 This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/TCDS.2021.3137316, IEEE Transactions on Cognitive and Developmental Systems

Hierarchical Deep Multi-task Learning with Attention Mechanism for Similarity Learning

Yan Huang, Qicong Wang, Wenming Yang, *Senior Member, IEEE,* Qingmin Liao, *Senior Member, IEEE,* Hongying Meng

***Abstract*—Similarity learning is often adopted as an auxiliary task of deep multi-task learning methods to learn discriminant features. Most existing approaches only use the single-layer features extracted by the last fully connected layer, which ignores the abundant information of feature channels in lower layers. Besides, small cliques are the most commonly used methods in similarity learning task to model the correlation of data, which can lead to the limited relation learning. In this paper, we present an end-to-end hierarchical deep multi-task learning framework for similarity learning which can learn more discriminant features by sharing information from different layers of network and dealing with complex correlation. Its main task is graph similarity inference. We build focus graphs for each sample. Then, an attention mechanism and a node feature enhancing model are introduced into backbone network to extract the abundant and important channel information from multiple layers of network. In similarity inference task, a relation enhancing mechanism is applied to graph convolutional network to leverage the crucial relation in channels, which can effectively facilitate the learning ability of the whole framework. Extensive experiments have been conducted to demonstrate the effectiveness of the proposed method on person re-identification and face clustering applications.**

***Index Terms*—Hierarchical learning, multi-task, graph similarity inference.**

I. Introduction

DEEP metric learning (DML) as one of the similarity learning methods has attracted more and more attentions recently in the field of deep learning [1]–[7]. Its common strategy is to exploit a deep end-to-end feature representation. DML methods take the relations between samples into consideration and map samples into a new embedding space where samples with the same label are closer while the samples with different labels are far apart. However, the features learned by deep metric learning methods may yield suboptimal results if they only modeled simple relations of the data. For this reason, many deep learning methods introduce a multi-task learning mechanism [8], [9], i.e., optimizing the classification task and similarity learning task at the same time. In this way, these methods can achieve better results. However, there are still two problems existed in these methods.

Y. Huang and Q. Wang are with the Department of Computer Science, Xiamen University, Xiamen 361000, China. (e-mail:qcwang@xmu.edu.cn)

W. Yang and Q. Liao are with the Shenzhen International Graduate School/Department of Electronics Engineering, Tsinghua University, Shenzhen 518055, China.

H. Meng is with the Department of Electronic and Electrical Engineering, Brunel University London, Uxbridge UB8 3PH, UK. (email: hongying.meng@brunel.ac.uk)

1) A variety of existing methods only extract single-layer features from the last fully-connected layer of deep neural networks [10]–[12]. As a matter of fact, the features extracted from images in low layers have abundant details like position, while high-layer features contain semantic information like shapes and targets. Hence, single-layer features may be sensitive to variations like viewpoints and illumination.

2) Most deep multi-task metric learning methods organize the training samples into small cliques to compute their correlations, such as pairs [13], [14], triplets [15]–[17], quadruplets [10]. Accordingly, the learned features may be discriminative only in cliques while not in the whole embedding space due to limited correlation.

To deal with the first problem mentioned above, we intend to obtain hierarchical features with different descriptive capabilities from both low and high layers of the backbone network. However, the channels of these hierarchical features contain redundant information, that is, some channels play a greater role in distinguishing categories, while some other channels play a negligible role. In the process of network learning, the important discriminative information should be reinforced gradually. Multi-task learning based on feature sharing [8], [16], [18]–[21] has been proven to be an effective implicit data addition mechanism, which can make full use of the learned hierarchical information here. In order to solve the second problem, modeling more complex correlation of samples is a promising way. It is noteworthy that some methods [22], [23] developed graph structures with a specific number of nodes to describe rich contextual similarities, and achieved competitive results. However, the graph structure and feature learning of these methods are completely separated, thus the advantages of the graph structure in feature learning are not exploited.

With the above ideas, we propose an end-to-end hierarchical deep multi-task learning framework for similarity learning, which contains three parts, i.e., a shared hierarchical node feature embedding work with attention called shared hierarchical attention network (shared HA-Net), an auxiliary task for node classification, and a main task for graph similarity inference based on a relation enhancing graph convolutional network (RE-GCN). Considering the rich and complex relationship between samples, we treat each sample as a focus node and then build a focus graph (F-Graph) for it. The whole framework takes F-graphs as inputs, and then extracts the node feature embeddings hierarchically through the shared HA-Net, where the attention mechanism is introduced to recalibrate the features, so that important feature channels get more responses. Besides, we can further calculate more

discriminative information contained in these channels using a node feature enhancing (NFE) model. Then, under multi-task learning mechanism, all hierarchical discriminative feature information can be shared to realize the parallel learning of the two tasks. On one hand, an identification loss is employed to optimize the node classification auxiliary task. On the other hand, the shared feature embeddings and the structures of F-Graphs are both inputted into RE-GCN for graph similarity inference task. In particular, in order to boost the inference performance of GCN, we also introduce an attention mechanism in RE-GCN to stress the channels with important relations between nodes while suppressing the channels with inferior relations. With the powerful adjacency aggregation ability of RE-GCN, the main task of graph similarity inference can fuse relevant neighbor information effectively to improve the performance of the whole framework.

In summary, the main contribution of our work is threefold:

(1) An F-Graph is devised to represent the rich correlation of data and then a RE-GCN, in which attention mechanism is introduced between adjacency aggregation processes, is designed to strengthen the relations between the node feature channels with discriminative information, promoting the deep inference of complex similarity between samples.

(2) We propose an end-to-end hierarchical deep multi-task learning framework for similarity learning, where node classification task is to assist similarity inference task based on the RE-GCN. Furthermore, we design a shared HA-Net equipped with attention mechanism and the NFE model to overcome the sensibility of single-layer features. The whole framework can fully exploit the hierarchical features in which low layers describe details while high layers represent semantic information to improve the performance of similarity learning.

(3) Our approach was validated on four public datasets on two visual tasks, namely face clustering and person re-identification, and it confirmed that the proposed network was compared favorably against the state-of-the-art methods.

II. Related works

Due to the fact that distinguishing similar or dissimilar is ubiquitous in many fields, deep metric learning (DML) has been widely applied to person re-identification [24], [25], image clustering [26], [27], image retrieval [28]–[30], and so on. For example, in person re-identification, similarity measurement is indispensable for associating the same people across different cameras. In image clustering, we need to allocate the corresponding labels according to the similarities between samples. The purpose of similarity learning is to learn a similarity measurement method which can make the samples with the same labels more similar while the samples with different labels more dissimilar. So, the key barrier to similarity learning is how to enlarge the intra-class distance and decrease inter-class distance.

A lot of works have combined multi-task learning with deep metric learning for avoiding suboptimal results. For example, Yao et al. [8] integrated a identified loss and a metric learning loss into a unified multi-task framework for affective image retrieval and classification. Gao et al. [16] proposed a deep multi-task similarity learning approach for classification and novel class detection on high-dimensional data streams, which is optimized with a identify loss and a triplet loss. Cheng et al. [13] trained the model with a cross entropy loss and a contrastive loss for solving the problems of within-class diversity and between-class similarity. Sangkloy et al. [18] proposed a triplet network for sketch-based image retrieval, in which a common feature space for sketches and photos is learned by the joint training of a loss for identification and a triplet loss. Although these methods take the advantage of multi-task learning for deep similarity learning, they are still constrained in the following respects.

Due to the unique advantages of deep convolutional neural networks, a number of deep multi-task similarity learning methods only extract the features from the last fully connected layer [10]–[13], [17], [18], [31]. For example, Sangkloy et al. [18] extracted the features of triplets from the last fully connected layer of GoogLeNet. The triplets in [11] were passed through three networks with shared parameters and were also extracted from the last fully connected layer. But actually, low-layer features have higher resolution and contain more detailed information, while high-layer features have semantic information. Therefore, the extracted high-layer features are weak in details and sensitive to changes (e.g. variations of illumination and viewpoints), which makes similarity learning more difficult.

What is more, small cliques are the most common used methods in deep multi-task similarity learning, such as pairs [13], [32], [33], triplets [8], [11], [15], [16], [18], [31], and quadruplets [10], [34]. Zhang et al. [11] considered the relationships between triplets to learn the fine-grained feature representations effectively. With multi-task learning, Karaman et al. [10] designed quadruplet selection methods to improve the performance. It is not hard to see that these methods rely on sampling strategies, which may lead to limited performance. If the selected samples are easy to distinguish, then the performance can be limited to samples that hard to distinguish. Besides, these methods only focus on small cliques that have limited relationships, so that the learned feature embeddings may only discriminative in cliques that they belong to while not in the whole embedding space. However, it is worth noting that graph structures have been used to represent more complex correlation of data [22], [23]. For example, Shi et al. [23] constructed hypergraphs to formulate the relationships in visual data and mine the underlying relationships with a hypergraph-induced convolutional network. However, in these methods, the feature extraction and the graph similarity inference are two stages of complete separation. That is to say, the features input to the graph model are learned and fixed, and feature extraction can not be optimized according to the inference process of graph similarity. They are not an organic whole.

