Neighborhood-Aware Attention Network for Semi-supervised Face Recognition

Qi Zhang[1,2] Zhen Lei[1,2*] Stan Z. Li[3]
[1] CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] Center for AI Research and Innovation, Westlake University, Hangzhou, China.
{qi.zhang2017,zlei,szli}@nlpr.ia.ac.cn

***Abstract*—Although face recognition has achieved fairly remarkable results in recent years, it heavily relies on large-scale labeled data to train the high-capacity deep convolutional neural networks. It is unrealistic to collect larger labeled datasets to further boost the performance, which requires burdensome and expensive annotation efforts. Meanwhile, there exist numerous unlabeled face images. It is challenging but promising to jointly utilize limited labeled and abundant unlabeled data to obtain higher performance gain, which is the target of semi-supervised learning. In this paper, we propose a bottom-up method, Neighborhood-Aware Attention Network (NAAN), for semi-supervised face recognition. It clusters unlabeled face images by collaboratively predicting pairwise relations based on their neighborhood information, where the neighborhood is defined as a k-hop ego network centered in the given sample called "ego". Considering the different importance of neighbors, we employ the graph attention network to learn the ego's representation. We evaluate our model on two face recognition datasets MegaFace and IJB-A, and it yields favorably comparable performance to the fully-supervised results.**

***Keywords*—Semi-supervised Learning, K-hop Ego Network, Face Recognition**

I. Introduction

Face recognition is one of the most important topics in the area of biometrics research. The research focuses on face recognition mainly lie in several directions, such as the designs of loss functions and network structures. Loss functions have evolved to ones with more discriminative and generalization ability, such as contrastive loss [1], [2], and variants of both triplet loss [3]–[7] and softmax loss [8]–[11]. At the same time, many general deep convolutional neural networks (DCNNs) for image recognition, including AlextNet [12], ResNet [13] and DenseNet [14], have also been proposed to improve the performance on accuracy, speed and model size. Benefiting from the rapid developments, face recognition has made tremendous strides [11], [15]–[18]. However, it requires large-scale labeled images to train high-capacity networks. In fact, the annotated data are limited due to high labeling costs, but we can easily collect numerous unlabeled face images. In order to get rid of blindly relying on the exponential growth of labeled face images to further improve performance, it is desirable to learn better networks with limited labeled data and abundant unlabeled data, which is the target of semi-supervised learning.

In the early stages, many previous methods [19]–[21] for semi-supervised face recognition are based on the common assumption that the label space is shared between labeled and unlabeled data. They either propose a self-training method to increase the labeled dataset by adding unlabeled samples classified with the highest confidence using the trained PCA-based classifier [20], or propose a semi-supervised gallery dictionary learning to model both linear and non-linear variation and leverage the unlabeled data to learn a more precise gallery dictionary [19]. However, the assumption is inconsistent with the real-world scenarios where the environment of data acquisition is complex, such as the Internet and video surveillance. We can collect numerous unlabeled data while they are independent with the labeled samples without any identity overlapping. On this premise, face recognition has developed from the early closed-set problem, assuming that the labeled data contain identities of subjects that are enrolled in the unlabeled data, to the recent open-set problem, making no assumption on the relationships between labeled and unlabeled samples.

Some works [22]–[24] have noticed the shift towards the open-set setting. They do not assume data distribution in advance, and can naturally adapt to various situations. Although adopting different designs to cluster face images, these methods are applicable to the semi-supervised face recognition task. They can be divided into two different categories, the top-down methods and the bottom-up ones. The top-down approaches take a global perspective to cluster faces based on the data distribution. Li et al. [23] propose a framework to combine a detection and a segmentation module to pinpoint face clusters, which is inspired by Mask R-CNN [25]. It provides an insight to perform top-down face clustering but its two-stage implementation is too complex to deploy. The bottom-up approaches are able to perceive local data structures and cluster faces by predicting pairwise relations. CDP [22] proposes a mediator to aggregate opinions of pairwise relations derived from a committee formed by several models. Wang et al. [24] propose the Instance Pivot Subgraphs (IPS) constructed with the residual vectors between features of pivot and its neighbors. It performs the graph convolution network to predict the pivot-neighbor relations. They either aggregate

* Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

the neighborhood information with hand-crafted features (e.g., the mean and variance vector of neighbors), or use only pre-defined vector differences in pairwise relation modeling.

In this paper, we propose a Neighborhood-Aware Attention Network (NAAN) for semi-supervised face recognition, which aims at jointly making use of labeled and unlabeled data to further improve model performance trained using only limited labels. The point lies in how to pinpoint each face cluster among unlabeled data and assign pseudo-labels for them to expand the labeled training set. Different from the bottom-up methods above, our method learns to make collaborative link predictions between unlabeled pairs in a data-driven way, without any pre-design to model the relations. Firstly, we construct k-hop ego networks for all unlabeled samples. The k-hop ego network for the target node consists of the node (named "ego") itself and all its neighbors (and edges) within k hops. Then the candidate pairs are derived from the ego nodes and all their one-hop neighbors. In order to predict relations of candidate pairs, NAAN applies the graph attention network on the ego networks. It obtains egos' representations by assigning different attention scores to nodes in the neighborhood according to their importance. The collaborative relation prediction is conducted with the representations, containing the neighborhood information from two samples. We further merge positive pairs gradually to cluster unlabeled faces and perform pseudo label propagation. Finally, NAAN trains the DCNN with both labeled and pseudo-labeled data in a multi-task fashion.

