest probability in $\mathbf{p}_i$, as $\hat{y}_i = \text{argmax}_k\, p_{ik}$.

### 3.2 Algorithm Overview

As shown in Fig. 2, our proposed propagation scheme is based on graph patches as the units for updating. Here, a *graph patch* is a sub-graph containing both

labeled and unlabeled vertices. The algorithm performs updates over a graph patch in each step of propagation.

The propagation proceeds as follows. (1) At each iteration, we first randomly select a vertex from the *high-confidence vertex set* $\mathcal{S}$, which contains both the initially labeled samples and those samples whose confidences are high enough to be considered as "labeled" as the propagation proceeds. (2) Starting from the selected *confident vertex*, we use a patch extractor to expand it into a graph patch, and then update the predictions on all unlabeled vertices in this patch, using a local predictor. (3) The path scheduler uses these predictions to re-estimate confidences for unlabeled vertices. In this work, both the local predictor and the path scheduler are formulated as a graph convolutional network (GCN) learned from the training data, in order to cope with the complexity of local graph structures. All the vertices whose confidence scores go beyond a threshold $c_\tau$ will be added into $\mathcal{S}$ and their predictions will not be updated again in future iterations. Note that the updated confidences would influence the choice of the next confident vertex and thus the propagation path. By iteratively updating predictions and confidences as above, the algorithm drives the propagation across the entire graph, gradually from high confident areas to the rest.

This propagation algorithm involves two components: a *local predictor* that generates confident graph patches and updates predictions thereon, and a *path scheduler* that estimates confidences and schedules the propagation path accordingly. Next, we will elaborate on these components in turn.

### 3.3 GCN-based Local Predictor

**Patch extractor.** A graph patch with the following properties is a good candidate for the next update. (1) *High confidence:* We define the *confidence of a graph patch* as the sum of its vertex confidences. A patch with high confidence is more likely to yield reliable predictions due to the availability of reliable information sources. (2) *Large expected confidence gain*: We define the *estimated confidence gain* of a patch $\mathcal{P}_i$ as $\sum_{v_j \in \mathcal{P}_i}(1-c_j)$, *i.e.* the maximum possible improvement on the total confidence. Performing updates on those patches with large expected confidence gain can potentially speed up the propagation. To maintain sufficient confidence gain while avoiding excessive patch sizes, we consider a patch as *viable* for the next update if the expected gain is above a threshold $\Delta c_\tau$ and the size is below the maximum size $s$. Besides, to avoid selecting highly overlapped patches, once a vertex is taken as the start point, its $m$-hop neighbors will all be excluded from selecting as start points in later propagation.

To generate a graph patch $\mathcal{P}$, we start from the most confident vertex and add its immediate neighbors into a queue. Each vertex in the queue continues to search its unvisited neighbors until (1) the expected gain is above $\Delta c_\tau$, which means that a viable patch is obtained; or (2) the size exceeds $s$, which means that no viable patch is found around the selected vertex and the algorithm randomly selects a new vertex from $\mathcal{S}$ to begin with. Note that our propagation can be parallelized by selecting multiple non-overlapped patches at the same time. We show the detailed algorithm in supplementary.

Graph patches are dynamically extracted along with the propagation. In early iterations, $\Delta c_\tau$ can often be achieved by a small number of unlabeled vertices, as most vertices are unlabeled and have low confidences. This results in more conservative exploration at the early stage. As the propagation proceeds, the number of confident vertices increases while the average expected confidence gain decreases, the algorithm encourages more aggressive updates over larger patches. Empirically, we found that on an affinity graph with $10K$ vertices with 1% of labeled seeds, it takes about 100 iterations to complete the propagation procedure, where the average size of graph patches is $1K$.