Consequently, we design a multi-task framework for deep similarity learning, which can extract hierarchical features and realize the complementary advantages of low-layer and high-layer features. Furthermore, similarity inference task based on graph structure are leveraged to explore the deeper latent

relationships between samples. Under the promotion of multi-task learning mechanism, this task can fully share hierarchical information to extract more discriminant features.

TABLE I
NOTATIONS AND DEFINITIONS

Notations	Definitions
o	the size of dataset
D	the set of all samples
v_f	a random node in D
$v_f^{(i)}$	i-th nearest node to v_f
$\mathcal{G}_{(f)}$	the graph constructed based on v_f
$V_{(f)}$	the node set of graph $\mathcal{G}_{(f)}$
$A_{(f)}$	the adjacent matrix of graph $\mathcal{G}_{(f)}$
$\widetilde{A}_{(f)}$	the Laplacian matrix of $\mathcal{G}_{(f)}$
V	the node sets of all graphs
$X_{(f)}^{(i)}$	the feature embeddings of a node in $\mathcal{G}_{(f)}$
$x_{(f)}^{(i)(c)}$	c-th channel of $X_{(f)}^{i}$
$y_{(f)}^{(i)(c)}$	the numerical descriptor of $x_c^{(i)}$
$X_{GAP}^{(i)}$	the feature embeddings of a node after GAP
$X_{GMP}^{(i)}$	the feature embeddings of a node after GMP
m	the feature embeddings of a node attained by the shared HA-Net
M_f	the node feature matrix of in graph $\mathcal{G}_{(f)}$
$Z_f^{(l)}$	the node feature matrix transformed by l-th GCN
$\widetilde{Z}_f^{(l)}$	the node feature matrix transformed by l-th RE-GCN
S_f	the inferred similarities between v_f and all the other nodes in graph $\mathcal{G}_{(f)}$
L_1	loss function for classification task
L_2	loss function for graph similarity inference task

III. METHODOLOGY

The proposed method utilizes graph structures to represent the rich relationships in data. With multi-task learning, the hierarchical information can be shared for similarity inference task. This section describes the hierarchical deep multi-task learning with attention mechanism for similarity learning in detail. The most important symbols are summarized in Table I.

A. Overview

We design a framework for hierarchical deep multi-task learning with attention for similarity learning to solving the problems mentioned above. The overall framework is shown in Fig. 1.

Firstly, the proposed graph construction method is utilized for original samples to build F-Graphs. Assuming that the set of samples is $D = \{v_1, v_2, \ldots, v_o\}$, where o represents the number of all samples. $\mathcal{G}_{(f)}$ is the F-Graph built with a node v_f in D as the focus node. $V_{(f)}$ denotes the sampled node set according to the focus node v_f and $A_{(f)}$ represents the graph structure of $\mathcal{G}_{(f)}$. As shown in Fig. 1, the node set $V_{(f)}$ is the input of shared HA-Net. Through shared HA-Net, shared feature embeddings m are achieved and treated as a bridge for two tasks. The first one is an auxiliary task, i.e. node classification. The other one is the main task, i.e. similarity inference. m and $A_{(f)}$ are both the input of RE-GCNs for similarity inference. Shared HA-Net is modified based on ResNet-50 [35], which attaches the attention block after each residual block and node feature enhancing model after the last three residual blocks. Then the last three feature embeddings are concatenated as shared feature embedding m. There are four RE-GCNs in architecture and each of them consists of two parts, i. e. graph convolutional layer and the proposed attention mechanism. Then with two fully connected layers, the similarities between nodes can be inferred. At last, two tasks are optimized with a joint loss. The details are depicted in the following sections.

B. The Construction of F-Graph

To consider the overall structure of data space, we propose to utilize graph structures to embed the relationships between samples. Due to large number of samples, it is impractical to embed all data into one graph. As a result, we propose to build an F-Graph for each sample.

We take an F-Graph based on focus node v_f as an example. As mentioned before, the F-Graph $\mathcal{G}_{(f)} = (V_{(f)}, A_{(f)})$ is built based on focus node v_f, where $V_{(f)}$ represents the nodes set that required to build $\mathcal{G}_{(f)}$ and $A_{(f)}$ is the adjacent matrix of $\mathcal{G}_{(f)}$, i.e., the graph structure. To build $\mathcal{G}_{(f)}$, we firstly need to determine which nodes should be choosen for $V_{(f)}$, then descide which connection will be built for $A_{(f)}$.

Firstly, the similarities between v_f and other nodes are calculated, which can be denoted as,

$$d(v_f, v_j) = (\sum_i (v_f^i - v_j^i)^2)^{\frac{1}{2}} \tag{1}$$

where $j \in \{t | 1 \leq t \leq o \ \& \ t \neq f \& \ j \in N^+\}$, $d(v_f, v_j)$ is the Euclidean distance between v_f and v_j. Based on Eq. (1), we calculate the distance between v_f and all the other nodes and then choose k nearest nodes for $V_{(f)}$, which can be defined as,

$$V_{(f)} = \underset{1 \leq j \leq o, \ j \neq f, \ j \in N^+}{min^{(k)}} d(v_f, v_j) \tag{2}$$

where $min^{(k)}$ represents choosing top k nearest nodes, $V_{(f)}$ denotes the nodes set of graph $\mathcal{G}_{(f)}$. $V_{(f)}$ can also be presented as a node matrix, i.e.,

$$V_{(f)} = [v_f^{(1)} \ v_f^{(2)} \ \ldots \ v_f^{(k)}] \tag{3}$$

where $v_f^{(i)}$ is the $i-th$ neighbor of v_f and $v_f^{(i)} \in D$. We build a F-Graph for each node and each F-Graph needs a node set. There are o nodes in dataset, so o node sets are needed. The node sets of o nodes are denoted as

$$V = [V_{(1)} \ V_{(2)} \ \ldots \ V_{(o)}]^T \tag{4}$$

where $V_{(i)}$ represents a node set of graph $\mathcal{G}_{(i)}$ which take v_i as the focus node. Once $V_{(f)}$ is attained, the connection between them will be built. For a non-focus node $v_f^{(i)}$ in graph $\mathcal{G}_{(f)}$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCDS.2021.3137316, IEEE Transactions on Cognitive and Developmental Systems