In summary, the main contribution of our work is the proposed unified Neighborhood-Aware Attention Network (NAAN) for semi-supervised face recognition. By collaboratively inferring pairwise relations with the learned neighborhood-aware representations, NAAN performs pseudo label propagation in unlabeled face images. The representation of each sample is derived from performing feature smoothing with the graph attention network, where the neighborhood is defined as the k-hop ego network around the center sample. Our experimental results on two face datasets MegaFace and IJB-A demonstrate the superiority of our proposed model.

II. Related Work

Our method aims at solving the problem of semi-supervised face recognition to bridge the gap between learning with all labeled data and partially labeled data. With numerous unlabeled data, recent works often construct local sub-graphs and apply graph convolutional networks to infer relations between unlabeled samples.

A. Semi-supervised Face Recognition

Semi-supervised learning (SSL) [26]–[28] has been proven to be powerful to learn a better prediction rule with labeled and unlabeled data together than based on labeled data alone, mitigating the reliance on large labeled datasets. There exist various methods to boost the performance of SSL. Generative modeling [29]–[31] makes efforts to generate new data by fitting the original data distribution. Co-training [32]–[35] and tri-training [36], [37] are the representatives of disagreement-based methods which train multiple learners and exploit the disagreements to classify unseen instances. Graph-based methods [38]–[40] are proposed to propagate pseudo-labels on the naturally existed graphs (e.g., social networks) or human established graphs (e.g., k-nearest neighbors graphs).

In terms of face recognition, it has achieved remarkable results in recent years. However, the development is limited due to the difficulties of collecting larger labeled datasets. Some recent approaches [22]–[24] adopt the more realistic open-set setting to utilize unlabeled face images. CDP [22] is formed with two modules, the committee and the mediator, which selects positive face pairs by carefully aggregating multi-view information from various DCNNs. Wang et al. [24] propose to formulate the problem of face clustering as a linkage prediction problem. By constructing the Instance Pivot Subgraphs (IPS), it performs reasoning to predict the link between two unlabeled samples with the graph convolution network (GCN). Li et al. [23] formulate face clustering as a detection and segmentation pipeline based on GCN. It learns to cluster faces instead of relying on hand-crafted criteria.

B. Graph Convolutional Networks

Since the convolutional neural network (CNN) was proposed, it has been widely used in many fields, such as image classification and video processing, where data are typically presented as regular grids in the Euclidean space. In contrast, there many non-Euclidean structure data naturally exist in the real-world applications, such as social analysis [41]–[43], fraud detection [44], [45], traffic prediction [46] and computer vision [23], [24], [47]. These non-grid data are usually presented in the form of graphs. CNN cannot directly deal with them because of the various and complex structures of graphs. Graph convolutional networks (GCNs) are natural extensions of CNNs on the graph domain to explore relations and interdependency between objects in graphs.

Many efforts have been devoted to generalizing convolutional operations on the graph domain, which has been intensely studied mainly in two ways, the spectral-based GCNs [48]–[50] and the spatial-based GCNs [51], [52]. The former ones define the convolution operation in the spectral domain based on the graph Fourier transform, an analogy with 1-D signal Fourier transform. The latter ones directly define convolution operations in the graphs with nodes and their neighbors. They both take advantage of the rich neighborhood information to obtain node representations. Besides, attention mechanism recently has gained popularity and been applied to various applications. It has been introduced into GCNs to focus on the most informative parts of the input. Many methods [44], [51] have validated the efficacy of paying different attention to neighbors.

In this paper, we construct a nearest neighbor graphs based on the cosine similarity of representations. In order to control the information flows to aggregate message to the target nodes, GCN together with the attention mechanism are introduced to perform feature smoothing on the constructed graphs.

