AdvKin: Adversarial Convolutional Network for Kinship Verification

Lei Zhang, *Senior Member, IEEE*, Qingyan Duan, *Student Member, IEEE*, David Zhang, *Fellow, IEEE*, Wei Jia, *Member, IEEE*, and Xizhao Wang, *Fellow, IEEE*

***Abstract*—Kinship verification in the wild is an interesting and challenging problem. The goal of kinship verification is to determine whether a pair of faces are blood relatives or not. Most previous methods for kinship verification can be divided as handcrafted features-based shallow learning methods and convolutional neural network (CNN)-based deep-learning methods. Nevertheless, these methods are still facing the challenging task of recognizing kinship cues from facial images. The reason is that the family ID information and the distribution difference of pairwise kin-faces are rarely considered in kinship verification tasks. To this end, a family ID-based adversarial convolutional network (AdvKin) method focused on discriminative Kin features is proposed for both small-scale and large-scale kinship verification in this article. The merits of this article are four-fold: 1) for kin-relation discovery, a simple yet effective self-adversarial mechanism based on a negative maximum mean discrepancy (NMMD) loss is formulated as attacks in the first fully connected layer; 2) a pairwise contrastive loss and family ID-based softmax loss are jointly formulated in the second and third fully connected layer, respectively, for supervised training; 3) a two-stream network architecture with residual connections is proposed in AdvKin; and 4) for more fine-grained deep kin-feature augmentation, an ensemble of patch-wise AdvKin networks is proposed (E-AdvKin). Extensive experiments on 4 small-scale benchmark KinFace datasets and 1 large-scale families in the wild (FIW) dataset from the first Large-Scale Kinship Recognition Data Challenge, show the superiority of our proposed AdvKin model over other state-of-the-art approaches.**

***Index Terms*—Adversarial loss (AL), convolutional neural networks (CNNs), kinship verification, maximum mean discrepancy (MMD).**

I. Introduction

HUMAN FACES carry abundant individual characteristics, such as identity, age, gender, race, emotion, etc., which can be generally distinguished by looking into the facial images. Face verification that aims to verify whether the two facial images belong to the same person [1] has been overstudied in the computer vision community. Generally, the purpose of kinship verification is to recognize whether the two people are from the same family or have some blood relation. However, discovering the facial kinship relations (i.e., kinship verification) of two given faces is more challenging and understudied. Kinship verification has encountered many challenging applications, such as the human social relations exploration, social-media analysis, crime scene investigations, and missing children searches, etc., [2]–[5]. Human face inspired visual perception is an intuitive approach for kinship similarity computation, because the appearance of members from the same family shows a more similar visual perception than those without blood relation. To this end, kinship verification in unconstrained conditions has received more attention in recent years. A study on four typical parent–child relations, such as Father–Daughter (F-D), Father–Son (F-S), Mother–Daughter (M-D), and Mother–Son (M-S), has achieved great progress. Four small-scale benchmarks (i.e., 4K in total), including KinFaceW-I [2], KinFaceW-II [2], Cornell KinFace [6], and UB KinFace [7] have been developed. Some facial image pairs with/without kinship are shown in Fig. 1, from which the difficulty for discovering implicit kin-relation is clearly shown. Besides, some kinship databases like WVU [8], families in the wild (FIW) [9], and UvA-NEMO [10] were also proposed, in which FIW is the largest kinship dataset (over 1 million) of seven kin-relations [11], including four conventional parent–child relations and three new sibling relations [i.e., Sister–Brother (SIBS), Brother–Brother (B-B), and Sister–Sister (S-S)]. Visually, Fig. 2 shows the pairwise faces for each kin-relation in FIW. In this article, both the small-scale and large-scale kinship verification tasks are explored.

Due to the various factors in unconstrained faces, such as pose, illumination, expression, background clutters, etc.,

Manuscript received October 10, 2019; accepted December 10, 2019. This work was supported in part by the National Science Fund of China under Grant 61771079, in part by Chongqing Youth Talent Program, in part by the NSFC under Grant 61732011 and in part by the Fundamental Research Funds of Chongqing under Grant cstc2018jcyjAX0250. This article was recommended by Associate Editor H. Qiao. *(Corresponding author: Lei Zhang.)*

Lei Zhang and Qingyan Duan are with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China (e-mail: leizhang@cqu.edu.cn; qyduan@cqu.edu.cn).

David Zhang is with the School of Science and Engineering, Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China (e-mail: csdzhang@comp.polyu.edu.hk).

Wei Jia is with the School of Computer and Information, Hefei University of Technology, Hefei 230061, China (e-mail: china.jiawei@139.com).

Xizhao Wang is with the College of Computer Science and Software Engineering and Guangdong Key Lab of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, China (e-mail: xizhaowang@ieee.org).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2959403

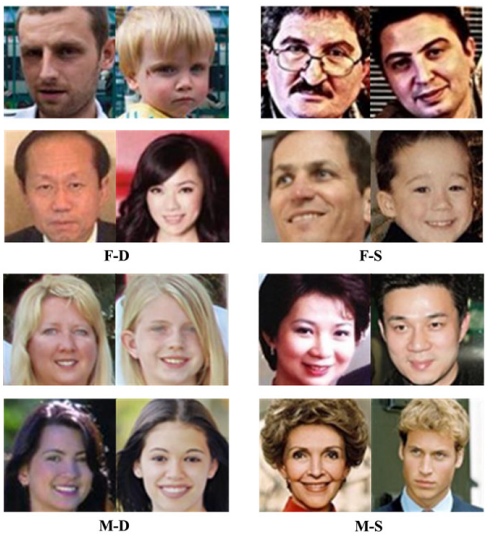

Fig. 1. Some positive (with kinship relation) and negative pairs (without kinship relation) from KinFaceW-I, KinFaceW-II, Cornell KinFace, and UB KinFace, respectively. The odd rows are positive pairs and the even rows are negative pairs.