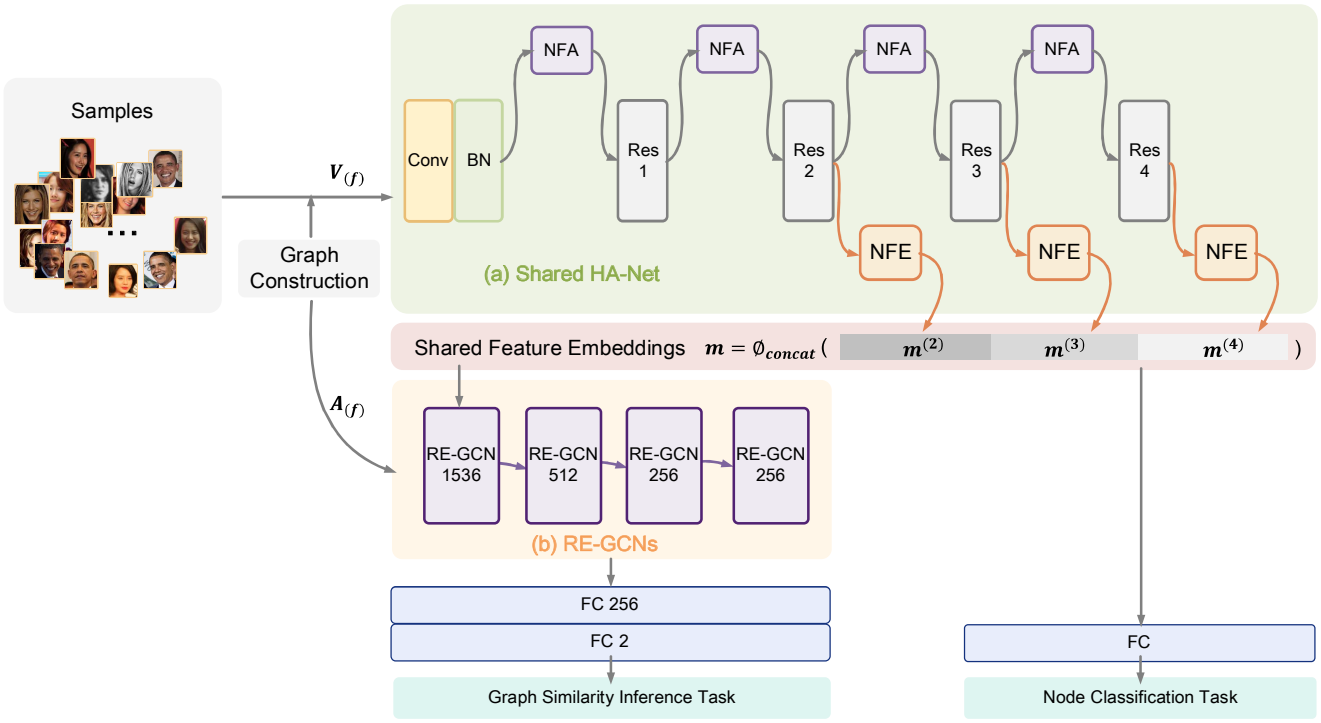

Fig. 1. The architecture of the proposed approach. Conv represents the convolutional layer and BN denotes batch normalization. Res 1, 2, 3, and 4 are four residual blocks. NFA and NFE are the proposed models which are described in Section III-C. RE-GCN 1536, 512, and 256 denote the output dimension of RE-GCN are 1536, 512, and 256 respectively. FC represents the fully connected layers.

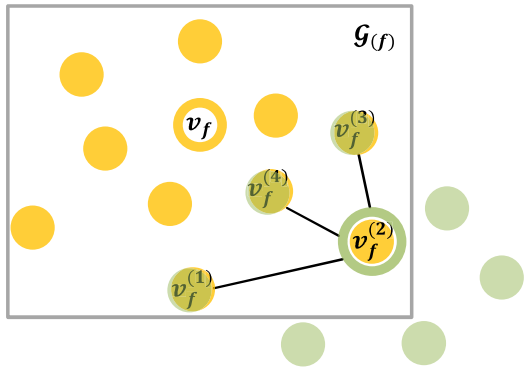

Fig. 2. F-Graph construction. The rectangle with gray borders denotes the F-Graph going to build. Yellow circles denote $V_{(f)}$. v_f denotes the focus node. Green circles denote the node set of v_f^2

there exists one F-Graph which takes $v_f^{(i)}$ as the focus node, so $v_f^{(i)}$ has its own node set. As shown in Fig. 2, if there are some nodes in the node set of $v_f^{(i)}$ and also appear in $V_{(f)}$, then connect these nodes with $v_f^{(i)}$.

In Fig. 2, v_f represents the focus node of $\mathcal{G}_{(f)}$ and the yellow nodes represent the node set $V_{(f)}$. For node $v_f^{(2)}$, it has its own node set which is shown in green and three of them are also in $V_{(f)}$, i.e. $v_f^{(1)}$, $v_f^{(3)}$, and $v_f^{(4)}$. As a result, $v_f^{(2)}$ will be connected to $v_f^{(1)}$, $v_f^{(3)}$, and $v_f^{(4)}$. We evaluate every node of $V_{(f)}$ in this way so that graph $\mathcal{G}_{(f)}$ is built.

C. Shared HA-Net and The Node Classification Task

To extract more discriminative feature embeddings and benefit for similarity inference, we propose a shared node feature embedding network based on hierarchical attention (shared HA-Net). The proposed shared HA-Net contains two important models, i.e. node feature embedding with attention (NFA) and node feature enhancer (NFE). Shared HA-Net takes $V_{(f)}$ as input. The architecture of shared HA-Net is illustrated in Fig. 3. ResNet-50 is chosen for the backbone which is mainly composed of a convolutional layer, max pooling layer, and four residual blocks. We add NFA model for each residual block and three NFE models for the last three residual blocks. The details of the NFA and NFE model are described as follows.

1) The NFA Model: Suppose the feature embeddings of a random node in $V_{(f)}$ after going through the four residual blocks are $X_{(f)}^{(i)} \in \mathbb{R}^{W\times H\times C}$ where $i \in \{1,2,3,4\}$, respectively. And $X_{(f)}^{(i)} = [x_{(f)}^{(i)(1)}, x_{(f)}^{(i)(2)}, \ldots, x_{(f)}^{(i)(C)}]$ where $x_{(f)}^{(i)(c)} \in \mathbb{R}^{W\times H}$, $x_{(f)}^{(i)(c)}$ represents the c-th channel of node feature embeddings $X_{(f)}^{(i)}$ in the i-th stage. The channel number of these four node feature embeddings are 64, 256, 512, and 1024 respectively. For $x_{(f)}^{(i)(c)}$, we first get the numerical descriptors of each channel with global average pooling

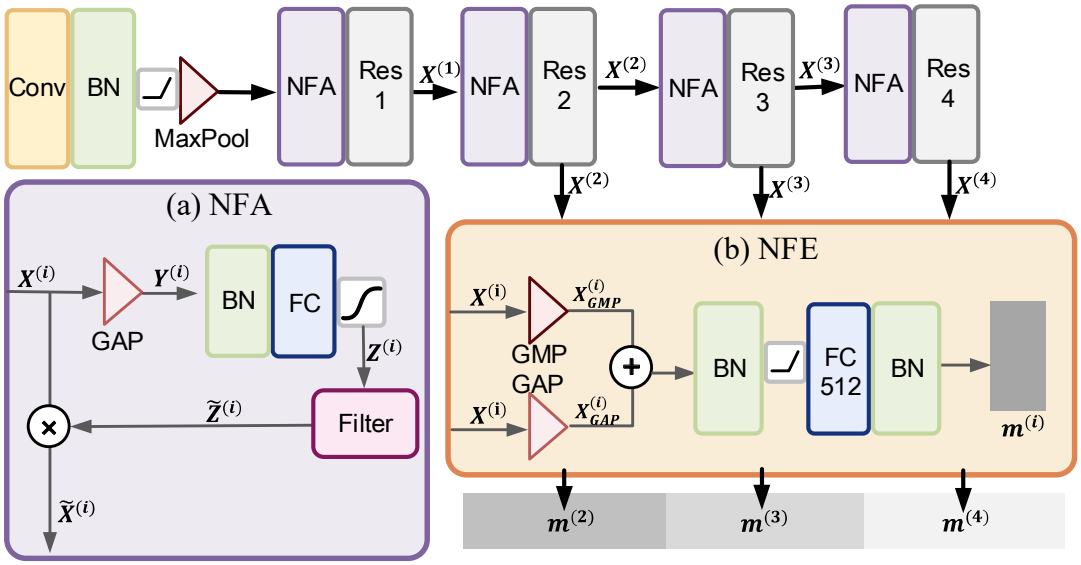

Fig. 3. The architecture of shared HA-Net. Conv is the convolutional layer and BN is batch normalization. Res 1, 2, 3, and 4 come from ResNet-50. The detailed architecture of NFA is also shown in (a). (b) is the proposed NFE model. GAP denotes global average pooling and GMP denotes global maximum pooling.

operation (GAP), i.e.,

$$y_{(f)}^{(i)(c)} = \frac{1}{\beta}\sum_{a=1}^{W}\sum_{b=1}^{H} x_{(f)}^{(i)(c)}(a,b) \tag{5}$$

where $\beta = W \times H$, $x_{(f)}^{(i)(c)}(a,b)$ represents the value at the (a,b) on the c-th channel in $X_{(f)}^{(i)}$, $a \in \{1,2,\dots,W\}$, and $b \in \{1,2,\dots H\}$. As a result, the numerical descriptor of $X_{(f)}^{(i)}$ is $Y_{(f)}^{(i)} = [y_{(f)}^{(i)(1)}, y_{(f)}^{(i)(2)}, \dots, y_{(f)}^{(i)(C)}]$. Then batch normalization, a fully connected layer are utilized for $Y_{(f)}^{(i)}$. To make the value of $Y_{(f)}^{(i)}$ range from 0 to 1, we use a sigmoid function. The process can be described as

$$B_{(f)}^{(i)} = Sigmoid(\psi(Y_{(f)}^{(i)})) \tag{6}$$

where $\psi(\cdot)$ denotes the fully connected layer. After the sigmoid function, the significant channels will have higher values. In order to make the $B_{(f)}^{(i)}$ applied reasonably, we adjust the value of it between 0.3 and 1. The new descriptor is defined as $\widetilde{B}_{(f)}^{(i)} = [\widetilde{b}_{(f)}^{(i)(1)}, \widetilde{b}_{(f)}^{(i)(2)}, \dots, \widetilde{b}_{(f)}^{(i)(C)}]$, where