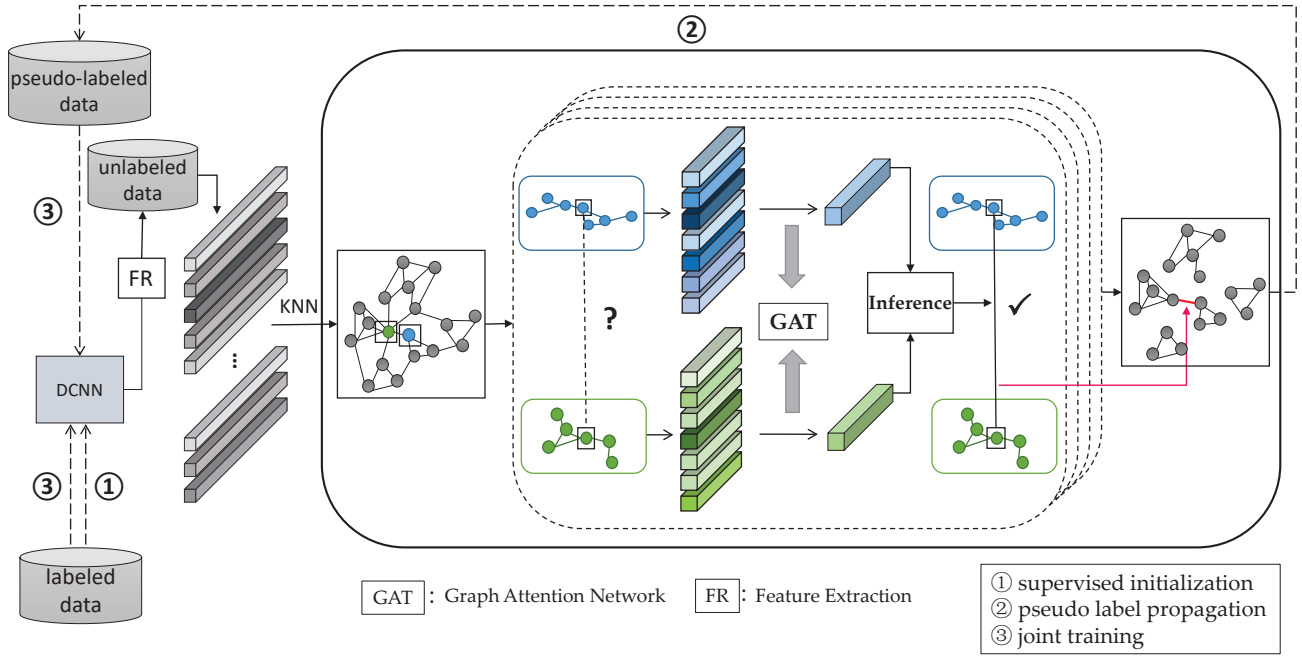


Fig. 1. The pipeline of our proposed Neighborhood-Aware Attention Network (NAAN) for semi-supervised face recognition. It consists of three modules, (a) supervised initialization for a DCNN with labeled data, (b) pseudo label propagation in unlabeled data after exploring pairwise relations with the graph attention network, (c) joint training with the labeled and pseudo-labeled data.

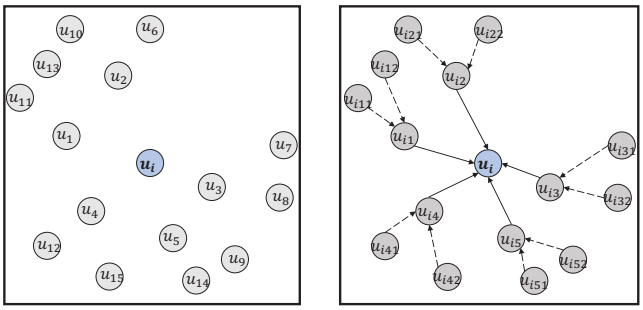
III. Method

In this section, we will present an overview of our proposed Neighborhood-Aware Attention Network (NAAN) for semi-supervised face recognition. As shown in Fig. 1, given labeled data, we first use them to initialize a DCNN and other parameters in a fully-supervised manner. We then extract features of unlabeled data with the pre-trained DCNN to construct a k-hop ego network for each sample. By applying neighborhood aggregation with the graph attention network, we generate neighborhood-aware embeddings. After collaborative predicting pairwise relations and merging positive pairs, the faces are clustered to perform pseudo label propagation for all unlabeled data. Finally, we jointly train labeled and unlabeled data in a multi-task fashion.

Next, we will introduce details of our proposed method from the following aspects: 1) supervised initialization, 2) pseudo label propagation, 3) joint training.

A. Supervised Initialization

In the large-scale semi-supervised learning scenario, only a small portion of data are labeled and a large number of data are without annotations. It is notable that there is no identity overlapping between labeled and unlabeled parts. We first train a DCNN equipped with the advanced ArcFace [11] serving as the loss function in a fully-supervised manner. The introduced ArcFace is proposed to obtain highly discriminative features for face recognition. With the trained DCNN, we then extract features for unlabeled data.



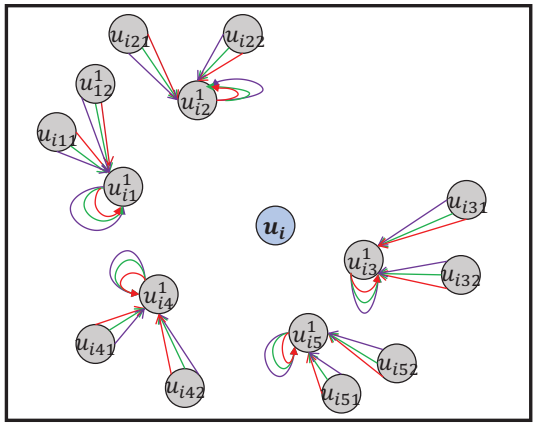
(a) Data Collection (b) 2-hop ego network

Fig. 2. The construction process of a 2-hop ego network. (a) The collection of features for unlabeled samples. (b) The 2-hop ego network centered in the ego node u_i. In the sub-figure (b), solid lines are used to represent the 1-hop neighborhood and dashed lines to present the second-order one for u_i. In the figure, we set k_1=5 and k_2=2, where k_1 is the number of 1-order neighbors and k_2 is the number of 2-order neighbors.

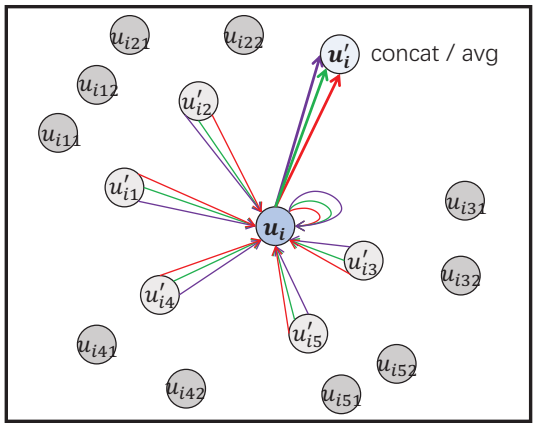
B. Pseudo Label Propagation

Inspired by the previous methods [22], [24], it is sufficient to predict the linkage likelihood between each sample and its nearest neighbors rather than all pairs to achieve a fairly good result. We adopt the same protocol to predict pairwise relations and assign a unique pseudo-label to each connected component. The pairwise relations are decided by comparing representations of two samples, which contain the local neighborhood information.