**Design of local predictor.** We introduce a graph convolutional network (GCN) to predict unknown labels for each graph patch. Given a graph patch $\mathcal{P}_i$ centered at $v_i \in \mathcal{V}$, the network takes as input the visual features $\mathbf{x}_i$, and the affinity sub-matrix restricted to $\mathcal{P}_i$, denoted as $\mathbf{A}(\mathcal{P}_i)$. Let $\mathbf{F}_0(\mathcal{P}_i)$ be a matrix of all vertex data for $\mathcal{P}_i$, where each row represents a vertex feature $\mathbf{x}_i$. The GCN takes $\mathbf{F}_0(\mathcal{P}_i)$ as the input to the bottom layer and carries out the computation through $L$ blocks as follows:

$$\mathbf{F}_{l+1}(\mathcal{P}_i) = \sigma\left(\tilde{\mathbf{D}}(\mathcal{P}_i)^{-1}\tilde{\mathbf{A}}(\mathcal{P}_i)\mathbf{F}_l(\mathcal{P}_i)\mathbf{W}_l\right), \tag{1}$$

where $\tilde{\mathbf{A}}(\mathcal{P}_i) = \mathbf{A}(\mathcal{P}_i) + \mathbf{I}$; $\tilde{\mathbf{D}} = \sum_j \tilde{\mathbf{A}}_{ij}(\mathcal{P}_i)$ is a diagonal degree matrix; $\mathbf{F}_l(\mathcal{P}_i)$ contains the embeddings at the $l$-th layer; $\mathbf{W}_l$ is a matrix to transform the embeddings; $\sigma$ is a nonlinear activation ($ReLU$ in this work). Intuitively, this formula expresses a procedure of taking weighted average of the features of each vertex and its neighbors based on affinity weights, transforming them into a new space with $\mathbf{W}_l$, and then feeding them through a nonlinear activation. Note that this GCN operates locally within a graph patch and thus the demand on memory would not increase as the whole graph grows, which makes it easy to scale out to massive graphs with millions of vertices.

As the propagation proceeds, each vertex may be covered by multiple patches, including those constructed in previous steps. Each patch that covers a vertex $v$ is called a *view* of $v$. We leverage the predictions from multiple views for higher reliability, and update the probability vector for each unlabeled vertex in $\mathcal{P}_i$ by averaging the predictions from all views, as

$$\mathbf{p}_i = \frac{1}{\sum \mathbb{1}_{v_i \in \mathcal{P}_j}} \sum_{v_i \in \mathcal{P}_j} \mathbf{F}_L(v_{i,j}). \tag{2}$$

### 3.4 Confidence-based Path Scheduler

Confidence estimation is the core of the path scheduler. A good estimation of confidences is crucial for reliable propagation, as it allows unreliable sources to be suppressed. Our confidence estimator involves a *Multi-view* confidence estimator and a learnable *ConfNet*, to form a two-stage procedure. Specifically, the former generates an initial confidence estimation by aggregating predictions from multiple patches. Then ConfNet learns from the most confident samples and

the most unconfident ones from the first stage, to further refine the confidence. The ultimate confidence is the average confidence of these two stages.

**Multi-view confidence estimation.** Previous studies [8, 22] have shown that neural networks usually yield over-confident predictions. In this work, we develop a simple but effective way to alleviate the over-confidence problem. We leverage the multiple views for each vertex $v_i$ derived along the propagation process. Particularly, the confidence for $v_i$ is defined as

$$c_i = \begin{cases} \max_k p_{ik}, & \text{if } v_i \text{ was visited multiple times} \\ \epsilon, & \text{if } v_i \text{ was visited only once} \end{cases} \tag{3}$$

where $\mathbf{p}_i$ is given in Eq. (2), and $\epsilon$ is a small positive value.

Here, we discuss why we use $c_i$ as defined above to measure the confidence. When a vertex has only been visited once, it is difficult to assess the quality of the prediction, therefore it is safe to assume a low confidence. When a vertex has been visited multiple times, a high value of $\max_k p_{ik}$ suggests that the predictions from different views are consistent with each other. If not, *i.e.* different views vote for different classes, then the average probability for the best class would be significantly lower. We provide a proof in the supplementary showing that $c_i$ takes a high value only when predictions are consistent and all with low entropy.

**ConfNet.** Among the initial confidence estimated from previous stage, the most confident samples are most likely to be genuine members while the most unconfident samples are most likely to be outliers, which can be regarded as positive samples and negative samples, respectively.