$$\widetilde{b}_{(f)}^{(i)(c)} = \lambda_1 * \epsilon + \lambda_2 * b_{(f)}^{(i)(c)} + \lambda_3 \tag{7}$$

where $c \in \{1,2,3,\dots,C\}$, $b_{(f)}^{(i)(c)}$ represents the value of c-th channel in $B_{(f)}^{(i)}$. ϵ is a hyperparameter and we set it to 0.3 according to our empirical practice. If $b_{(f)}^{(i)(c)}$ is less than ϵ, λ_1 equals to 1, λ_2 and λ_3 equals to 0. If $b_{(f)}^{(i)(c)}$ is bigger than 0.9, λ_3 equals to 1, λ_1 and λ_2 equals to 0. Otherwise, λ_2 equals to 1, λ_1 and λ_3 equals to 0. So, minimum value of $\widetilde{B}_{(f)}^{(i)}$ is ϵ. And if some values are larger than 0.9, we set them to 1 which indicates that these channels are important. After that, we can achieve the feature embeddings with attention by using the weighted sum of numerical descriptors, which can be defined as

$$\widetilde{\mathbf{x}}_{(f)}^{(i)(c)}(a,b) = \mathbf{x}_{(f)}^{(i)(c)}(a,b) \times \widetilde{b}_{(f)}^{(i)(c)} \tag{8}$$

where the weighted feature embeddings can be used for the next stage. With the attention mechanism, the discriminative information in node feature embeddings has high responses while the interfering information has low responses.

2) The NFE Model: In order to make full use of the common information and the discriminative information in the channel, a node feature enhancer is proposed. The feature embeddings after the last three residual blocks are $X_{(f)}^{(i)} \in \mathbb{R}^{W \times H \times C}$ respectively, where $i \in \{2,3,4\}$. We leverage global average pooling (GAP) and global maximum pooling (GMP) for $X_{(f)}^{(i)}$. So $X_{(f)}^{(i)}$ is transformed to $X_{GAP}^{(i)}$ and $X_{GMP}^{(i)}$ respectively. $X_{GAP}^{(i)}$ means that each feature map is globally pooled and therefore has a global receptive field while $X_{GMP}^{(i)}$ mainly focus on the salient channels in feature maps. We attach a fully connected layer after the sum of $X_{GAP}^{(i)}$ and $X_{GMP}^{(i)}$, which can be denoted as

$$m^{(i)} = FC_1(X_{GAP}^{(i)} \oplus X_{GMP}^{(i)}) \tag{9}$$

where $\oplus$ represents the operation of adding the values at the corresponding entries of the features, FC_1 is the fully connected layer and the number of neurons is 512. After that, $m^{(2)}$, $m^{(3)}$ and $m^{(4)}$ are attained. We concatenate these feature embeddings along the feature dimension, i.e.,

$$m = \phi_{concat}(m^{(2)}, m^{(3)}, m^{(4)}) \tag{10}$$

where ϕ_{concat} is the operation of concatenation and the dimention of m is 1536. m denotes the feature embedding

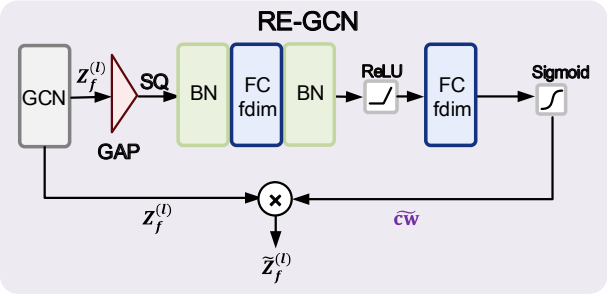

Fig. 4. The architecture of RE-GCN. GCN is the graph convolutional layer in Eq. (11). SQ is the operation of squeeze and fdim denotes the dimension of outputs.

of a node in graph $\mathcal{G}_{(f)}$, which is given by shared HA-Net. Similarly, the node feature embeddings of all nodes in $V_{(f)}$ can be achieved, we define it as M_f and $M_f \in \mathbb{R}^{k\times 1536}$.

3) The Node Classification Task: Next, M_f is used for two tasks, i.e. auxiliary task and main task. For auxiliary task, M_f will go through a fully connected layer for node classification task and we use a multi-class cross entropy loss, i.e.,

$$L_1 = -\frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{Cls} t_{ij} log(\hat{t}_{ij}) \tag{11}$$

where Cls represents the number of node classes. t_{ij} equals to 1 if node i belongs to class j and equlas to 0 if not. $\hat{t}_{ij}$ is the predicted probability of the node belonging to class j.

D. The Graph Similarity Inference Task Based on RE-GCNs

In order to mine the similarities between nodes in the F-Graph, a relation enhancing graph convolutional network is proposed. We add an attention mechanism between GCN layers, thus enhance the important relationships. Specifically, the nodes in the graph can take into account the correlation between samples with adjacency aggregation. Therefore, the proposed attention mechanism is able to make the channels with important relation information get more responses so that important relations are enhanced. The architecture of RE-GCNs is shown in Fig. 1 (b).

The shared HA-Net introduced before mainly process the node set $V_{(f)}$ and output a node feature matrix $M_{(f)}$. The RE-GCN takes $M_{(f)}$ and $A_{(f)}$ as input. The architecture of the proposed RE-GCN is shown in Fig. 4. In RE-GCN, the GCN can be expressed as

$$Z_f^{(l)} = \sigma(hstack(\widetilde{Z}_f^{(l-1)}, \widetilde{A}_f \cdot \widetilde{Z}_f^{(l-1)}) \cdot W^{(l-1)}) \tag{12}$$

where l denotes the layer number of graph convolution, $l \in \{1,2,3,4\}$, Z_f represents the results that M_f transformed by graph convolution, $\widetilde{Z}_f$ denotes Z_f with attention and $\widetilde{Z}_f^{(0)} = M_f$, $\widetilde{A}_f$ is the Laplacian matrix of graph and $\widetilde{A}_f = D^{-\frac{1}{2}} A_{(f)} D^{-\frac{1}{2}}$, D represents the diagonal degree matrix of $A_{(f)}$, $W^{(l-1)}$ is the leanable parameters of $(l-1)$-th graph convolutional layer, σ denotes the activation function, $hstack(\cdot)$ represents stacking two feature matrices horizontally.

After graph convolution, the node feature embeddings are embedded with neighbor information by adjacent aggregation. Then an attention model is designed to enhance the feature channels with important relationships, as illustrated in Fig. 4. Given $Z_f^{(l)} \in \mathbb{R}^{k\times fdim}$, where $fdim \in \{1536, 512, 256\}$, we first extend the dimension of $Z_f^{(l)}$ to $\mathbb{R}^{k\times fdim\times 1\times 1}$, then GAP is utilized to obtain the initial weights of feature channels cw. After that, cw is squeezed and batch normalized. Then, cw is input into a fully connected layer and an activation function ReLU. Finally, the sigmoid function is utilized to make the value of cw range from 0 to 1 and we use $\widetilde{cw}$ to denote it. Therefore, the enhanced node feature embeddings can be described as

$$\widetilde{Z}_f^{(l)} = Z_f^{(l)} \odot \widetilde{cw} \tag{13}$$

where $Z_f^{(l)} \in \mathbb{R}^{(k\times fdim)}$, $\widetilde{cw} \in \mathbb{R}^{k\times fdim}$, $\odot$ denotes the multiplication of the corresponding entries of the two matrices. To inference the similarities between nodes, we first initialize the edge features E_f with node feature matrix $\widetilde{Z}_f^{(4)}$, then a fully connected layer is leveraged to transform the node features to edge features. After an activation function, another fully connected layer is used to classify the edges into two classes. The process can be described as

$$S_f = softmax(\sigma_{e1}(E_f W_1 + b_1)W_2 + b_2) \tag{14}$$

where W_1 and b_1 are the parameters and bias of the first fully connected layer, σ_{e1} is the activation function, W_2 and b_2 are the parameters and bias of the second fully connected layer. S_f is the inferred similarities between the focus node and all the other nodes in $\mathcal{G}_{(f)}$. Similarly, we can obtain such similarities of all F-Graphs, which can be represented as S and $S = [S_1, S_2, \ldots, S_o]$.

In the training stage, we use a binary cross entropy loss function, i.e.,

$$L_2 = -\frac{1}{k}\sum_{i=1}^{k} y_{fi} log\hat{y}_{fi} + (1-y_{fi}) log(1-\hat{y}_{fi}) \tag{15}$$

where y_{fi} equals to 1 if the focus node v_f is truly connected to node v_i and equals to 0 if not. $\hat{y}_{fi}$ denotes the predicted probability that v_f connects to v_i. As a result, the joint loss of the proposed method is $L = L_1 + L_2$. With the optimization of the joint loss, the auxiliary task can help the main task improve the performance for similarity learning.