Construction of K-hop Ego Networks. We feed the trained DCNN with unlabeled images D_u as input and extract fea-



(a) Update from the second layer to the first layer



(b) Update from the first layer to the ego

Fig. 3. An illustration of the attentional modulation process between u_i and its neighborhood. It takes two steps to update the ego's representation. In this figure, the number of heads is set to three. Different line and arrow colors denote independent attention computations across multi-head attention. (a) Update from the second layer to the first layer. The 1-hop neighbors of u_i fuses information from their 1-hop neighborhood, respectively. (b) Update from the first layer to the ego. The final representation of u_i is concatenated or averaged the outputs of multi heads, which contains information from the whole neighborhood. The message aggregation flows from the outermost layer to the target node u_i.

tures, forming a set $U = \{u_i \in \mathbb{R}^d | i = 1, \ldots, N_u\}$, where d is the dimension of features and N_u is the number of unlabeled images. For the convenience of description, u_i represents both the extracted feature and the image itself in the following sections. For each unlabeled sample u_i, we introduce the construction of its k-hop ego network in detail. The ego network is the collection of nodes and their connected edges, where nodes include the focal one named "ego" and others that have connections to the ego. It usually refers to the one-hop ego network containing ego and nodes directly connected to it. In this paper, we extend the original definition to the k-hop ego network, containing the ego and all its k-hop neighbors. The k-hop neighbors range from 1-order ones to k-order ones of u_i. Specifically, we find the k_1 nearest neighbors for u_i, forming the one-hop neighborhood $\mathbf{u_i^1} = \{u_{i1}, u_{i2}, \ldots, u_{ik_1}\}$. In terms of each sample $u_{ij} \in \mathbf{u_i^1}$, we then find its k_2 nearest neighbors $\mathbf{u_{ij}^2} = \{u_{ij1}, u_{ij2}, \ldots, u_{ijk_2}\}$. Together with others, the 2-order neighborhood of u_i is formed as $\mathbf{u_i^2} = \{\mathbf{u_{i1}^2}, \ldots, \mathbf{u_{ik_1}^2}\}$. Similarly, we obtain the neighborhood from the 1-order one to the k-order one. All of them form the k-hop neighborhood of u_i as $\mathbf{u_i} = \{\mathbf{u_i^2}, \ldots, \mathbf{u_i^k}\}$. In Fig. 2, we provide an illustration of the construction of a 2-hop ego network centered in u_i. Actually, we set $k = 2$ in our experiments as some similar works [24], [44] to fully utilize the rich neighborhood information.

Attentional Modulation on Ego Networks. In order to infer on pairwise relations, we should first aggregate the neighborhood information and obtain representations of all unlabeled images. We adopt the idea of [51] to apply the graph attention network on the 2-hop ego networks centered in u_i. As shown in Fig. 3, it needs two steps to obtain the final representation. The message aggregation starts from the outermost layer to the target node u_i. In the first step, representations of the 1-hop neighbors of u_i are updated by their 1-hop neighbors, which are also the 2-order neighbors of u_i. We take the updating process of u_{ij} for example. The attention coefficient between u_{ij} and its 1-hop neighbor u_{ijt} is defined as:

$$e(u_{ij}, u_{ijt}) = f(Wu_{ij}, Wu_{ijt}), \tag{1}$$

where $W \in \mathbb{R}^{d' \times d}$ is the parameter matrix to project the original embeddings and $f(\cdot)$ is the attention function. $e(u_{ij}, u_{ijt})$ computes the importance of node u_{ijt} to u_{ij}. In our experiments, $f(\cdot)$ is a fully-connected layer with the parameter matrix $P_a \in \mathbb{R}^{2d'}$. Equation (1) is updated as:

$$e(u_{ij}, u_{ijt}) = \mathrm{LeakyReLU}(P_a[Wu_{ij}||Wu_{ijt}]), \tag{2}$$

where $\mathrm{LeakyReLU}(\cdot)$ is the activation function and $||$ represents the vector concatenation operation. To make coefficients easily comparable across different nodes, we introduce the softmax function to perform normalization as:

$$\alpha(u_{ij}, u_{ijt}) = \frac{\exp(e(u_{ij}, u_{ijt}))}{\sum_{m=0}^{k_2} \exp(e(u_{ij}, u_{ijm}))}, j \in [0, 1, ..., k_2] \tag{3}$$

where u_{ij0} is the neighbor with index 0 in the 1-hop neighborhood of u_{ij}, which is the node u_{ij} itself. In a sense, the attention coefficient $e(u_{ij}, u_{ij0})$ can be interpreted as one kind of neighborhood information derived from itself. By computing weighted combination of the 1-hop neighbors of u_{ij}, the representation is updated as:

$$u'_{ij} = \sigma(\sum_{t=0}^{k_2} \alpha(u_{ij}, u_{ijt}) W u_{ijt}), \tag{4}$$

where $\sigma(\cdot)$ is the activation function. We extend to use multi-head attention to stabilize the process of self-attention as:

$$u'_{ij} = \frac{1}{N_{h_1}} \sum_{l=1}^{N_{h_1}} \sigma(\sum_{t=0}^{k_2} \alpha_l(u_{ij}, u_{ijt}) W_l u_{ij}), \tag{5}$$

where N_{h_1} is the number of heads. It uses average pooling between different heads and the concatenation operation is another choice. Similarly, other 1-hop neighbors of u_i are