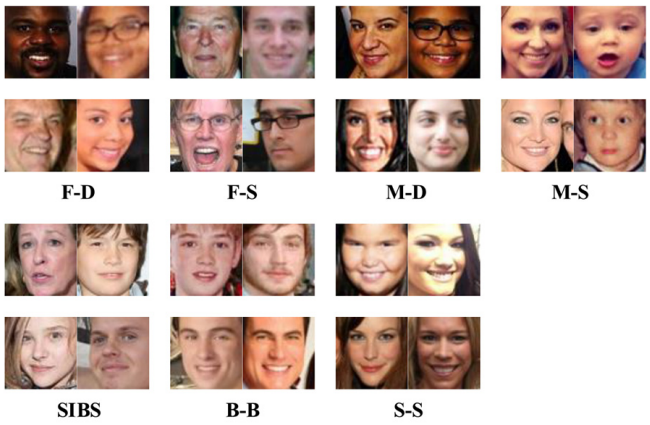

Fig. 2. Some positive (with kinship relation) and negative pairs (without kinship relation) from the FIW dataset. The odd rows are positive pairs and the even rows are negative pairs.

kinship verification is still a challenging and unsolved topic. Different from face recognition-based discriminative feature representation, the kin-relation feature is implicit and hard to discover. Although there are many algorithms proposed for kinship verification, most of these works follow a similar technical routine that large-margin discriminative metrics are learned based on handcrafted features, for example, local binary patterns (LBPs), histogram of oriented gradient (HOG), etc. A representative work can be referred to as [2], in which a neighborhood repulsed metric learning (NRML) was proposed and achieved excellent verification performance. However, these kinship verification algorithms [12]–[14] focus on the learning of distance metrics, since those off-the-shelf low-level descriptors cannot well find the implicit kin-relation specific features. As a result, the implicit and abstract kinship relation features cannot be adequately represented [15], and the kinship verification performance is restricted.

Deep learning, proposed by Hinton and Salakhutdinov [16] and Lecun *et al.* [17], is the most popular machine learning algorithm for discovering discriminative middle-level and high-level representations in a hierarchical manner [18]. Recently, a hierarchical kinship verification was proposed based on the DBN method [8]. In particular, convolutional neural networks (CNNs) have recently achieved a great success in various computer vision tasks, such as face recognition [1], [19], [20], object recognition [21]–[24], etc. Also, CNN has been used for kinship verification [15], [25]–[27]. Although these works greatly promote kinship verification, they adopted a conventional CNN architecture with a single loss function, such as softmax loss (SL) or triplet loss, to train the network from scratch, by adopting face verification-based similar strategy to solve kinship verification problem. In addition, for training CNNs, a large number of kinship data is very necessary. From the viewpoint of data augmentation, generative adversarial net (GAN) [28] can be used for generating photo-realistic examples through adversarial learning. However, due to the data scarcity of labeled Kinship faces, training an effective GAN is very difficult. Therefore, it is not very appropriate to introduce GAN into kinship verification directly.

Motivation: For face recognition/verification task, the general idea is to construct the different classes or doublet/triplet pairs [1], [20], then minimize the variance of intra-class/positive pairs from the same individual and maximize the variance of interclass/negative pairs from different individuals, such that high similarity can be preserved for positive image pairs. However, different from face recognition/verification, a significant feature distribution difference between pairwise faces across generation exists in kinship verification. Consequently, the kin-faces cannot be well interpreted by using a conventional deep model. Undoubtedly, discovering the implicit kinship specific feature is more challenging than the identity specific feature. Therefore, learning kin-related features with deep networks becomes a challenge.

Idea: In this article, inspired by maximum mean discrepancy (MMD) [29] and GAN [28], a novel adversarial loss (AL) is proposed to interpret the distribution difference between pairwise faces. Specifically, the proposed AL is imposed in the first fully connected layer, which tends to minimize the interclass discrepancy and maximize the intraclass discrepancy based on the proposed negative MMD (NMMD). On the contrary, a contrastive loss (CL) is formulated to maximize the interclass distance and minimize the intraclass distance in the second fully connected layer. Naturally, the adversarial process between the AL and the CL in the two fully connected layers is tailored to promoting the discrimination of feature representation by introducing self-attacks in the network. For fully exploiting the family ID (class label), an SL can be further formulated for improving the recognition performance on the large-scale kinship verification task. The proposed AdvKin model with a two-stream shared deep network is described in Fig. 3, from which we observe that the loss model is imposed on the shared fully connected layers. It is worth noting that the residual structure and SL described in dashed lines are used for large-scale kinship verification. For further augmenting the

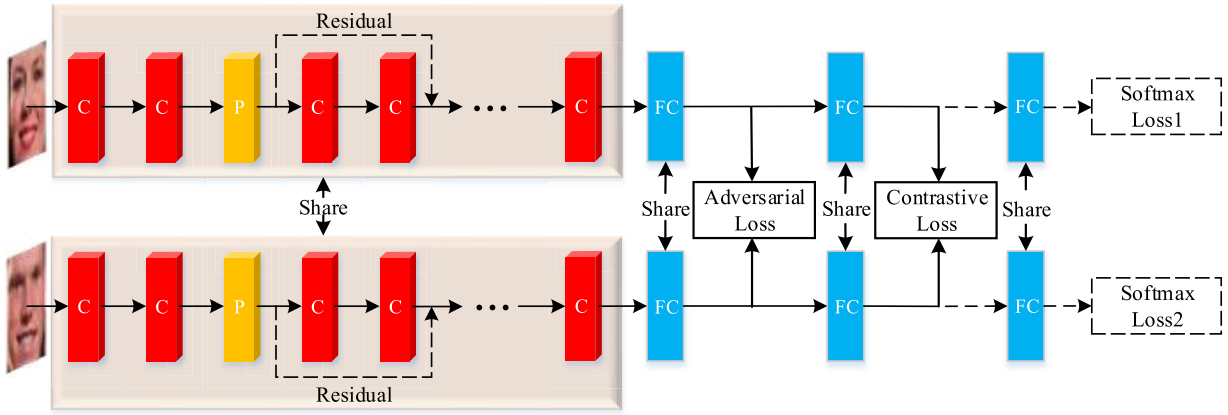

Fig. 3. Pipeline of our proposed two-stream shared AdvKin approach. *C* denotes convolution layer, *P* denotes pooling layer, and *FC* denotes fully connected layer. Note that the parts (i.e., residual connection versus SL layer) indicated by dashed lines are specifically added for large-scale kinship verification tasks.

kin-related features, two ensembles of the AdvKin network (E-AdvKin) are proposed.

This article is an extended version of our conference work [30], [31] in model formulation, optimization algorithms, experiments, and model analysis, such that the proposed model is more interpretable, discriminative, and competitive. The contributions of this article are summarized as follows.