ConfNet is introduced to learn from the "discovered" genuine members and outliers. It aims to output a probability value for each vertex $v$ to indicate how likely it is a genuine member instead of an outlier. Similar to the local predictor, we implement ConfNet as a graph convolutional network, following Eq. (1). Given a percentage $\eta$ and sampled graph patches, we take the top-$\eta$ confident vertices as the positive samples and the top-$\eta$ unconfident vertices as the negative ones. Then we train the ConfNet using the vertex-wise binary cross-entropy as the loss function. The final confidence of a vertex is estimated as the average of multi-view confidence and the predicted confidence from the learned ConfNet.

### 3.5 Training of Local Predictor

Here we introduce how to train the local predictor. The training samples consist of graph patches with at least one labeled vertex. Instead of selecting graph patches consecutively during propagation, we sample a set of graph patches parallel for training. The sampling of graph patches follows the same principle, *i.e.*, selecting those with high confidence. Based on the sampled subgraphs, the local predictor predicts labels for all labeled vertices on sampled subgraphs. The cross-entropy error between predictions and ground-truth is then minimized over all labeled data to optimize the local predictor.

# 4 Experiments

## 4.1 Experimental Settings

**Dataset.** We conduct our experiments on two real-world datasets, namely, ImageNet [6] and Ms-Celeb-1M [9]. ImageNet compromises $1M$ images from $1,000$ classes, which is the most widely used image classification dataset. Ms-Celeb-1M is a large-scale face recognition dataset consisting of $100K$ identities, and each identity has about 100 facial images. Transductive learning in vision tasks considers a practical setting that obtains a pretrained model but its training data are unavailable. Given only the pretrained model and another unlabeled set with limited labeled data, it aims to predict labels for the unlabeled set. We simulate this setting with the following steps: (1) We randomly sample 10% data from ImageNet to train the feature extractor $\mathcal{F}$. (2) We use $\mathcal{F}$ to extract features for the rest 90% samples to construct $\mathcal{D}_{all}$. (3) We randomly sample 10 classes from $\mathcal{D}_{all}$ as $\mathcal{D}$, and randomly split 1% data from $\mathcal{D}$ as the labeled set $\mathcal{D}_l$. (4) With a noise ratio $\rho$, we construct the outlier set $\mathcal{D}_o$ by randomly sampling data from $\mathcal{D}_{all} \setminus \mathcal{D}$. (5) $\mathcal{D}_u$ is a union set of $\mathcal{D} \setminus \mathcal{D}_l$ and $\mathcal{D}_o$. Experiments on Ms-Celeb-1M follow the same setting except sampling 100 classes. We sample a small validation set $\mathcal{D}_v$ with the same size as $\mathcal{D}_l$, to determine the outlier threshold $o_\tau$. To evaluate performance on graphs with different noise ratio, we set the noise ratio $\rho$ to 0%, 10%, 30% and 50% .

**Metrics.** We assess the performance under the noisy transductive learning. Given the ground-truth of the unlabeled set, where the ground-truth of out-of-class outliers is set to $-1$, transductive learning aims to predict the label of each sample in $\mathcal{D}_u$, where the performance is measured by *top*-1 accuracy.

**Implementation Details.** We take ResNet-50 [11] as the feature extractor in our experiments. $K = 30$ is used to build the $K$NN affinity graph. $c_\tau$ is set to 0.9 as the threshold to fix high confident vertices. $s$ and $\Delta c_\tau$ for generating graph patches is 3000 and 500. We use SGCs [39] for both local predictor and ConfNet. The depth of SGC is set to 2 and 1 for local predictor and ConfNet, respectively. The Adam optimizer is used with a start learning rate 0.01 and the training epoch is set to 200 and 100 for local predictor and ConfNet, respectively.

## 4.2 Method Comparison

We compare the proposed method with a series of transductive baselines. Since all these methods are not designed for noisy label propagation, we adapt them to this setting by adopting the same strategy as our method. Specifically, we first determine the outlier threshold $o_\tau$ on a validation set $\mathcal{D}_v$, and then take the samples whose confidence below the threshold $o_\tau$ as the noisy samples. The methods are briefly described below.