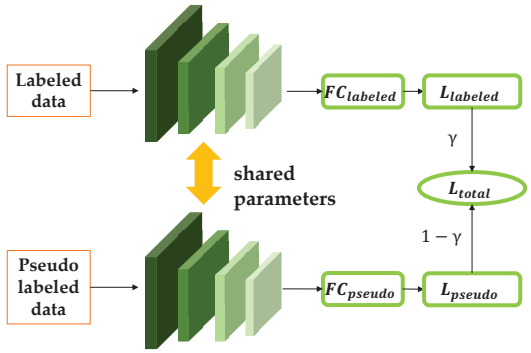


Fig. 4. The framework of joint training. The parameters of DCNNs for extracting features are shared in the two streams. The fully connected layers for training labeled and pseudo-labeled data are independent. The final objective function is the weighted combination of loss functions in the two streams.

updated their representations by incorporating the information of their k_2 nearest neighbors, respectively. The 1-hop neighborhood is updated as $(\mathbf{u_i^1})' = \{u'_{i1}, u'_{i2}, \ldots, u'_{ik_1}\}$.

The second step is to perform attentional modulation between u_i and its updated neighborhood $(\mathbf{u_i^1})'$. Finally, the representation of the ego u_i is written as follows:

$$u'_i = \frac{1}{N_{h_2}} \sum_{l=1}^{N_{h_2}} \sigma(\sum_{j=0}^{k_1} \alpha_l(u_i, u'_{ij}) W_l u'_{ij}), \tag{6}$$

where the attention coefficient $\alpha_l(u_i, u'_{ij})$ is computed as follows:

$$\alpha_l(u_i, u'_{ij}) = \frac{\exp(e(u_i, u'_{ij}))}{\sum_{m=0}^{k_1} \exp(e(u_i, u'_{im}))}. \tag{7}$$

In the updating process, we leverage the attention mechanism to identify informative neighbors. It calculates the optimal coefficients and adaptively adjusts the contribution of different neighbors. The new representation encodes the information of the whole neighborhood in the 2-hop ego network.

Inference on Pairwise Relations and Assignment of Pseudo-labels. Different from the previous methods [22], [24], we do not use the pre-defined designs on the 1-hop ($i.e.$ k_1 nearest neighbors) or 2-hop neighborhood to predict relations between two unlabeled samples. We make the collaborative prediction about pairwise relations based on both of their neighborhoods. In other words, the relation prediction of any two unlabeled samples is to compare their representations obtained by integrating their respective 2-hop ego network information. Actually, considering the computation efficiency, the candidate pairs are limited to nodes and its k_β nearest neighbors, where k_β is a hyper-parameter selected on the validation set. We only use the decision-maker to predict pairwise relations of candidate pairs. The decision-maker is formulated as a 2-layer MLP classifier to predict the link likelihood of each pair, which is trained in the labeled data in an end-to-end manner.

The data collection has as many egos as it has samples. We loop over all samples and follow the same process described above to obtain candidate pairs with linkage likelihood. Instead of setting a threshold and cutting all edges below it, we follow the same strategy as [22], [24] to perform pseudo label propagation. It uses all edges to form connected clusters and add them to a queue. Then it cuts off low-score edges if the size of connected cluster is larger than a fixed size and we re-add it to the queue. The low-score edges are those lower than the given threshold, which increase by the parameter η at each iteration. We will discuss the setting of η in the following sections in detail. The connected clusters with satisfying size are assigned unique pseudo-labels. The iterations are not finished until the queue is empty.

C. Joint Training

As shown in Fig. 4, we train labeled and pseudo-labeled data in a multi-task manner, which share parameters in feature extraction layers, $i.e.$, DCNNs. The fully-connected layers are different for non-overlapping classes between labeled data $X_l = \{(l_i, y_i)|i \in [1, N_l]\}$ and pseudo-labeled data $X_u = \{(u_i, p_i)|i \in [1, N_u]\}$. The joint loss function can be written as:

$$\begin{aligned} L &= \gamma L_{labeled} + (1-\gamma) L_{pseudo} \\ &= \gamma \sum_{i=1}^{N_l} c(l_i, y_i) + (1-\gamma) \sum_{i=1}^{N_u} c(u_i, p_i), \end{aligned} \tag{8}$$