1) In this article, a novel two-stream adversarial convolutional network (AdvKin) model is proposed for both small-scale and large-scale kinship verification, which exploits a self-adversarial strategy and CL in the fully connected layers for feature distribution discrepancy reduction and discriminative feature representation.
2) The simple yet effective self-adversarial mechanism is formulated by designing an NMMD-based AL in the first fully connected layer. It can be used to impose learning difficulty on the convolutional network to improve the robustness of kin-relation features by minimizing the interclass distribution discrepancy and maximizing the intraclass discrepancy, simultaneously.
3) In order to eventually decrease the intraclass discrepancy (positive pairs) while increasing the interclass discrepancy (negative pairs), the proposed AL is combined with the $\mathcal{L}_2$-distance-based CL to achieve the adversarial process. In addition, for large-scale kinship verification, the family ID-based SL is formulated with a deeper residual structure.
4) For better discovering the implicit kin-related feature representation, an E-AdvKin models is naturally proposed for deep feature augmentation. Specifically, we adopt two types of feature augmentation methods. Specifically, for the small-scale kinship verification task, in order to increase the data, a patch-wise feature augmentation that concatenates the deep features of multiple overlapped facial regions (patches) is considered. For large-scale kinship verification task, because of the richness of kinship data, the deep feature concatenation from multiple deep networks is proposed.

II. Related Work

A. Shallow Kinship Verification

In recent years, a number of shallow models and algorithms for kinship verification have been proposed, which can be divided into two categories: 1) low-level feature-based approaches [2], [6] and 2) model-based metric-learning approaches [32], [33]. For the former, existing feature descriptors include HOG [6], [33], [34], scale-invariant feature transform (SIFT) [2], and LBP [2]. A discriminative compact binary face descriptor (D-CBFD) from a set of weakly labeled samples for kinship verification was proposed in [35]. These methods tend to use low-level facial features or their combination for kinship verification. For the latter, a simple yet discriminative metric is required for distinguishing whether two face images are with kinship relation or not. The representative work can be referred to as NRML proposed by Lu *et al.* [2]; prototype-based discriminative feature learning (PDFL) proposed by Yan *et al.* [32]; transfer subspace learning (TSL) [7], [36]; support vector machine (SVM) [32], discriminative multimetric learning [12], [37]; large-margin multimetric learning (LM^{3}L) [14]; ensemble similarity learning (ESL) [33]; deep kinship verification (DKV) that integrates excellent deep-learning architecture into metric learning [25]; and multiple kernel similarity metric learning [13]. Although these previous works have achieved a great progress on the challenging kinship verification, the problem is that the low-level features are general representation of faces without better exploiting the structural kinship characteristic.

B. Deep Kinship Verification

CNN [16], as an end-to-end supervised deep-learning methods from image pixels to high-level semantics, has shown a huge success in face recognition [1], [19], [20], [38], [39]. The features from the bottom layer to the top layer in the network can be identified as hierarchical image representation from low level and high level. There are several popular CNN models. VGG-Face [40] was pretrained on large-scale faces with the VGG network and shows state-of-the-art face

verification performance. ResNet [21] adopts the short connection to improve the performance of object recognition. Multitask CNN (MTCNN) [41] used the candidate CNNs to detect facial landmarks. FaceNet [1] constructed a triplet-loss model to improve face verification accuracy. The center-loss model proposed in [19] aims to learn within-class separable features. Angular softmax (A-Softmax) loss was proposed in SphereFace [20] to learn angularly discriminative features for face recognition.

Recently, CNN has also emerged in kinship verification. For example, SMCNN proposed by Li *et al.* [15] achieved the kin-relation verification through two identical CNNs supervised by similarity metric-based loss function. The CNN-points method proposed by Zhang *et al.* [26] employed ten facial regions to learn a group of CNNs for kinship verification. Also, a Siamese-like coupled convolutional encoder–decoder network was proposed for kinship verification [42]. Since the faces from the same photograph are more likely to be from the same family, so in [27] a CNN classifier was trained to determine whether the two faces are from the same photograph or not. Weighted graph embedding-based metric-learning (WGEML) [43] framework jointly learns multiple metrics from multiple handcrafted features and CNN features by constructing an intrinsic graph and two penalty graphs to characterize the intraclass compactness and interclass separability for each feature representation, respectively. Then, both the consistency and complementarity among multiple features can be fully exploited. Although these approaches have achieved surprisingly good performance, the progress is still insufficient and the deep convolutional network is also understudied due to the data scarcity.

C. Generative Adversarial Network

GAN [28] has been widely used in computer vision issues, such as image generation [44], image super resolution [45], and text to image synthesis [46]. Several popular modifications of GAN are proposed in different scenarios, such as semisupervised GAN (SSGAN) [47], deep convolutional GAN (DCGAN) [48], CycleGAN [49] for style transfer learning, and disentangled representation learning GAN (DRGAN) [50] for pose-invariant face recognition. Essentially, the success of GAN lies in this adversarial learning mechanism with min–max loss-based adversarial optimization.

However, the effective training of GAN mainly depends on abundant annotated examples and tricks, which does not hold in the small-scale kinship verification task. In this article, motivated by the adversarial learning mechanism in GAN, for improving the discrimination of kinship feature representation, a simple yet effective self-adversarial idea is proposed. Notably, this article is essentially different from GAN that our objective is not for generating images, but for general discriminative feature learning and kinship verification.

D. Differences From the SMCNN and CNN-Points

The proposed AdvKin model is closely related but essentially different from SMCNN [15] and CNN-points [26], which are the two representative works in kinship verification using the CNN model. In SMCNN, a similarity metric loss was proposed for general network training. In CNN-points, a one-stream ensemble network of ten patches was trained supervised by a binary SL function.

Specifically, the differences and advantages between the proposed AdvKin model and both SMCNN and CNN-points are fourfold.

1) A simple yet effective AL is proposed as attacks of the first fully connected layer, which improves the learning capability of the proposed CL in the second fully connected layer by the adversarial learning mechanism.
2) The proposed AdvKin is a two-stream and flexible convolutional network by introducing a residual structure and a family ID-based SL.
3) From the viewpoint of data augmentation and model augmentation, two kinds of ensemble strategies have been proposed by considering patch level fusion and network level fusion.
4) We have experimented on both small-scale and large-scale kinship verification tasks on almost all the available kinship datasets for the comprehensive evaluation of the proposed model.

III. Proposed AdvKin Model

The proposed AdvKin method is established with a two-stream network architecture. The basic idea of the proposed AdvKin model is shown in Fig. 4. It is clear that we tend to learn discriminative kin-relation features by self-adversarial learning between the adversarial and the contrastive layer.