**(1) LP [49]** is the most widely used transductive learning approach, which aggregates the labels from the neighborhoods by linear combination.

**(2) GCN [21]** is devised to capture complex graph structure, where each layer consists of a non-linear transformation and an aggregation function.

**(3) GraphSAGE [10]** is originally designed for node embedding, which applies trainable aggregation functions on sampled neighbors. We adapt it for transductive learning by replacing the unsupervised loss with the cross-entropy loss.
**(4) GAT [38]** introduces a self-attention mechanism to GCN, which enables specifying different weights to different nodes in a neighborhood.
**(5) FastGCN [4]** addresses the memory issue of GCN by a sampling scheme. Compared with GraphSAGE, it saves more memory by sampling vertices rather than neighbors at each layer.
**(6) SGC [39]** simplifies the non-linear transformation of GCN, which comprises a linear local smooth filter followed by a standard linear classifier.
**(7) Ours** incorporates a local predictor and a confidence-based path scheduler. The two closely coupled components learn to propagate on noisy graphs reliably.

**Table 1:** Performance comparison of transductive methods on noisy affinity graphs. GraphSAGE$^\dagger$ denotes using GCN as the aggregation function. For both ImageNet and Ms-Celeb-1M, 1% labeled images are randomly selected as seeds. We randomly select classes and initial seeds for 5 times and report the average results of 5 runs (see supplementary for the standard deviation of all experiments)

| | ImageNet | | | | Ms-Celeb-1M | | | |
|---|---|---|---|---|---|---|---|---|
| Noise ratio $\rho$ | 0% | 10% | 30% | 50% | 0% | 10% | 30% | 50% |
| LP [49] | 77.74 | 70.51 | 59.47 | 51.43 | 95.13 | 89.01 | 88.31 | 87.19 |
| GCN [21] | 83.17 | 75.37 | 66.28 | 64.09 | 99.6 | 99.6 | 96.37 | 96.3 |
| GAT [38] | 83.93 | 75.99 | 66.3 | 63.34 | 99.59 | 96.48 | 94.55 | 94.01 |
| GraphSAGE [10] | 82.42 | 73.42 | 63.84 | 59.12 | 99.57 | 95.68 | 92.21 | 91.06 |
| GraphSAGE$^\dagger$ [10] | 81.39 | 73.53 | 63.42 | 58.99 | 99.59 | 95.62 | 92.38 | 91.19 |
| FastGCN [4] | 81.34 | 74.08 | 63.79 | 58.81 | 99.62 | 95.6 | 92.08 | 90.83 |
| SGC [39] | 84.78 | 76.71 | 67.97 | 65.63 | 99.63 | 97.43 | 96.71 | 96.5 |
| **Ours** | **85.16** | **76.96** | **69.28** | **68.25** | **99.66** | **97.59** | **96.93** | **96.81** |

**Results.** Table. 1 shows that: (1) For LP, the performance is inferior to other learning-based approaches. (2) GCN shows competitive results under different settings, although it is not designed for the noisy scenario. (3) We employ GraphSAGE with GCN aggregation and mean aggregation. Although it achieves a higher speedup than GCN, not considering the confidence of predictions makes the sampling-based method very sensitive to outliers. (4) Although GAT yields promising results when the graph size is $20K$, it incurs excessive memory demand when scaling to larger graphs, as shown in Fig. 3. Despite FastGCN is efficient, it suffers from the similar problem as GraphSAGE. (6) SGC, as a simplified version of GCN, achieves competitive results to GCN and GAT. As it has less training parameters, it may not easily overfit when the initial seeds are sparse. Fig. 3 indicates that the performance of SGC becomes inferior to GCN when the graph size becomes large. (7) Table. 1 illustrates that the noisy setting is very challenging, which deteriorates the performance of all algorithms marginally. The proposed method improves the accuracy under all noise ratios,

with more significant improvement as the noise ratio becomes larger. It not only surpasses the sampling-based approaches by a large margin, but also outperforms the GNNs with the entire graph as inputs. Even in the well-learned face manifold, which is less sensitive to out-of-sample noise, our method still reduces the error rate from 3.5% to 3.19%. Note that the proposed method can be easily extended to the iterative scheme by using self-training [30]. As it can effectively estimate confidence, applying it iteratively can potentially lead to better results.