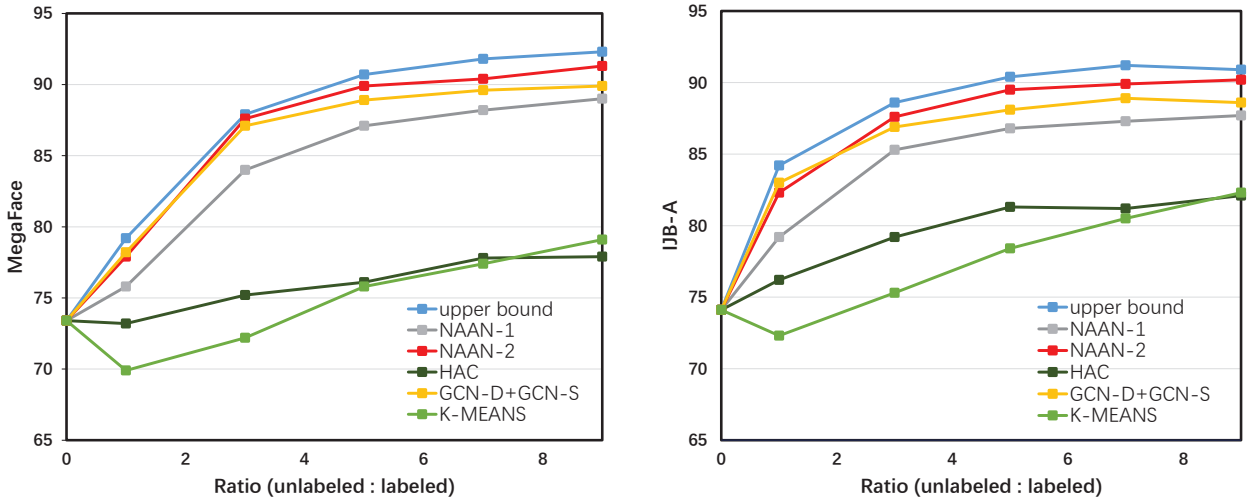
where $\gamma \in (0, 1)$ is a weighting parameter to balance two tasks and $c(\cdot)$ is the loss function. In our experiments, we adopt the same loss function ArcFace [11] as in supervised initialization stage.

IV. Experiments

A. Datasets and Metrics

Training set. We trained our proposed network on the MS-Celeb-1M dataset [53], which is one of the largest face recognition datasets containing 98,685 celebrities and 10 million images. Since the original dataset contains annotation noises, it is cleaned based on the annotations from ArcFace [11]. There are 5.8M images from 86k identities remaining. We follow [23] to split the dataset into 10 parts with an almost equal number of identities. Each part contains 8.6K identities with around 580K images, without any identity overlapping between different parts. The supervised initialization uses only one part as labeled data. These face images are horizontally flipped for data augmentation. The other nine part are regarded as unlabeled data. We randomly select one part as the validation set and adopt different experimental settings with 1, 3, 5, 7, 9 parts of unlabeled data as part of training set, respectively.

Testing set. We evaluate our network on two face recognition dataset MegaFace [54] and IJB-A [55]. MegaFace is the largest publicly available dataset for face recognition. It includes a gallery set with 1M images and a probe set from FaceScrub [56] with 3,530 images. Considering the noisy labels in



(a) MegaFace top-1 identification rate@1M

(b) IJBA TPR@FPR=0.001

Fig. 5. Comparison with other methods on MegaFace identification protocol and IJB-A verification protocol. In both sub-figures, all methods start from the same leftmost point where we only perform supervised initialization with labeled data. The upper bound refers to fully-supervised learning with corresponding numbers of labeled samples.

MegaFace, we follow ArcFace [11] to refine the dataset. IJB-A contains 5,712 images of 500 identities.

Evaluation Metrics. We adopt face identification benchmark and face verification protocol in MegaFace and IJB-A, respectively. The top-1 identification rate is used in MegaFace benchmark, which is the percentage of ground truth data appearing in the top-1 lists. For IJB-A benchmark, we use the true positive rate under the condition that the false positive rate is 0.001. Besides, in order to evaluate the performance of unlabeled samples in the training set, we introduce a widely used measurement, F-score, to take into account both pairwise precision P and recall Q. It is defined as $F = \frac{2PR}{P+R}$.

B. Implementation Details

We update nodes in the k-hop ego networks with two graph attention network layers. The first layer consists of 4 attention heads, following the second layer with 2 attention heads. The GCN layers and the 2-layer classifier are trained end-to-end and the Adam optimizer [57] is employed for optimization.

C. Experimental Comparison

1) Baseline Methods: In our experiments, we compare our proposed method with the following baselines, including both traditional methods and a recently published one.

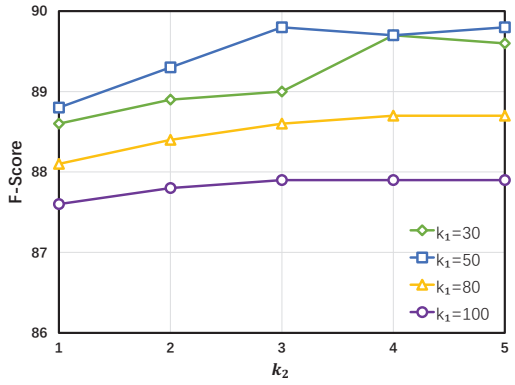
- **DCNN+K-Means Clustering** [58]. K-Means clustering is a popular clustering method, which partitions features extracted with the trained DCNN into k clusters by minimizing total within-cluster variances.
- **DCNN+HAC** [59]. Hierarchical agglomerative clustering (HAC) seeks to build a hierarchy of clusters, which relies on a linkage criterion which specifies the dissimilarity of sets.
- **LTC** [23]. LTC adopts a pipeline similar to the Mask R-CNN [25], combining a detection and a segmentation module to pinpoint face clusters. It is a two-stage version, where GCN-S is introduced to refine the output of GCN-D. GCN-S detects and discards noises inside clusters.
- **NAAN-1**. It is the implementation of our proposed model with singleton clusters (isolated points).
- **NAAN-2**. It is the version of NAAN **removing** singleton clusters after performing pseudo label propagation.