In the testing stage, the rank results of person Re-ID could be easily achieved by sorting the similarities in S. For face clustering, we need to design pseudo label propagation. It focuses on whether two nodes that have the same true label are assigned to the same class rather than the labels it assigned. Consequently, we first traverse S. In every iteration, the nodes that are connected with high edge weights will be added to the same class. But if the number of nodes in a certain class is bigger than a predefined number, then this class is left for the next iteration. When iterations are finished, the clusters are obtained and we assign labels to these clusters starting from 1.

IV. EXPERIMENTS

A. Datasets and Settings

In order to verify the effectiveness of the proposed method, we evaluate it on two tasks, i.e. person Re-ID and face clustering.

1) Person Re-ID: Market-1501 dataset [36] and DukeMTMC-reID dataset [37] are used for evaluating the proposed method on person Re-ID task. Market-1501 dataset used six cameras, including 5 high-resolution cameras, and one low-resolution camera. The dataset contains 32,668 annotated bounding boxes of 1,501 identities which were collected in Tsinghua University. There are 12,936 images of 751 identities for training and 19,732 images of 750 identities for testing. DukeMTMC-reID dataset is a subset of the DukeMTMC dataset. It contains 36,411 annotated bounding boxes of 1,404 identities. There are 17,661 images of 702 identities for training, 16,522 images of 702 identities for testing and 2,228 imagse as query images.

Cumulative Matching Characteristics (CMC) curves is the most common evaluation metric for person Re-ID task. The abscissa of the curve is the number of rank, and the ordinate is the recognition rate. Rank-n denote the top n results in the descending order of similarity that contain targets. The recognition rate denotes the ratio of Rank-n to the total number of query samples.

2) Face Clustering: For face clustering, we adopted the CASIA-WebFace dataset [38] for training, the IARPA Janus Benchmark-B (IJB-B) dataset [39] and the IARPA Janus Benchmark-C (IJB-C) dataset [40] for testing. The CASIA-WebFace dataset is collected from the Internet and contains 494,414 images of 10,575 subjects. 5,000 subjects were randomly chosen for training. IJB-B and IJB-C datasets were proposed by the National Institute of Standards and Technology (NIST). IJB-B provides seven subsets for face clustering. There are 32, 64, 128, 256, 512, 1024, 1845 subjects in seven subsets, respectively. We choose the largest three subsets for testing. IJB-C dataset is an upgraded version of IJB-B dataset. It has four subsets which contains 32, 1021, 1839, 3531 subjects, respectively. We select the largest three subsets that contains 41,074, 71,392, and 140,623 images respectively.

The evaluation criteria of face clustering are diverse, but the normalized mutual information (NMI) is commonly used, so we adopt it as an evaluation metric. Mutual Information can measure the similarity of two data distributions. Suppose C is the true clusters of N samples and $\hat{C}$ is the predicted clusters of N samples, then the entropies of two distributions can be denoted as

$$H(C) = \sum_{i=1}^{|C|} P(i) log(P(i))$$

$$H(\hat{C}) = \sum_{j=1}^{|\hat{C}|} P^{'}(j) log(P^{'}(j))$$

where $p(i) = |C_i|/N$ and $P^{'}(j) = |\hat{C}_j|/N$, then MI has the formulation

$$MI(C, \hat{C}) = \sum_{i=1}^{|C|} \sum_{j=1}^{|\hat{C}|} P(i,j) log(\frac{P(i,j)}{P(i)P^{'}(j)})$$

where $P(i,j) = |C_i \bigcap \hat{C}_j|/N$, so the NMI is denoted as

$$NMI(C, \hat{C}) = \frac{MI(C, \hat{C})}{\sqrt{H(C)H(\hat{C})}}$$

The value of NMI ranges from 0 to 1 and the high value represents the predicted clusters are similar to the real situation.

B. Implementation Details

*1) **System Settings**:* We implemented the proposed method with Pytorch deep learning framework, including torch 1.6.0, cudnn 7.6.3, CUDA 10.1.243. The python version is 3.8.5. The hardware of the server contains 12G GeForce RTX 2080 Ti, Intel(R) Core(TM) i9-9820X CPU @ 3.30GHz. The operating system is Ubuntu 16.04.7 LTS.

*2) **Training Settings**:* The origin images are all resized to 256×128 and randomly horizontal flipped for data augmentation. Stochastic Gradient Descent (SGD) is utilized to optimize the proposed model with an initial learning rate of 0.001 and the momentum is 0.9. The value of weight decay is 0.0001. The proposed model is trained for 20 epochs. At epoch 12, 16, and 18, the learning rate is multiplied by 0.1 at each time.

C. Ablation Study

In order to demonstrate the impact of each part of the proposed method, various experiments are conducted on Market-1501 dataset, as shown in Table II.

In these ablation experiments, the graph construction parameter k is set to 50 and the layer number l is set to 4. In Table II, "Variants Numeration" denotes the number for marking variants. "CNN" denotes ResNet-50 baseline. "Graph Process" means which kind of graph convolution is utilized, "GCNs" represents four layers of graph convolutional layers in Eq. (12), and "RE-GCNs" denotes four layers of the proposed RE-GCNs.

There are three NFE models in the proposed method. So "1st" denotes the first NFE model, "2nd" denotes the second NEF model, and "3rd" denotes the third NFE model. "NFA" denotes the introduced NFA model in shared HA-Net. "mAP" and "Rank-1" represent the mAP abd Rank-1 performance. "✓" denotes using this part.

Variant 1 leverages the graph convolution in Eq. (12). Compared with variant 6, the performance is 88.41% on mAP and 95.96% on Rank-1, i.e., the RE-GCNs brings the improvement of 1.24% on mAP and 2.37% on Rank-1. In RE-GCNs, the feature embeddings have aggregated the information of neighbors after GCN. Then, with attention mechanism, these feature embeddings are recalibrated. With the help of training, the RE-GCNs make the channels with discriminative information gain larger weights. This is the reason that the performance of variant 6 is better than variant 1.

TABLE II
ABLATION STUDY FOR REVEALING THE IMPACT OF IMPROTANT COMPONENTS ON FINAL PERFORMANCE

Variants Numeration	CNN	Graph Process		NFE			NFA	mAP	Rank-1
		GCNs	RE-GCNs	1st	2nd	3rd			
1	✓	✓		✓	✓	✓	✓	0.8841	0.9596
2	✓		✓	✓	✓	✓		0.8949	0.9821
3	✓		✓				✓	0.8753	0.9459
4	✓		✓			✓	✓	0.8920	0.9789
5	✓		✓		✓	✓	✓	0.8947	0.9830
6	✓		✓	✓	✓	✓	✓	**0.8965**	**0.9833**

Variant 2 removes the NFA model in contrast to variant 6. As shown in Table II, the performance of variant 2 is worse than variant, i.e., the performance gain of the NFA model is 0.16% on mAP and 0.12% on Rank-1.

Variant 3 to variant 5 are to demonstrate the effectiveness of the NFE model and the hierarchical features. Variant 3 removes the NFE model in contrast to variant 6 and the performance gain of the NFE model is 2.12% on mAP and 3.74% on Rank-1, which improves the effectiveness of the proposed NFE model. Variant 4 only uses the high-layer features for classification and variant 5 uses the second and the third ones. The hierarchical features of variant 6 bring the mAP improvement of 0.45% compared to variant 4 and 0.18% compared to variant 5. The hierarchical features of variant 6 bring the Rank-1 improvement of 0.44% compared to variant 4 and 0.03% compared to variant 5. Low-layer features provide detailed information that can contribute to discriminative features. And high-layer features possess semantic information, the importance of which is self-evident. Consequently, the performance of variant 6 is better than variant 4 and 5.

From variant 1 to 6, it is not hard to learn that every component in isolation improves the performance. By means of these components, the proposed method (i.e. variant 6) reaches the mAP of 89.65% and Rank-1 of 98.33%.

D. Parameters Analysis

TABLE III
PARAMETER k INFLUENCE ON PERFORMANCE

k	mAP	Rank-1	Rank-5	Rank-10
10	0.8762	0.9655	0.9955	0.9997
20	0.8839	0.9685	0.9928	0.9994
30	0.8842	0.9673	0.9934	0.9994
40	0.8893	0.9720	0.9946	0.9997
50	0.8928	0.9756	0.9970	0.9994

In this section, we are varying the layer number of RE-GCN l and the graph construction parameter k to analyze their influence. l denotes the number of layers of RE-GCN involved and k denotes the number of nodes chosen for $V_{(f)}$. The results are reported on Market-1501 dataset.