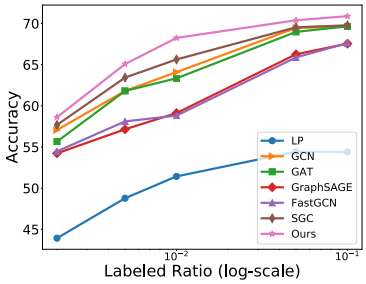


**Fig. 3:** Influence of labeled ratio

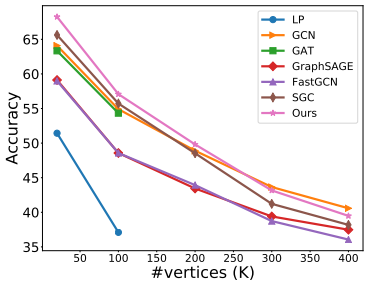


**Fig. 4:** Influence of graph size

**Labeled Ratio.** When the noise ratio $\rho$ is 50%, we study the influence of different labeled ratios: 0.2%, 0.5%, 1%, 5% and 10%. Fig. 3 shows that our method consistently outperforms other methods under all labeled ratios. When the initial seeds are very sparse, it becomes more challenging for both label propagation and confidence estimation. As our method learns from the discovered confident and unconfident samples along with the propagation, our method still performs well when there are a few initial seeds.

**Graph Scale.** The local update design makes the proposed method capable of scaling to large-scale graphs. As Fig. 4 illustrates, LP suffers a severe performance drop when the graph size increases. GAT exceeds the memory limits when the number of vertices is beyond $100K$. Two sampling-based methods, GraphSAGE and FastGCN, are inferior to their counterparts operating on the entire graph. Although our method also operates on subgraphs, the reliable strategy enables it to perform well on noisy graphs under different scales. Note that when the graph size is $400K$, GCN performs better than ours. As we adopt SGC as the local predictor in our experiments, without non-linear transformation may limit its capability when graph scale is large. In real practice, we have the flexibility to select different local predictors according to the graph scale.

### 4.3 Ablation Study

We adopt a setting on ImageNet, where the labeled ratio is 1% and the noise ratio is 50%, to study some important designs in our framework.

**Local predictor.** In our framework, the local predictor can be flexibly replaced with different graph-based algorithms. We compare the effectiveness of three

**Table 2:** Comparison on local predictors and confidences. ConfNet$^\dagger$ computes confidence as the average confidence from Multi-view and ConfNet

| Confidence | GAT | GCN | SGC |
|---|---|---|---|
| Random | 63.16 | 62.62 | 64.84 |
| Multi-view | 64.11 | 63.84 | 65.81 |
| ConfNet | 64.83 | 63.93 | 67.79 |
| ConfNet$^\dagger$ | 65.95 | 65.21 | 68.25 |
| GT | 83.17 | 83.93 | 84.78 |

**Table 3:** Comparison on different source of initial confidence. FNR denotes *false noise ratio* of positive samples and TNR denotes *true noise ratio* of negative samples

| Initial Confidence | Num | FNR | TNR | Acc |
|---|---|---|---|---|
| SGC($\eta$=0.05) | 973 | 3.8% | 66% | 67.44 |
| Multi-view($\eta$=0.01) | 194 | 1.6% | 70% | 66.78 |
| Multi-view($\eta$=0.05) | 973 | 3.2% | 65% | 68.79 |
| Multi-view($\eta$=0.1) | 1947 | 4.1% | 63% | 67.81 |
| GT($\eta$=0.05) | 973 | 0% | 100% | 76.29 |

learnable local predictors, namely GAT, GCN, and SGC. All three methods take the vertex features as input, and predict labels for unlabeled vertices. Comparing different columns in Table. 2, all three local predictors outperforms LP (see Table. 1) significantly, even using random confidence. The results demonstrate the advantage of learning-based approaches in handling complex graph structure.