2) Results: As shown in Fig. 5, our proposed model has fairly good results on both benchmarks. We can observe that our method obtains significant and steady performance gain compared to the leftmost points where the supervised initialization process is conducted without unlabeled data. It verifies the effectiveness of the pseudo label propagation to cluster unlabeled face images. Both NAAN-1 and NAAN-2 surpass the traditional methods K-Means and HAC by a large margin. With a limited number of unlabeled data, K-Means even falls into performance degradation due to noisy pseudo-labels. We can see that our model outperforms the recent method LTC [23] and further improves the performance by 1.4% on MegaFace dataset with all unlabeled data. After removing singleton clusters, NAAN-2 performs better than NAAN-1 with higher recall scores. It is notable that NAAN-2 is close to the upper bound which is fully-supervised trained with corresponding numbers of labeled samples.

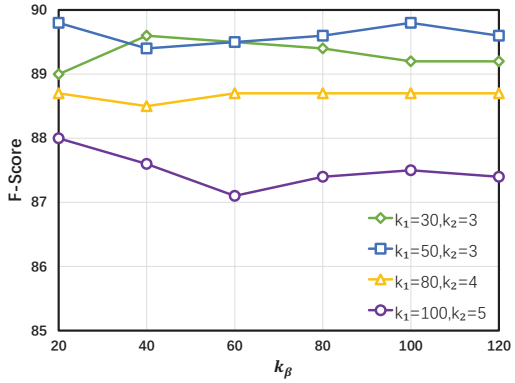
D. Ablation Study

In this section, we conduct ablation studies to quantify the effects of our proposed NAAN with different choices of pairwise prediction and various thresholds chosen in the pseudo label assignment process.

Different choices of pairwise prediction. In the training phase, we set $k_1 = 100$ and $k_2 = 5$ to obtain enough neighborhood information to distinguish the informative neighbors from mistaken ones. The parameter k_β is the number of



(a) Different choices of ego networks



(b) Different choices of k_β

Fig. 6. Ablation studies on k_β, k_1 and k_2. (a) With the fixed value $k_\beta = 20$, the influence of the range of neighbors in ego networks. (b) With various settings of ego networks, the influence of k_β to provide different numbers of candidate pairs.

candidate pairs. Considering the generalization ability, k_β is set to 100 to guarantee a relatively balanced sampling of negative and positive pairs. In the testing phase, we investigate the influences brought by different choices of k_β and ego networks, especially the settings of k_1 and k_2. Fig. 6 (a) indicates k_1 has a greater impact on the results compared with k_2, because the 1-hop neighbors are directly involved in the representation computation of the ego node in the second step of attentional modulation. NAAN achieves better performance with $k_1 = 50$ and $k_2 = 3$. As shown in Fig. 6 (b), higher k_β results in more candidate pairs, which include more false pairs to lower the precision scores but bring in more positive ones to improve the performance of recall scores. Besides, k_β is directly related to the computational costs. We find that $k_\beta = 20$ provides a relatively good trade-off between efficiency and performance. We finally take the setting of $k_1 = 50$, $k_2 = 3$ and $k_\beta = 20$ in the testing phase.

The influence of parameter η. The increasing factor η is to control the threshold to cut off low-score edges in the pseudo label assignment process. With various values of η, we report the number of clusters, pairwise precision, pairwise recall and F-score in Table I. With a larger value of η, the assignment will cut off more edges when the size of connected component

TABLE I
EVALUATION FOR DIFFERENT THRESHOLDS **W/WO** SINGLETON CLUSTERS ON MS-CELEB-1M.

	#clusters	precision	recall	F-score	discard ratio
with singleton clusters					
η=0.1	22756	73.6	**91.0**	81.3	-
η=0.2	23963	76.2	90.7	82.8	-
η=0.4	26585	80.4	90.2	85.0	-
η=0.6	30645	85.7	89.4	87.5	-
η=0.8	36417	**89.8**	88.0	**88.9**	-
without singleton clusters					
η=0.1	9087	73.0	**94.8**	82.5	2.3
η=0.2	9533	75.6	94.8	84.2	2.5
η=0.4	10338	79.8	94.8	86.7	2.8
η=0.6	11346	85.3	94.8	89.8	3.3
η=0.8	12626	**89.4**	94.6	**91.9**	4.1

is larger than the given number. It often leads to more clusters especially the singleton ones, which can be inferred from the comparison between the first and second group in Table I. We can see a trade-off between pairwise recall and pairwise precision from Table I. With the increase of η, edges with higher scores are preserved, leading to higher precision scores. At the same time, the pruning strategy will introduce some limited number of isolated points without any connection with other samples. Compared to the huge performance gain in pairwise precision, the performance degradation of pairwise recall is rarely small. As shown in the second group of Table I, after removing singleton clusters, the pairwise recall scores become steady and our method still selects high-precision pairs with different values of η. Despite discarding some samples, our model obtains purer pseudo-label clusters.

V. CONCLUSION

In this paper, we propose a unified Neighborhood-Aware Attention Network (NAAN) for semi-supervised face recognition to bridge the gap between learning with fully labeled data and partially labeled data. By applying graph attention network on the constructed k-hop ego networks, the context-aware representations of egos are derived from fully utilizing local sub-graph information. The pseudo label propagation is conducted after the collaborative prediction on pairwise relations of the learned representations of unlabeled data. Our model demonstrates its effectiveness by achieving comparable results when compared to the fully-supervised training on the datasets IJB-A and MegaFace.

As for the future work, we plan to extend our proposed method to perform semi-supervised learning on the general image recognition task.

ACKNOWLEDGEMENTS

This work has been partially supported by the Chinese National Natural Science Foundation Projects #61872367, #61876178, #61806196, #61806203, #61976229.

REFERENCES

[1] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005, pp. 539–546.

[2] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006.