A. Mathematical Notations

Let $\mathbf{x}_n^1$ and $\mathbf{x}_n^2$ denote the feature vector of the nth kinship image pair (I_n^1, I_n^2), respectively. N denotes the batch size. $d = ||\mathbf{x}_n^1 - \mathbf{x}_n^2||_2$ is the $\mathcal{L}_2$-distance between $\mathbf{x}_n^1$ and $\mathbf{x}_n^2$. $\delta(\cdot)$ is an indicator function and $\delta(\text{condition}) = 1$ if the condition is satisfied, otherwise $\delta(\text{condition}) = 0$. y_n^1 and y_n^2 are the family IDs of the input kinship pairs $\mathbf{x}_n^1$ and $\mathbf{x}_n^2$, respectively. Let $\hbar$ be the reproducing kernel Hilbert space (RKHS). Given two distributions s and t, and mapped to a RKHS by using an implicit function $\phi(\cdot)$. $E_{\mathbf{x}^s \sim s}[\phi(\cdot)]$ denotes the expectation with respect to the distribution s, and $||\phi||_\hbar \leq 1$ defines a set of functions in the unit ball of RKHS $\hbar$.

B. Family ID-Based Contrastive Loss

FIW is by far the largest and the most comprehensive kinship dataset available in computer vision and multimedia communities. Different from the previous four small-scale kinship datasets, that has only pairwise kinship mode (e.g., KinFaceW-I), FIW also provides the family tree to reflect the real data distribution of a family and their members. In order to improve the performance of our method, the family ID is also used in our model to obtain more discriminative deep features, such that the kin-relation can be better interpreted. Nevertheless, it is worth noting that the existing small-scale kinship datasets have no family information. Therefore, we select the positive pairs of parent–child images and manually mark each positive pair as different family ID (label) starting

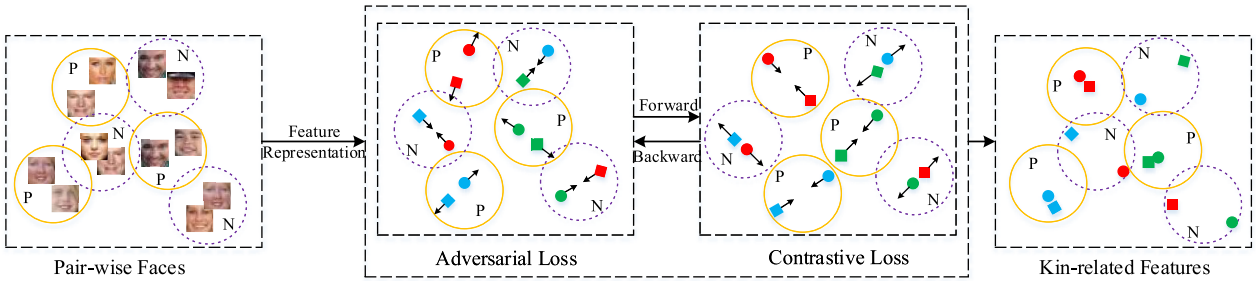

Fig. 4. Basic idea of our AdvKin. It describes the adversarial process between AL and CL. The square and round denote a pair of faces. In CL, the pairwise data points of the same color in the solid circles represent positive pairs (P), therefore, these points are attracted each other. Also, the pairwise data points of different colors in the dashed circles represent negative pairs (N), so they are repulsed each other. In the AL, the positive pairs are formulated to repulse each other, while the negative pairs are attracted each other. After adversarial learning, discrimination is desired.

from 0. That is, for each positive *parent-child* pair, they are marked as the same family ID. Note that only the faces with blood relation can share the same family ID and be constructed as positives.

In the two-stream network, the CL acts as a supervisory signal. For kinship verification tasks, the family IDs have been provided, which means that the pair of kin-relation samples must have the same family ID. In order to verify the kinship relation by integrating the family ID information, the CL is presented as follows:

$$\min\ L_C = \frac{1}{2N}\sum_{n=1}^{N}\Big(\delta\left(y_n^1 = y_n^2\right)d^2 + \delta\left(y_n^1 \neq y_n^2\right)\max(\text{margin} - d, 0)^2\Big) \quad (1)$$

where margin is an adjustable parameter used to control the maximum distance of negative pair.

Generally, the CL is formulated by pulling the positive pairs as close as possible, while repulsing the negative pairs as far as possible, simultaneously. However, the distribution discrepancy of pairwise kinship faces from different sources is rarely considered. To this end, an AL layer is formulated as well as the CL, such that a more generalized network can be trained by imposing attacks before the CL layer.

C. Family ID-Based Adversarial Loss

MMD is a straight-forward test statistic to quantize the distribution difference between domain feature embedding, which is usually employed to reduce the domain bias and shift in transfer learning community [29], [51]–[54]. The MMD between s and t is then defined as [55]

$$\text{MMD}^2(s,t) = \sup_{||\phi||_{\hbar}\leq 1}\left\|E_{\mathbf{x}^s\sim s}\left[\phi\left(\mathbf{x}^s\right)\right] - E_{\mathbf{x}^t\sim t}\left[\phi\left(\mathbf{x}^t\right)\right]\right\|_{\hbar}^2. \quad (2)$$

The most important property is that we have $\text{MMD}(s,t) = 0$ if and only if $s = t$. Inspired by MMD, the distribution difference can be reduced by minimizing the discrepancy between pairwise kin-faces. Therefore, an MMD-based pairwise loss is formulated with a general idea that it should minimize the intraclass variations (kin face pair) while keeping the interclass features separable (nonkin face pair). Specifically, the MMD-based pairwise loss is formulated as

$$\min\ L_{\text{MMD}} = \frac{1}{2N}\sum_{n=1}^{N}\Big(\delta\left(y_n^1 = y_n^2\right)\left\|\phi\left(\mathbf{x}_n^1\right) - \phi\left(\mathbf{x}_n^2\right)\right\|_{\hbar}^2 - \delta\left(y_n^1 \neq y_n^2\right)\left\|\phi\left(\mathbf{x}_n^1\right) - \phi\left(\mathbf{x}_n^2\right)\right\|_{\hbar}^2\Big). \quad (3)$$

It can be seen that the MMD-based pairwise loss is a straightforward method to decrease the distribution difference across different kinship domains. Besides, some indirect approaches can be used to strengthen the network. For example, CNN training can be improved by introducing additive noise. Also, as GAN [28] does, the generative model tends to generate the data that cannot be distinguished from the real data, while the discriminative model contributes to distinguish the generated data from real data as much as possible. Although the objectives of the generative model and discriminative model are exactly reverse, the generation performance is promoted due to the adversarial learning mechanism. Inspired by the adversarial characteristic of GAN, in order to further improve the discrimination of deep kin-relation features, a self-adversarial learning mechanism is formulated by proposing a NMMD-based AL as follows:

$$\min\ L_A = -\frac{1}{2N}\sum_{n=1}^{N}\Big(\delta\left(y_n^1 = y_n^2\right)\left\|\phi\left(\mathbf{x}_n^1\right) - \phi\left(\mathbf{x}_n^2\right)\right\|_{\hbar}^2 - \delta\left(y_n^1 \neq y_n^2\right)\left\|\phi\left(\mathbf{x}_n^1\right) - \phi\left(\mathbf{x}_n^2\right)\right\|_{\hbar}^2\Big). \quad (4)$$