**Path scheduler.** As shown in different rows in Table. 2, we study confidence choices with different local predictors. (1) *Random* refers to using random score between 0 and 1 as the confidence, which severely impairs the performance. (2) *Multi-view* denotes our first stage confidence estimation, *i.e.*, aggregating predictions from multiple graph patches, which provides a good initial confidence. (3) *ConfNet* indicates using the confidence predicted from ConfNet. Compared to Multi-view, the significant performance gain demonstrates the effectiveness of ConfNet. (4) *ConfNet*$^\dagger$ is the ultimate confidence in our approach. It further increases the performance by averaging confidence from two previous stages, which shows that the confidence from Multi-view and ConfNet may be complementary to some extent. (5) *GT (Ground-truth)* denotes knowing all outliers in advance, which corresponds to the setting that noise ratio is 0 in Table. 1. It indicates that the performance can be greatly boosted if identifying all outliers correctly.

**Confidence estimation** Table. 3 analyzes the source of initial confidence for ConfNet training. $\eta$ denotes the proportion of the most confident and unconfident samples, as defined in Sec. 3.4. SGC refers to using the prediction probabilities without Multi-view strategy. It shows that: (1) Comparison between SGC ($\eta$=0.05) and Multi-view ($\eta$=0.05) indicates that ConfNet is affected by the quality of initial confidence set. As Multi-view gives more precise confidence estimation, it provides more reliable samples for ConfNet training, leading to a better performance. (2) Comparison between GT ($\eta$=0.05) and Multi-view ($\eta$=0.05) further indicates that training on a reliable initial confidence set is a crucial design. (3) Comparison between Multi-view with three different $\eta$ shows that choosing a proper proportion is important to ConfNet training. When $\eta$ is small, although the positive and negative samples are more pure, training on a few samples impairs the final accuracy. When $\eta$ is large, the introduction of noise in both positive and negative samples lead to the limited performance gain.

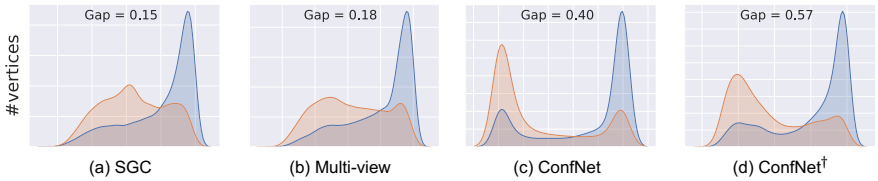


**Fig. 5:** Confidence distribution of outliers and genuine members. Orange represents the out-of-class noisy samples, while blue denotes the in-class unlabeled ones. Gap is computed as the difference between the mean of two distributions. It indicates that the proposed confidence estimation approach can enlarge the confidence gap between outliers and genuine ones, which is the key to our performance gain

From another perspective, Fig. 5 illustrates that the success of Multi-view and ConfNet is mainly due to altering the confidence distribution, where the gap between outliers and genuine members is enlarged and thus outliers can be identified more easily. Fig. 7 shows that using ConfNet as a post-processing module in previous methods can also improve their capability of identifying outliers, leading to a significant accuracy gain with limited computational budget.