TABLE IV
PARAMETER l INFLUENCE ON PERFORMANCE

l	mAP	Rank-1	Rank-5	Rank-10
2	0.8897	0.9759	0.9967	0.9994
3	0.8928	0.9756	0.9970	0.9994
4	0.8965	0.9833	0.9955	0.9997

k is varied in $\{10, 20, 30, 40, 50\}$ and the layer number of RE-GCNs is set to 3. k equals to 10 means there are 10 nodes in the graph. The results are shown in Fig. 5. We use k_i to denote the situation when k equals to i.

As shown in Fig. 5, the mAP gain of k_{50} is 0.35% compared to k_{40}, 0.86% compared to k_{30}, 0.89% compared to k_{20}, and 1.66% compared to k_{10}. And the Rank-1 gain of k_{50} is 0.36% compared to k_{40}, 0.83% compared to k_{30}, 0.71% compared to k_{20}, and 1.01% compared to k_{10}. In conclusion, the results are relatively stable and reach the best performance when k is set to 50.

l is varied in $\{2, 3, 4\}$ and k is set to 50. The results are shown in Fig. 6. Also, we use l_i to denote the situation when l sets to i. As shown in Fig. 6, the mAP gain of l_4 is 0.37% compared to l_3, and 0.68% compared to l_2. The Rank-1 gain of l_4 is 0.77% compared to l_3, and 0.74% compared to l_2. The accuracy is relatively stable and reaches the best performance when l is set to 4.

E. Performance Comparison

1) Comparison on Person Re-ID Task: The proposed method is compared to other state-of-the-art methods on Market-1501 and DukeMTMC-reID datasets that are briefly introduced as follows.

MHN [41] treated person Re-ID as zero-shot learning task and proposed mixed high-order attention network. CBN [42] considered the distribution of all cameras and proposed a camera-based formulation. SAN [43] is a sampling-based attention mechanism which is sharper than gating-based soft

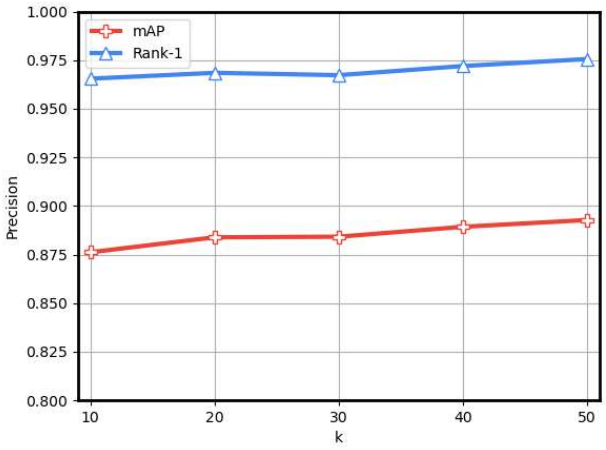

Fig. 5. Parameter k influence on performance

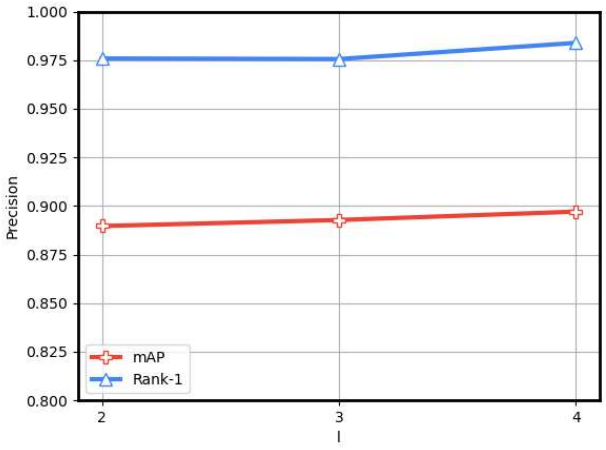

Fig. 6. Parameter l influence on performance

TABLE V
RESULTS COMPARISONS OVER MARKET-1501

Methods	Ref	Market-1501			
		mAP(%)	Rank-1(%)	Rank-5(%)	Rank-10(%)
MHN [41]	ICCV2019	85.0	95.1	98.1	98.9
CBN [42]	ECCV2020	83.6	94.3	97.9	98.7
SAN [43]	TCSVT2019	70.1	85.9	94.9	97.0
PCB + RPP [44]	ECCV2018	81.6	93.8	97.5	98.5
MuDeep [45]	TPAMI2020	84.6	95.3	98.1	98.7
DLPA [46]	ICCV2017	63.4	81.0	92.0	94.7
MVP [47]	ICCV2019	80.5	91.4	-	-
pyramidal [48]	CVPR2019	88.2	95.7	98.4	99.0
SVDNet [49]	ICCV2017	62.1	82.3	92.3	95.2
Structural [50]	TNNLS2019	67.3	84.3	93.6	96.0
Group-shuffling [51]	CVPR2018	82.5	92.7	96.9	98.1
SGGNN [52]	ECCV2018	82.8	92.3	96.1	97.4
CACE-Net [53]	arXiv2020	**90.3**	95.9	-	-
Proposed		89.6	**98.3**	**99.5**	**99.7**

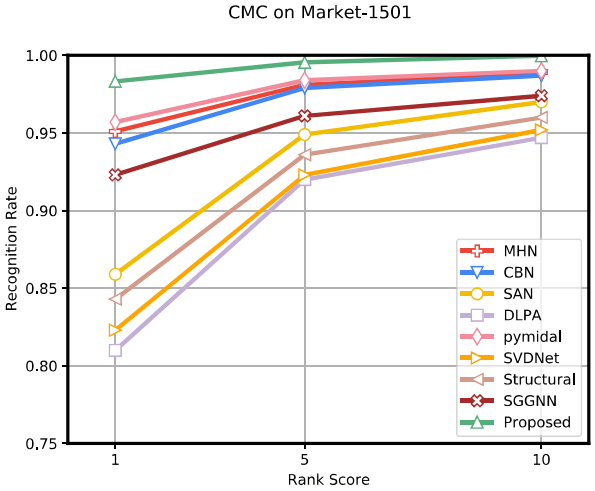

Fig. 7. CMC comparison on Market-1501 dataset

attention. PCB + RPP [44] is a part-based convolutional baseline with refined part pooling for learning part-level features. MuDeep [45] is a multi-scale deep learning model for person Re-ID. DLPA [46] computes the representations over the regions that are discriminative.

The proposed method is also compared to deep metric learning methods. MVP [47] is a method for mining hard sampled pairs within metric learning framework. Pyramidal [48] is a coarse-to-fine pyramidal model. SVDNet [49] optimizes the deep representation learning process with singular vector decomposition. Structural [50] utilizes a hardness-aware structural metric learning objective for learning feature representations and distance metric.

In addition, the proposed method is compared to graph-based methods. Group-shuffling [51] method has a novel group-shuffling random walk layer to obtain probe-to-gallery affinities. SGGNN [52] incorporates the inter-gallery-image relations to enhance feature learning process. CACE-Net [53] integrated visual clue alignment and conditional feature embedding for person Re-ID.

The comparison results of CMC curves on Market-1501 dataset are shown in Fig. 7. It can be seen that the proposed method achieves the best performance from Rank-1 to Rank-10. The numerical results are shown in Table V.

As shown in Table V, the mAP improvement of the proposed method is 4.6% against MHN, 6.0% against CBN, 19.5% against SAN, 8.0% against PCB + RPP, 5.0% against MuDeep, and 26.2% against DLPA. Compared to deep metric learning methods, the mAP improvement of the proposed method is 9.1% against MVP, 1.4% against pyramidal, 27.5% against SVDNet, and 22.3% against Structural. Compared to graph-based methods, the mAP improvement of the proposed method is 7.1% against Group-shuffling and 6.8% against SGGNN. The mAP of CACE-Net is higher than ours, but the Rank-1 improvement of the proposed method is 2.4% against CACE-Net.