By comparing (4) with (3), the only difference is the minus sign. It means that the NMMD-based AL plays an opposite role as the MMD-based pairwise loss does. For the network deployment, the AL is added on the first fully connected layer, so that the adversarial process can be formulated with the CL in the second fully connected layer. Therefore, the AdvKin model can be trained by combining the AL together with the CL, such that more discriminative features can be learned. Specifically, the objective function of our

AdvKin is

$$\begin{aligned} L &= L_C + \lambda L_A \\ &= \frac{1}{2N}\sum_{n=1}^{N}\Big(\delta\Big(y_n^1 = y_n^2\Big)d^2 + \delta\Big(y_n^1 \neq y_n^2\Big) \\ &\qquad \times \max(\text{margin} - d, 0)^2\Big) \\ &\quad - \lambda\Bigg(\frac{1}{2N}\sum_{n=1}^{N}\Big(\delta\Big(y_n^1 = y_n^2\Big) - \delta\Big(y_n^1 \neq y_n^2\Big)\Big) \\ &\qquad \times \left\|\phi\Big(\mathbf{x}_n^1\Big) - \phi\Big(\mathbf{x}_n^2\Big)\right\|_{\hbar}^2\Bigg) \end{aligned} \tag{5}$$

where λ is a scalar coefficient used for tradeoff between the two losses. The CL can be considered as a special case of this joint supervision, when λ is set to 0. The AL works as an attack on the convolutional network by minimizing the interclass distribution discrepancy and maximizing the intraclass discrepancy in first fully connected layer, simultaneously. But the CL is formulated to maximize the interclass distance and simultaneously minimize the intraclass distance in the second fully connected layer for feature discrimination and convergence. Through the game between the AL and the CL, the discrimination of the deep feature layer can be further improved, as the basic idea of AdvKin describes in Fig. 4.

As shown in Fig. 4, the proposed AdvKin benefits from the self-adversarial mechanism between the NMMD-based AL and the CL. The AL is imposed in the first fully connected layer to minimize the interclass discrepancy and maximize the intraclass discrepancy in RKHS. Essentially, the model is improved by increasing the difficulty of training. That is, by automatically generating "hard features" in the AL layer, that is, the similar pairs are repulsed and the dissimilar pairs are attracted in feature space, then the CL layer can be learned more carefully for aligning these hard features. With back-propagation optimization between the AL layer and CL layer, the performance of AdvKin can be progressively boosted.

In (5), $\phi(\cdot)$ denotes the implicit feature map function, which can be induced by using the kernel function $k(\mathbf{x}_n^1, \mathbf{x}_n^2) = \langle\phi(\mathbf{x}_n^1), \phi(\mathbf{x}_n^2)\rangle$. Thus, the (4) can be rewritten as

$$\begin{aligned} L_A = \frac{1}{2N}\sum_{n=1}^{N}&\Big(\delta\Big(y_n^1 \neq y_n^2\Big) - \delta\Big(y_n^1 = y_n^2\Big)\Big) \\ &\times \Big(k\Big(\mathbf{x}_n^1, \mathbf{x}_n^1\Big) + k\Big(\mathbf{x}_n^2, \mathbf{x}_n^2\Big) - 2k\Big(\mathbf{x}_n^1, \mathbf{x}_n^2\Big)\Big) \end{aligned} \tag{6}$$

where k denotes the Gaussian kernel function with bandwidth (kernel parameter) σ^2. Then, (6) can be rewritten as

$$\begin{aligned} L_A = \frac{1}{N}\sum_{n=1}^{N}&\Big(\delta\Big(y_n^1 \neq y_n^2\Big) - \delta\Big(y_n^1 = y_n^2\Big)\Big) \\ &\times \left(1 - \exp\left(-\frac{\left\|\mathbf{x}_n^1 - \mathbf{x}_n^2\right\|_2^2}{2\sigma^2}\right)\right). \end{aligned} \tag{7}$$

In the training stage of CNNs, the back-propagation algorithm is deployed to update the parameters of AdvKin network. Mini-batch stochastic gradient descent (SGD) is one of the most commonly used back-propagation algorithms. For optimization, the gradients (derivatives) of the AL function L_A with respect to $\mathbf{x}_n^1$ and $\mathbf{x}_n^2$ can be computed as

$$\begin{aligned} \frac{\partial L_{\mathrm{A}}}{\partial \mathbf{x}_n^1} = &\frac{1}{N\sigma^2}\Big(\delta\Big(y_n^1 \neq y_n^2\Big) - \delta\Big(y_n^1 = y_n^2\Big)\Big) \\ &\times \exp\left(-\frac{\left\|\mathbf{x}_n^1 - \mathbf{x}_n^2\right\|_2^2}{2\sigma^2}\right)\Big(\mathbf{x}_n^1 - \mathbf{x}_n^2\Big) \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial L_{\mathrm{A}}}{\partial \mathbf{x}_n^2} = &\frac{1}{N\sigma^2}\Big(\delta\Big(y_n^1 \neq y_n^2\Big) - \delta\Big(y_n^1 = y_n^2\Big)\Big) \\ &\times \exp\left(-\frac{\left\|\mathbf{x}_n^1 - \mathbf{x}_n^2\right\|_2^2}{2\sigma^2}\right)\Big(\mathbf{x}_n^2 - \mathbf{x}_n^1\Big). \end{aligned} \tag{9}$$

D. Family ID-Based Joint Loss With Softmax

Different from the small-scale kinship verification, in terms of the training protocol of the large-scale kinship dataset, the family ID for each kin-face is provided in large-scale FIW data which contains 300 families. Therefore, it is reasonable to exploit the general supervisory signal (i.e., family ID) by integrating a two-stream SL into the AdvKin model.

Different from the CL and the AL, the SL aims to improve the family class separability of deep features. With this motivation, SL is also integrated into our AdvKin to further discover the implicit kin-relation of deep features. Considering the pairwise structure of the two-stream network architecture, two SL functions can be formulated for each branch. Specifically, the joint loss is formulated as

$$L = L_C + \lambda L_A + L_{S1} + L_{S2} \tag{10}$$

where L_{S1} and L_{S2} denote the SL (cross entropy) for $\mathbf{x}_n^1$ and $\mathbf{x}_n^2$, respectively. L_C and L_A have been presented in (1) and (4), respectively.