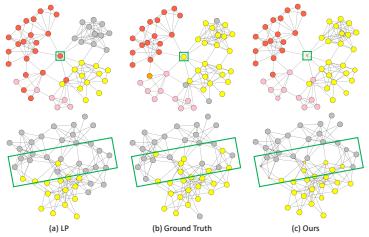


**Fig. 6:** Two graph patches with predictions from ImageNet, where different colors represent different classes

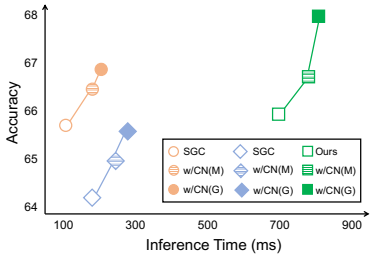


**Fig. 7:** Apply ConfNet to different GNNs. CN(M) denotes ConfNet using MLP and CN(G) denotes ConfNet using GCN

### 4.4 Further Analysis

**Efficiency of Path Extraction.** We refer to *visited times* of a vertex as the number of patches it belongs to. We conduct experiments on ImageNet with $10K$ vertices with $c_\tau = 0.9, \Delta c_\tau = 500$ and $s = 3000$. When propagating 100 iterations, the average visited times of vertices are about 6. Most samples are visited 2 times and only a very few samples are visited more than 10 times.

**Conservative Prediction on Hard Cases.** Except the out-of-sample noise, we also visualize the low confident samples when noise ratio is 0. As Fig. 6 shows, when dealing with a hard case (the green box in the first row), our method gives the right prediction with very low confidence (small size of vertices) while LP

gives a wrong prediction and misleads the predictions of downstream vertices. The second row shows that our confidence can identify inter-class boundaries, and remain conservative to these predictions, as highlighted in the green box.

**Table 4:** Two applications of our method in vision tasks. (a) We use the estimated confidence as indicators in active learning. (b) We apply the predicted labels to face recognition training in an inductive manner. (see supplementary for more details)

| Labeled | Baseline | Random | GCN | Ours |
|---|---|---|---|---|
| 1% | 65.63 | 65.71 | 66.7 | 68.6 |

(a)

| Test Protocol | Baseline | CDP | GCN | Ours |
|---|---|---|---|---|
| MegaFace [20] | 58.21 | 59.15 | 59.33 | 60.02 |

(b)

### 4.5 Applications

**Active Learning.** Active learning desires an effective indicator to select representative unlabeled samples. Table. 4(a) shows that our estimated confidence outperforms two widely used indicators. Specifically, the first one *randomly* selects unlabeled samples for annotation, while the second one applies a trained *GCN* to unlabeled samples and select those with large predicted entropy. *Baseline* refers to the accuracy before annotation. The result shows that our method brings larger accuracy gain by annotating the same number of unlabeled data.
**Inductive Learning.** The predicted labels from transductive learning can be used as "pseudo labels" in inductive learning. We randomly selects $1K$ person with $120K$ images from Ms-Celeb-1M, sampling 1% as the labeled data. We compare with CDP [45] and GCN [21] for generating "pseudo labels". Compared to these two methods, Table. 4(b) shows our method brings larger performance gain on MegaFace [20], which demonstrates that the proposed method generates pseudo labels with higher quality.

## 5 Conclusion

In this paper, we propose a reliable label propagation approach to extend the transductive learning to a practical noisy setting. The proposed method consists of two learnable components, namely a GCN-based local predictor and a confidence-based path scheduler. Experiments on two real-world datasets show that the proposed approach outperforms previous state-of-the-art methods with reasonable computational cost. Ablation study shows that exploiting both confident and unconfident samples is a crucial design in our confidence estimation. Extending the proposed method to different kinds of noise, such as adversarial noise [27], is desired to be explored in the future.
**Acknowledgment** This work is partially supported by the SenseTime Collaborative Grant on Large-scale Multi-modality Analysis (CUHK Agreement No. TS1610626 & No. TS1712093), the General Research Fund (GRF) of Hong Kong (No. 14203518 & No. 14205719).