[3] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *NeurIPS*, 2005, pp. 1473–1480.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.

[5] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[6] R. Manmatha, C. Wu, A. J. Smola, and P. Krähenbühl, "Sampling matters in deep embedding learning," in *ICCV*, 2017, pp. 2859–2867.

[7] W. Ge, W. Huang, D. Dong, and M. R. Scott, "Deep metric learning with hierarchical triplet loss," in *ECCV*, 2018, pp. 272–288.

[8] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *ICML*, 2016, pp. 507–516.

[9] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017, pp. 6738–6746.

[10] F. Wang, W. Liu, H. Dai, H. Liu, and J. Cheng, "Additive margin softmax for face verification," in *ICLR workshop*, 2018.

[11] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4690–4699.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012, pp. 1106–1114.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[14] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 2261–2269.

[15] J. Guo, X. Zhu, C. Zhao, D. Cao, Z. Lei, and S. Z. Li, "Learning meta face recognition in unseen domains," *arXiv preprint arXiv:2003.07733*, 2020.

[16] J. Guo, X. Zhu, Z. Lei, and S. Z. Li, "Face synthesis for eyeglass-robust face recognition," in *Chinese Conference on Biometric Recognition*. Springer, 2018, pp. 275–284.

[17] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, "Adaptiveface: Adaptive margin and sampling for face recognition," in *CVPR*, 2019, pp. 11 947–11 956.

[18] X. Zhu, H. Liu, Z. Lei, H. Shi, F. Yang, D. Yi, G. Qi, and S. Z. Li, "Large-scale bisample learning on ID versus spot face recognition," *IJCV*, vol. 127, no. 6-7, pp. 684–700, 2019.

[19] Y. Gao, J. Ma, and A. L. Yuille, "Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples," *IEEE Trans. Image Processing*, vol. 26, no. 5, pp. 2545–2560, 2017.

[20] F. Roli and G. L. Marcialis, "Semi-supervised pca-based face recognition using self-training," in *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2006 and SPR*, 2006, pp. 560–568.

[21] X. Zhao, N. W. D. Evans, and J. Dugelay, "Semi-supervised face recognition with LDA self-training," in *ICIP*, 2011, pp. 3041–3044.

[22] X. Zhan, Z. Liu, J. Yan, D. Lin, and C. C. Loy, "Consensus-driven propagation in massive unlabeled data for face recognition," in *ECCV*, 2018, pp. 576–592.

[23] L. Yang, X. Zhan, D. Chen, J. Yan, C. C. Loy, and D. Lin, "Learning to cluster faces on an affinity graph," in *CVPR*, 2019, pp. 2298–2306.

[24] Z. Wang, L. Zheng, Y. Li, and S. Wang, "Linkage based face clustering via graph convolution network," in *CVPR*, 2019, pp. 1117–1125.

[25] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *ICCV*, 2017, pp. 2980–2988.

[26] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. The MIT Press, 2006.

[27] Z. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowl. Inf. Syst.*, vol. 24, no. 3, pp. 415–439, 2010.

[28] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[29] E. L. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," *arXiv preprint arXiv:1611.06430*, 2016.

[30] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[31] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *NeurIPS*, 2014, pp. 3581–3589.

[32] Z. Zhou and M. Li, "Semi-supervised regression with co-training," in *IJCAI*, 2005, pp. 908–916.

[33] W. Wang and Z. Zhou, "A new analysis of co-training," in *ICML*, 2010, pp. 1135–1142.

[34] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998, pp. 92–100.

[35] M. Balcan, A. Blum, and K. Yang, "Co-training and expansion: Towards bridging theory and practice," in *NeurIPS*, 2004, pp. 89–96.

[36] Z. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, 2005.

[37] D. Chen, W. Wang, W. Gao, and Z. Zhou, "Tri-net for semi-supervised deep learning," in *IJCAI*, 2018, pp. 2014–2020.

[38] W. Chen, Y. Gu, Z. Ren, X. He, H. Xie, T. Guo, D. Yin, and Y. Zhang, "Semi-supervised user profiling with heterogeneous graph attention networks," in *IJCAI*, 2019, pp. 2116–2122.

[39] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-supervised learning with graph learning-convolutional networks," in *CVPR*, 2019, pp. 11 313–11 320.

[40] L. Chen and Z. Zhong, "Progressive graph-based subspace transductive learning for semi-supervised classification," *IET Image Processing*, vol. 13, no. 14, pp. 2753–2762, 2019.

[41] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *KDD*, 2014, pp. 701–710.

[42] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," in *KDD*, 2018, pp. 2110–2119.

[43] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *WSDM*, 2011, pp. 635–644.

[44] M. Liu, J. Liao, J. Wang, and Q. Qi, "AGRM: attention-based graph representation model for telecom fraud detection," in *ICC*, 2019, pp. 1–6.

[45] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Min. Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, 2015.

[46] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, 2018.

[47] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *CVPR*, 2017.

[48] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[49] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016, pp. 3837–3845.

[50] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *ICLR*, 2014.

[51] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[52] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *NeurIPS*, 2016, pp. 1993–2001.

[53] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *ECCV*, 2016, pp. 87–102.

[54] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *CVPR*, 2016, pp. 4873–4882.

[55] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. J. Burge, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: IARPA janus benchmark A," in *CVPR*, 2015, pp. 1931–1939.

[56] H. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *ICIP*, 2014, pp. 343–347.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[58] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–136, 1982.

[59] R. Sibson, "SLINK: an optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, 1973.