In the network, a new output layer (i.e., softmax layer) with 300 neurons (i.e., 300 families) is added after the CL layer, as shown in Fig. 3 indicated by dashed lines.

IV. Proposed Ensemble of AdvKin

Consider that the performance of the model can be improved by feature augmentation and fusion [26], [56], two slightly different E-AdvKin are proposed.

A. E-AdvKin for Small-Scale Kinship Verification

The similarity between the two kin-related facial images is presented in some local facial areas, such as eyes, nose, etc., [26]. To this end, the facial patches are exploited to discover the local kin-related feature. The key-points-based patches benefit to kinship analysis, therefore we detect five key points, including the centers of eyes, the corners of mouth, and the tip of nose. Then, each facial image is cropped and aligned as 64×64 around the five key points. The five facial regions (patches) extraction from a raw image is shown in Fig. 5. Since each facial region shows valuable kin-related information, it is reasonable to fuse the knowledge of all patches together for discriminative kin-specific features. To this end, we propose a patch-wise E-AdvKin approach, which is shown in Fig. 6(a) from the viewpoint of data augmentation. As shown

TABLE I
TWO-STREAM ADVKIN NETWORK ARCHITECTURE FOR SMALL-SCALE KINSHIP VERIFICATION TASK

Conv1	Pool1	Conv2	Pool2	Conv3	Pool3	Conv4	FC
conv11-6 conv12-6	max-2	conv21-16 conv22-16	max-2	conv31-30 conv32-30	max-2	conv4-60	FC1-128 FC2-80

TABLE II
FACE INDEX OF THE FIVE FOLDS CROSS-VALIDATION ON SMALL-SCALE KINSHIP DATASETS. THE NUMBER DENOTES THE INDEX

Fold	KinFaceW-I				KinFaceW-II	UB	Cor
	F-S	F-D	M-S	M-D	all subset	all subset	all subset
1	[1,31]	[1,27]	[1,23]	[1,25]	[1,50]	[1,40]	[1,29]
2	[32,62]	[28,54]	[24,46]	[26,50]	[51,100]	[41,80]	[30,58]
3	[63,93]	[55,81]	[47,69]	[51,75]	[101,150]	[81,120]	[59,87]
4	[94,124]	[82,108]	[70,92]	[76,100]	[151, 200]	[121,160]	[88,115]
5	[125,156]	[109,134]	[93,116]	[101,127]	[201,250]	[161, 200]	[116, 143]

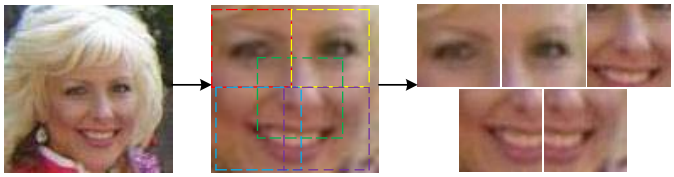

Fig. 5. Five key point facial regions (patches) partition process from the raw image in our E-AdvKin network.

in Fig. 6(a), the new structure contains six AdvKin networks and each of which produces 80-D kin-related deep features. Finally, after concatenation, the total feature dimension is 480 (80×6) for kinship verification.

B. E-AdvKin for Large-Scale Kinship Verification

For large-scale kin-data, the patch-wise feature augmentation has a large computational burden and becomes unsuitable. Because the features are hierarchically distributed throughout the CNN network [57], different features imply different levels of kinship relation. Therefore, networks with different depth are concatenated in the feature level. Furthermore, the extracted deep features from AdvKin networks with different supervisory signals are complementary to some extent. Therefore, from the viewpoint of model augmentation, four networks, including one VGG-Face network [40] and three AdvKin nets with different loss and depth are concatenated, which is described in Fig. 6(b).

V. EXPERIMENTS FOR SMALL-SCALE TASK

A. Description of Network Architecture and Datasets

In the two-stream network (Fig. 3), the parameters of all layers are shared. For small-scale kinship verification tasks, the AdvKin employs a shallow CNN model. Besides, we prefer using smaller convolutional kernel (3×3) instead of a bigger one (5×5), so that the network can be deeper without increasing the number of network parameters. Specifically, the network architecture for a small-scale task is described in Table I and the inputs are pairwise kinship facial images of 64×64.

In experiments, four small-scale kinship benchmarks, such as KinFaceW-I, KinFaceW-II [2], Cornell KinFace [6], and UB KinFace [7], are considered.

1) Both KinFaceW-I and KinFaceW-II include four different types of kin relationships: F-S, F-D, M-S, and M-D. The former consists of 156, 134, 116, and 127 pairs. The latter consists of 250 pairs for each relationship.
2) Cornell KinFace contains totally 150 parent–child pairs.
3) UB KinFace contains 200 triplets and each triplet is structured by a child, young parent, and old parent.

B. Experimental Setup

For the small-scale kinship verification task, the fivefold cross-validation strategy is employed. Therefore, the kin faces of fourfolds include 3162 images of 1500 classes are used for model training. For each kinship database, except the UB KinFace data, two images per class (i.e., family ID) are considered. UB KinFace is different from the other three kinship datasets in that it is constructed in triplet: 1) children; 2) young parents; and 3) old parents. That is, the young parent and the old parent in each triplet are with the same identity but different ages. Therefore, for UB KinFace, three images per family ID are used. The positive and negative kin pairs are with the same and different family ID, respectively. Obviously, the number of negative pairs is much larger than that of the positive pairs. In order to balance the sample, the same number of positives and negatives are selected for training. In evaluation, with fourfolds for training and the remaining onefold for testing, the average accuracy of fivefold is reported. Note that cosine distance is used for kinship verification with a threshold determined via the Validation set. Specifically, the image index set of the four datasets for each fold is shown in Table II.

We compare with ten state-of-the-art methods, including four shallow learning methods, such as MNRML [2], MPDFL [32], ESL [33], and D-CBFD [35], and six deep-learning methods, such as SMCNN [15], DKV [25], CNN-points [26], DDMML [56], FSP [27], and WGEML [43]. In addition, the comparison with human score [32] is also analyzed. Notably, for all algorithms, fivefold cross-validation is used.