## References


1. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of machine learning research **7**(Nov), 2399–2434 (2006)
2. Bojchevski, A., Günnemann, S.: Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint arXiv:1707.03815 (2017)
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys **41**(3), 1–58 (2009)
4. Chen, J., Ma, T., Xiao, C.: Fastgcn: fast learning with graph convolutional networks via importance sampling. arXiv preprint arXiv:1801.10247 (2018)
5. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems. pp. 7–10 (2016)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. Ieee (2009)
7. Deng, J., Guo, J., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. arXiv preprint arXiv:1801.07698 (2018)
8. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
9. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European Conference on Computer Vision. pp. 87–102. Springer (2016)
10. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. pp. 1024–1034 (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
13. Huang, H., Zhang, Y., Huang, Q., Guo, Z., Liu, Z., Lin, D.: Placepedia: Comprehensive place understanding with multi-faceted annotations. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
14. Huang, Q., Liu, W., Lin, D.: Person search in videos with one portrait through visual and temporal links. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 425–441 (2018)
15. Huang, Q., Xiong, Y., Lin, D.: Unifying identification and context learning for person recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
16. Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: Movienet: A holistic dataset for movie understanding. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
17. Huang, Q., Xiong, Y., Xiong, Y., Zhang, Y., Lin, D.: From trailers to storylines: An efficient way to learn from movies. arXiv preprint arXiv:1806.05341 (2018)
18. Huang, Q., Yang, L., Huang, H., Wu, T., Lin, D.: Caption-supervised face recognition: Training a state-of-the-art face model without manual annotation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)

19. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5070–5079 (2019)
20. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: CVPR (2016)
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
22. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. pp. 6402–6413 (2017)
23. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML. vol. 3, p. 2 (2013)
24. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 631–636 (2003)
25. Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In: Advances in Neural Information Processing Systems. pp. 3239–3250 (2018)
26. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710. ACM (2014)
27. Qiu, H., Xiao, C., Yang, L., Yan, X., Lee, H., Li, B.: Semanticadv: Generating adversarial examples via attribute-conditional image editing. arXiv preprint arXiv:1906.07927 (2019)
28. Rao, A., Wang, J., Xu, L., Jiang, Xuekun, H.Q., Zhou, B., Lin, D.: A unified framework for shot type classification based on subject centric lens. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
29. Rao, A., Xu, L., Xiong, Y., Xu, G., Huang, Q., Zhou, B., Lin, D.: A local-to-global approach to multi-modal movie scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10146–10155 (2020)
30. Rosenberg, C., Hebert, M., Schneiderman, H.: Semi-supervised self-training of object detection models. WACV/MOTION **2** (2005)
31. Saunders, C., Gammerman, A., Vovk, V.: Transduction with confidence and credibility (1999)
32. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)
33. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR (2015)
34. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: NeurIPS (2014)
35. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in neural information processing systems. pp. 1195–1204 (2017)
36. Vapnik, V.: 24 transductive inference and semi-supervised learning (2006)
37. Vashishth, S., Yadav, P., Bhandari, M., Talukdar, P.: Confidence-based graph convolutional networks for semi-supervised learning. arXiv preprint arXiv:1901.08255 (2019)

38. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
39. Wu, F., Zhang, T., Souza Jr, A.H.d., Fifty, C., Yu, T., Weinberger, K.Q.: Simplifying graph convolutional networks. arXiv preprint arXiv:1902.07153 (2019)
40. Wu, T., Huang, Q., Liu, Z., Wang, Y., Lin, D.: Distribution-balanced loss for multi-label classification in long-tailed datasets. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
41. Xia, J., Rao, A., Xu, L., Huang, Q., Wen, J., Lin, D.: Online multi-modal person search in videos. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
42. Xiong, Y., Huang, Q., Guo, L., Zhou, H., Zhou, B., Lin, D.: A graph-based framework to bridge movies and synopses. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
43. Yang, L., Chen, D., Zhan, X., Zhao, R., Loy, C.C., Lin, D.: Learning to cluster faces via confidence and connectivity estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
44. Yang, L., Zhan, X., Chen, D., Yan, J., Loy, C.C., Lin, D.: Learning to cluster faces on an affinity graph. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2298–2306 (2019)
45. Zhan, X., Liu, Z., Yan, J., Lin, D., Loy, C.C.: Consensus-driven propagation in massive unlabeled data for face recognition. In: ECCV (2018)
46. Zhang, X., Yang, L., Yan, J., Lin, D.: Accelerated training for massive classification via dynamic class selection. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
47. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in neural information processing systems. pp. 321–328 (2004)
48. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)
49. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. rep., Citeseer (2002)
50. Zhu, X.J.: Semi-supervised learning literature survey. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2005)