The proposed method is also evaluated on DukeMTMC-reID dataset. P^2-Net [54] extracts dual part-aligned representations for person Re-ID. SPReID [55] employs human semantic parsing to hardness local visual cues. CL [58] reduces the inter-class correlation with orthogonalization. PIE [56] is pose invariant embedding for person re-identification. AVA-reID [57] is a principled adversarial feature learning approach to

TABLE VI
RESULTS COMPARISONS OVER DUKEMTMC-REID

Methods	Ref	DukeMTMC-reID			
		mAP(%)	Rank-1(%)	Rank-5(%)	Rank-10(%)
MHN [41]	ICCV2019	77.2	89.1	94.6	96.2
CBN [42]	ECCV2020	70.1	84.8	92.5	95.2
P^2-Net [54]	ICCV2019	73.1	86.5	93.1	95.0
SPReID [55]	CVPR2018	71.0	84.4	91.9	93.7
PIE [56]	TIP2019	64.1	80.8	88.3	90.7
AVA-reID [57]	TCSVT2020	67.2	80.1	89.5	-
MVP [47]	ICCV2019	70.0	83.4	-	-
CL [58]	TIP2021	79.0	87.7	94.1	96.1
pyramidal [48]	CVPR2019	79.0	89.0	94.7	96.2
SVDNet [49]	ICCV2017	56.8	76.7	86.4	89.9
Group-shuffling [51]	CVPR2018	66.4	80.7	88.5	90.8
SGGNN [52]	ECCV2018	68.2	81.1	88.4	91.2
CACE-Net [53]	arXiv2020	81.3	90.9	-	-
Proposed		**85.4**	**95.3**	**96.9**	**97.8**

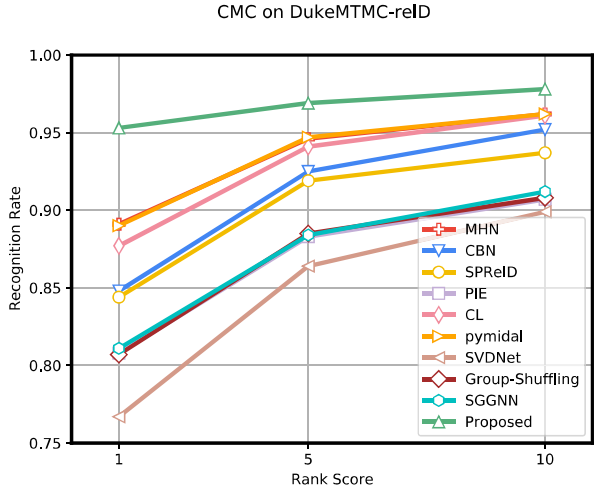

Fig. 8. CMC comparison on DukeMTMC-reID dataset

learn a latent viewinvariant feature space.

The comparison results of CMC curves on DukeMTMC-reID dataset are shown in Fig. 8. It can be seen that the proposed method achieves the best performance from Rank-1 to Rank-10. The numerical results are shown in Table VI.

As shown in Table VI, the mAP improvement of the proposed method is 8.2% against MHN, 15.3% against CBN, 12.3% against P^2-Net, 14.4% against SPReID, 21.3% against PIE, and 18.2% against AVA-reID. Compared to deep metric learning methods, the mAP improvement of the proposed method is 15.4% against MVP, 6.4% against CL, 6.4% against pyramidal, and 28.6% against SVDNet. Compared to graph-based methods, the mAP improvement of the proposed method is 19.0% against Group-shuffling, 17.2% against SGGNN, and 4.1% against CACE-Net.

In comparison with deep metric learning methods, i.e., MVP, CL, pyramidal, and SVDNet, the proposed method uses F-Graphs to allow more abundant relations of data to be considered. Otherwise, the relations in triplets or quadruplets are constrained in taking the whole feature embedding space into account. In comparison with graph-based methods, the proposed method utilizes RE-GCNs to strengthen the node feature channels with discriminative information. Besides, a shared HA-Net is designed to assist the RE-GCNs to infer similarities, where NFA model for attention mechanism and NFE model for feature enhancing are both used.

2) Comparison on Face Clustering Task: The proposed approach is compared to other methods on IJB-B and IJB-C datasets introduced as follows.

K-Means [59] clusters samples based on the distance between them with the known number of clusters. DBSCAN [60] is a clustering method based on density. It assumes that the density of samples that have the same labels is small. So the samples with small density can be classified into the same class. EnSC [61] is a subspace clustering method. Spectral [62] is a subspace clustering method. ARO [63] achieved the desired scalability and accuracy with a Rank-Order clustering algorithm. GCN [64] is a GCN-based method for face clustering. The feature embeddings extracted by ResNet-50 are utilized for all methods. The results on IJB-B dataset are shown in Table VII and Fig. 9. The results on IJB-C dataset are shown in Table VIII and Fig. 10.

TABLE VII
METHOD COMPARISON ON THREE SUBSETS OF IJB-B

Methods	NMI		
	IJB-B-512	IJB-B-1024	IJB-B-1845
K-Means [59]	0.7193	0.6784	0.6392
DBSCAN [60]	0.5058	0.5123	0.5120
EnSC [61]	0.6462	0.6008	0.5572
Spectral [62]	0.7840	0.7920	0.7850
ARO [63]	0.8299	0.8312	0.8379
GCN [64]	0.7594	0.7770	0.7853
proposed	**0.8552**	**0.8666**	**0.8721**

TABLE VIII
METHOD COMPARISON ON THREE SUBSETS OF IJB-C

Methods	NMI		
	IJB-C-1021	IJB-C-1839	IJB-C-3531
K-Means [59]	0.7683	0.7708	0.7932
DBSCAN [60]	0.5511	0.5352	0.5153
EnSC [61]	0.6318	0.5681	0.5029
ARO [63]	0.8503	0.8533	0.8543
GCN [64]	0.7650	0.7831	0.7932
proposed	**0.8943**	**0.8996**	**0.8995**

From Fig. 9 and Fig. 10, it can be seen that the proposed method achieves the best performance compared to other methods. ARO and K-Means perform well on both datasets, but the NMI performance of DBSCAN and EnSC is relatively low.

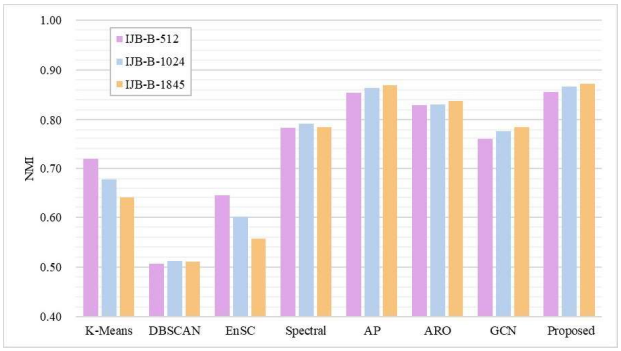

Fig. 9. Comparison results on IJB-B dataset

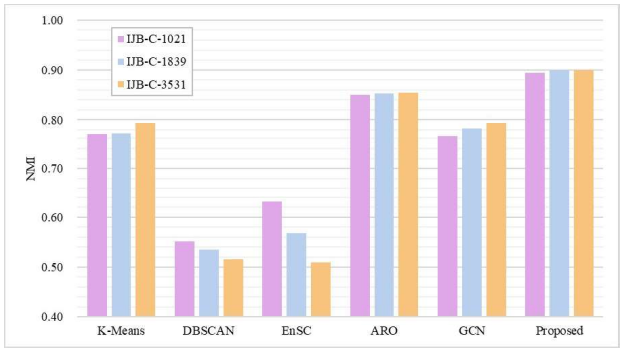

Fig. 10. Comparison results on IJB-C dataset

As shown in Table VII, on the subset that contains 512 subjects, the NMI improvement of the proposed approach is 13.59% against K-Means, 34.94% against DBSCAN, 20.90% against EnSC, 7.12% against Spectral, 0.12% against AP, 2.53% against ARO, and 9.58% against GCN. On the subset that contains 1024 subjects, the NMI improvement of the proposed approach is 18.82% against K-Means, 35.43% against DBSCAN, 26.58% against EnSC, 7.46% against Spectral, 0.26% against AP, 3.54% against ARO, and 8.96% against GCN. On the subset that contains 1845 subjects, the NMI improvement of the proposed approach is 23.29% against K-Means, 36.01% against DBSCAN, 31.49% against EnSC, 8.71% against Spectral, 0.31% against AP, 3.42% against ARO, and 8.68% against GCN.

As shown in Table VIII, similar results are achieved. For example, on the subset that contains 1021 subjects, the NMI improvement of the proposed approach is 12.60% against K-Means, 34.32% against DBSCAN, 26.25% against EnSC, 4.40% against ARO, and 12.93% against GCN.

There are three reasons for the better performance of the proposed approach.

First of all, the proposed method is equipped with the NFA and NFE models to extract more discriminative features. With the NFA model, the important channels can get more responses while with the NFE model, the features can be more discriminative. Then, the features from the second residual block to the fourth residual block are concatenated with both detail and semantic information. Secondly, we use F-Graphs to represent correlation of samples. With the adjacency aggregation process of RE-GCNs, the node similarities in F-Graphs could be inferred. Finally, the feature learning and graph similarity inference process are unified in an end-to-end multi-task learning framework. Therefore, the features could be adjusted according to the feedback of graph similarity inference with back-propagation mechanism of network error.