In optimization, the mini-batch SGD is used, with an initial learning rate of 10^{-2}. The margin of CL is set as 1. For the

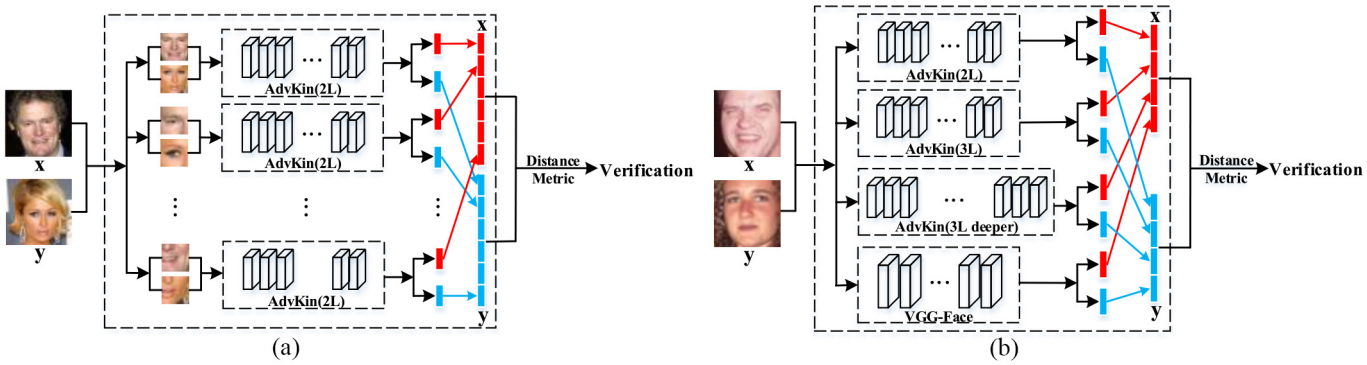

Fig. 6. Structures of the proposed E-AdvKin models for small-scale (a) and large-scale (b) kinship verification tasks, respectively. Note that the AdvKin(2L) model represents AL + CL and the AdvKin(3L) model represents AL + CL + SL.

TABLE III
ACCURACY OF DIFFERENT METHODS ON SMALL-SCALE KINSHIP VERIFICATION

Methods	KinFaceW-I					KinFaceW-II					UB			Cor
	F-S	F-D	M-S	M-D	Mean	F-S	F-D	M-S	M-D	Mean	0-1	0-2	Mean	-
Human A [32]	62.0	60.0	68.0	72.0	65.6	63.0	63.0	71.0	75.0	68.0	-	-	-	-
Human B [32]	68.0	66.5	74.0	75.0	70.9	72.0	72.5	77.0	80.0	75.4	-	-	-	-
MNRML [2]	72.5	66.5	66.2	72.0	69.9	76.9	74.3	77.4	77.6	76.5	67.3	66.8	67.1	71.6
MPDFL [32]	73.5	67.5	66.1	73.1	70.1	77.3	74.7	77.8	78.0	77.0	67.5	67.0	67.3	71.9
ESL (HOG) [33]	83.9	76.0	73.5	81.5	78.6	81.2	73.0	75.6	73.0	75.7	-	-	-	-
D-CBFD [35]	79.6	73.6	76.1	81.5	77.6	79.0	74.2	75.4	77.3	78.5	-	-	-	-
SMCNN [15]	75.0	75.0	68.7	72.2	72.7	75.0	79.0	78.0	85.0	79.3	-	-	-	-
DKV [25]	71.8	62.7	66.4	66.6	66.9	73.4	68.2	71.0	72.8	71.3	-	-	-	-
CNN-Points [26]	76.1	71.8	78.0	84.1	77.5	89.4	81.9	89.9	92.4	88.4	-	-	-	-
DDMML (All) [56]	**86.4**	**79.1**	**81.4**	**87.0**	**83.5**	87.4	83.8	83.2	83.0	84.3	-	-	-	-
FSP [27]	74.6	74.9	78.3	86.0	76.8	**92.3**	84.5	**90.3**	**94.8**	**90.2**	-	-	-	76.7
WGEML [43]	78.5	73.9	80.6	81.9	78.7	88.6	77.4	83.4	81.6	82.8	-	-	-	-
AdvKin	75.7	78.3	77.6	83.1	78.7	88.4	**85.8**	88.0	89.8	88.0	**75.0**	**75.0**	**75.0**	**81.4**
E-AdvKin	76.6	77.3	78.4	86.2	79.6	91.6	85.2	90.2	92.4	89.9	-	-	-	80.4

Note: The best results are highlighted in bold type, and the second-best results are underlined.

small-scale task, the batch size is set as 151 and trained on one NVIDIA 1080Ti GPU for about 13 s.

C. Comparison With Previous Methods

The verification results of the proposed AdvKin and E-AdvKin methods on four benchmark kinship datasets are shown in Table III, from which we observe that as follows.

1) The proposed AdvKin methods consistently outperform state-of-the-art shallow methods deployed with handcrafted feature ensemble and metric learning. The effectiveness of our AdvKin is shown.
2) The proposed AdvKin methods also outperform the deep learning-based face verification methods, such as SMCNN [15], DKV [25], CNN-points [26], DDMML [56], FSP [27], and WGEML [43]. Different from them, our methods focus on an adversarial learning, so that the kin-related feature can be well captured adequately. Note that DDMML as a multilayer perception outperforms ours and other CNN-based methods in KinFaceW-I but worse than others in KinFaceW-II. The reason may be that the number of faces in KinFaceW-I is smaller than KinFaceW-II, and generally, CNN-based deep methods cannot work well on a smaller dataset.
3) The depth of these deep methods, such as SMCNN, CNN-Points, FSP, and WGEML is 5, 5, 11, and 16, respectively. Our method has nine layers that need to be trained. With the architecture of similar depth, the performance of our method is better than others in totally.
4) By comparing our method with human knowledge on the KinFaceW-I and KinFaceW-II, the results show that our AdvKin methods also outperform human's evaluation.
5) By comparing AdvKin with E-AdvKin, we obtain that E-AdvKin shows superiority to AdvKin. Thus, more fine-grained kin-related features can be learned with the patch-wise ensemble, such that the information of the augmented features is more complete and discriminative.
6) Since the UB dataset is deployed with triplet samples, in order to obtain more discriminative features, we employ a coarse-to-fine transfer method [58]. Different from [58], in fine-tune step, we remove the original fully connected layers, and add two new fully connected layers, which have 128 and 80 neurons as shown in Table I. The parameters of convolutional layers are frozen, and the fully connected layers are trained on the UB data. By transfer learning from face recognition to kinship verification task, the performance is improved.