V. Conclusion

In this paper, we propose a new method of hierarchical deep multi-task learning with attention mechanism for similarity learning. Firstly, F-Graphs are constructed to consider the abundant and underlying similarity relationships of data. Then, a shared HA-Net is designed to extract the hierarchical feature embeddings for similarity learning and classification tasks. Within it, attention mechanism and NFA model are added to gradually enhance the salient feature channels. In addition, we use NFE model to further capture discriminative common and differential information in these channels. In the main task, the RE-GCNs are developed to perform strong similarity inference by strengthening the channels with important relations and adjacent aggregation. For evaluation, we conduct extensive experiments for person re-identification and face clustering applications. The experimental results on four datasets clearly show that the proposed network is competitive over state-of-the-art methods. For future work, an effective way of exploiting rich structural information can be explored. For example, we can mix feature embeddings of neighbors at various distances. In this way, the receptive field of GCNs will become wider, which may be beneficial to similarity learning.

References

[1] S. Ibrahimi, N. van Noord, Z. Geradts, and M. Worring, "Deep metric learning for cross-domain fashion instance retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 3165–3168.

[2] X. Yang, P. Zhou, and M. Wang, "Person reidentification via structural deep metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 2987–2998, 2019.

[3] H. Jain, G. Harit, and A. Sharma, "Action quality assessment using siamese network-based deep metric learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. [Online]. Available: https://doi.org/10.1109/TCSVT.2020.3017727

[4] J. Lu, J. Hu, and Y. Tan, "Discriminative deep metric learning for face and kinship verification," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4269–4282, 2017.

[5] K. Li, Y. Kong, and Y. Fu, "Visual object tracking via multi-stream deep similarity learning networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 3311–3320, 2020.

[6] W. Liao, M. Y. Yang, N. Zhan, and B. Rosenhahn, "Triplet-based deep similarity learning for person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 385–393.

[7] H. Manack and T. L. Van Zyl, "Deep similarity learning for soccer team ranking," in *Proceedings of the IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020, pp. 1–7.

[8] X. Yao, D. She, H. Zhang, J. Yang, M. Cheng, and L. Wang, "Adaptive deep metric learning for affective image retrieval and classification," *IEEE Transactions on Multimedia*, 2020. [Online]. Available: https://doi.org/10.1109/TMM.2020.3001527

[9] H. Zhi, H. Yu, S. Li, and C. Gao, "Dmmln: A deep multi-task and metric learning based network for video classification," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–7.

[10] K. Karaman, E. Gundogdu, A. Koç, and A. A. Alatan, "Quadruplet selection methods for deep embedding learning," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 3452–3456.

[11] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1114–1123.

[12] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 499–515.

[13] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 5, pp. 2811–2821, 2018.

[14] S. Tian, X. Liu, M. Liu, S. Li, and B. Yin, "Siamese tracking network with informative enhanced loss," *IEEE Transactions on Multimedia*, vol. 23, pp. 120–132, 2021.

[15] Q. Suo, W. Zhong, F. Ma, Y. Ye, M. Huai, and A. Zhang, "Multi-task sparse metric learning for monitoring patient similarity progression," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 477–486.

[16] Y. Gao, Y. Li, B. Dong, Y. Lin, and L. Khan, "Sim: Open-world multi-task stream classifier with integral similarity metrics," in *Proceedings of IEEE International Conference on Big Data (Big Data)*, 2019, pp. 751–760.

[17] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3d object retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 1945–1954.

[18] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: Learning to retrieve badly drawn bunnies," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016. [Online]. Available: https://doi.org/10.1145/2897824.2925954

[19] G. Tu, Y. Fu, B. Li, J. Gao, Y. Jiang, and X. Xue, "A multi-task neural approach for emotion attribution, classification, and summarization," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 148–159, 2020.

[20] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task cnn model for attribute prediction," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1949–1959, 2015.

[21] X. Xia, R. Togneri, F. Sohel, Y. Zhao, and D. Huang, "Multi-task learning for acoustic event detection using event and frame position information," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 569–578, 2020.

[22] Y. Yan, Q. Zhang, B. Ni, W. Zhang, M. Xu, and X. Yang, "Learning context graph for person search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2153–2162.

[23] H. Shi, Y. Zhang, Z. Zhang, N. Ma, X. Zhao, Y. Gao, and J. Sun, "Hypergraph-induced convolutional networks for visual classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 2963–2972, 2019.

[24] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv preprint arXiv:1703.07737*, 2017. [Online]. Available: https://arxiv.org/abs/1703.07737

[25] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1335–1344.

[26] I. Elezi, S. Vascon, A. Torcinovich, M. Pelillo, and L. Leal-Taixé, "The group loss for deep metric learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 277–294.

[27] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson, "Ranked list loss for deep metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 5207–5216.

[28] O. Seddati, S. Dupont, and S. Mahmoudi, "Quadruplet networks for sketch-based image retrieval," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 2017, pp. 184–191.

[29] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4766–4779, 2015.

[30] J. Liang, Q. Hu, W. Wang, and Y. Han, "Semisupervised online multikernel similarity learning for image retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 5, pp. 1077–1089, 2017.

[31] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[32] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 527–11 535.

[33] H. Choi, A. Som, and P. Turaga, "Amc-loss: Angular margin contrastive loss for improved explainability in image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 3659–3666.

[34] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5620–5629.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[36] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124.

[37] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3754–3762.

[38] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014. [Online]. Available: http://arxiv.org/abs/1411.7923

[39] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother, "Iarpa janus benchmark-b face dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017, pp. 90–98.

[40] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother, "Iarpa janus benchmark - c: Face dataset and protocol," in *Proceedings of International Conference on Biometrics (ICB)*, 2018, pp. 158–165.

[41] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 371–381.

[42] Z. Zhuang, L. Wei, L. Xie, T. Zhang, H. Zhang, H. Wu, H. Ai, and Q. Tian, "Rethinking the distribution gap of person re-identification with camera-based batch normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 140–157.

[43] C. Shen, G. Qi, R. Jiang, Z. Jin, H. Yong, Y. Chen, and X. Hua, "Sharp attention network via adaptive sampling for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3016–3027, 2019.

[44] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018, pp. 480–496.

[45] X. Qian, Y. Fu, T. Xiang, Y. G. Jiang, and X. Xue, "Leader-based multi-scale attention deep architecture for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 371–385, 2020.

[46] L. Zhao, X. Li, Y. Zhuang, and J. Wang, "Deeply-learned part-aligned representations for person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3219–3228.

[47] H. Sun, Z. Chen, S. Yan, and L. Xu, "Mvp matching: A maximum-value perfect matching for mining hard samples, with application to person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6737–6747.

[48] F. Zheng, C. Deng, X. Sun, X. Jiang, X. Guo, Z. Yu, F. Huang, and R. Ji, "Pyramidal person re-identification via multi-loss dynamic training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8514–8522.

[49] Y. Sun, L. Zheng, W. Deng, and S. Wang, "Svdnet for pedestrian retrieval," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3820–3828.

[50] X. Yang, P. Zhou, and M. Wang, "Person reidentification via structural deep metric learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 2987–2998, 2019.

[51] Y. Shen, H. Li, T. Xiao, S. Yi, D. Chen, and X. Wang, "Deep group-shuffling random walk for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 2265–2274.

[52] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, "Person re-identification with deep similarity-guided graph neural network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018, pp. 486–504.

[53] X. Jiang, F. Yu, Y. Gong, S. Zhao, X. Guo, F. Huang, W.-S. Zheng, and X. Sun, "Devil's in the detail: Graph-based key-point alignment and embedding for person re-id," *arXiv preprint arXiv:2009.05250*, 2020. [Online]. Available: http://arxiv.org/abs/2009.05250v1

[54] J. Guo, Y. Yuan, L. Huang, C. Zhang, J.-G. Yao, and K. Han, "Beyond human parts: Dual part-aligned representations for person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019, pp. 3642–3651.

[55] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 1062–1071.

[56] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose-invariant embedding for deep person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4500–4509, 2019.

[57] L. Wu, R. Hong, Y. Wang, and M. Wang, "Cross-entropy adversarial view adaptation for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2081–2092, 2020.

[58] W. Wang, W. Pei, Q. Cao, S. Liu, G. Lu, and Y. W. Tai, "Push for center learning via orthogonalization and subspace masking for person re-identification," *IEEE Transactions on Image Processing*, vol. 30, pp. 907–920, 2021.

[59] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford InfoLab, Technical Report 2006-13, June 2006. [Online]. Available: http://ilpubs.stanford.edu:8090/778/

[60] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.

[61] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition (CVPR)*, 2016, pp. 3928–3937.

[62] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[63] C. Otto, D. Wang, and A. K. Jain, "Clustering millions of faces by identity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 289–303, 2018.

[64] Z. Wang, L. Zheng, Y. Li, and S. Wang, "Linkage based face clustering via graph convolution network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 1117–1125.