To better visualize the performance of different methods, the receiving operating characteristic (ROC) curves of different methods are shown in Fig. 7, in which Fig. 7(a)–(h) describe the ROC curves of the results on KinFaceW-I and KinFaceW-II

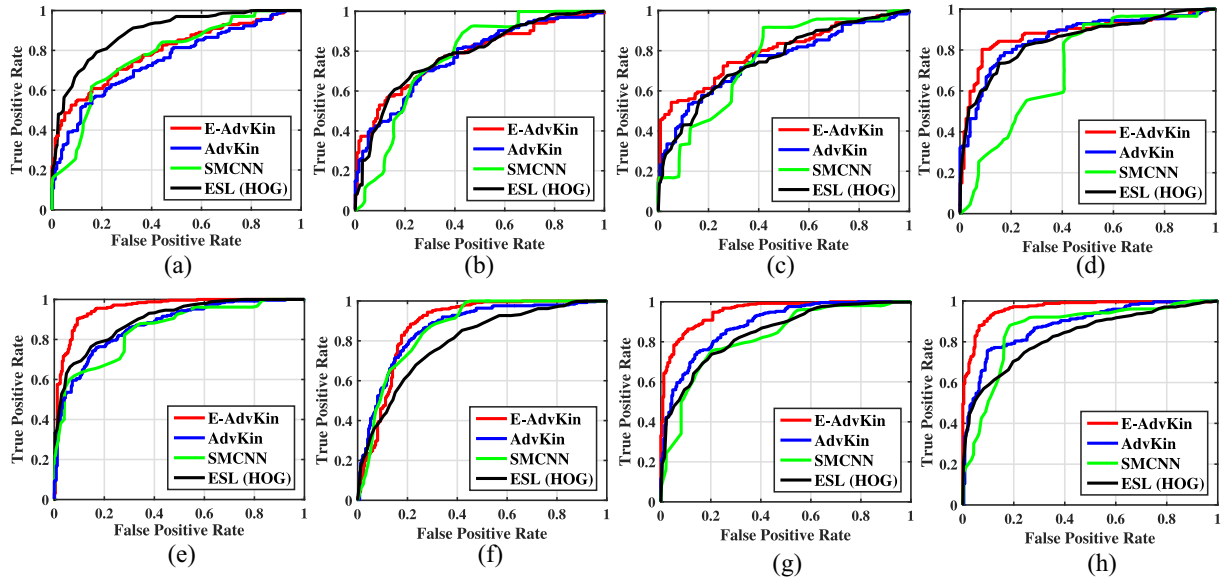

Fig. 7. ROC curves of different methods on KinFaceW-I (upper row) and KinFaceW-II (lower row) datasets. (a) F-S. (b) F-D. (c) M-S. (d) M-D. (e) F-S. (f) F-D. (g) M-S. (h) M-D.

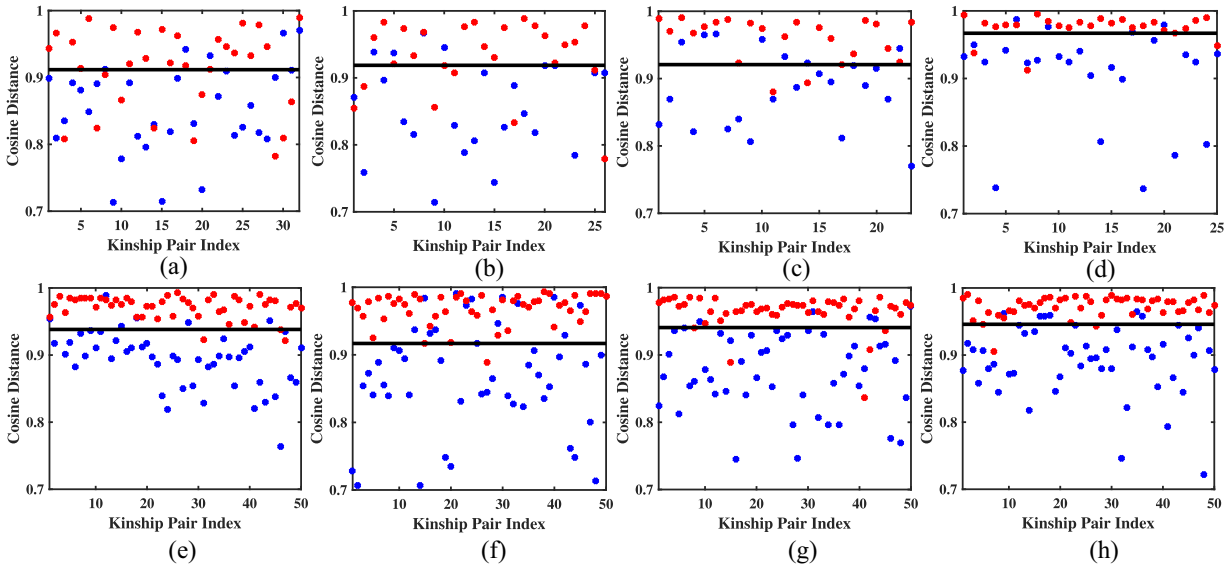

Fig. 8. Cosine distances of kinship pairs on KinFaceW-I (upper row) and KinFaceW-II (lower row) datasets. The red and blue points denote the kinship pairs (positive) and nonkinship pairs (negative), respectively. The black line denotes the threshold. (a) F-S. (b) F-D. (c) M-S. (d) M-D. (e) F-S. (f) F-D. (g) M-S. (h) M-D.

dataset, respectively. We can observe from the results that the proposed AdvKin method can yield competitive performance than others in terms of the ROC curves. Noteworthily, for KinFaceW-I data, the superiority of the proposed AdvKin models is not significant because of the smaller data size. Especially, the ESL (HOG) method is much better than ours in the F-S kinship task as shown in Fig. 7(a). This fully shows that CNN-based methods are more suitable for larger datasets. In addition, the cosine distances between pairwise samples are visualized in Fig. 8. We see that the kin pairs and nonkin pairs are easy to be distinguished.

D. Ablation Analysis of Loss Functions in AdvKin

In order to demonstrate the effectiveness of the AL, the ablation analysis of AdvKin is presented in Table IV. By comparing the MMD-based loss (i.e., ML) with the CL, the proposed methods outperform the CL method with 2% improvement on average. Furthermore, the AL-based AdvKin is superior to ML-based AdvKin with 3% improvement. Thus, the proposed AL can improve the discrimination and robustness of features.

E. Comparison With Previous Feature Fusion Methods

As shown in Table V, by comparing with the previous feature fusion methods, AdvKin still outperforms other methods, except DDMML on KinFaceW-I. It is demonstrated that, the E-AdvKin can further improve the discrimination of deep kin-related feature representation. To be specific, the proposed E-AdvKin shows the best performance (89.9%), which outperforms the AdvKin with 1.9% in average accuracy.

TABLE IV
ACCURACY OF DIFFERENT METHODS WITH DIFFERENT LOSSES IN SMALL-SCALE KINSHIP VERIFICATION TASK

Methods	KinFaceW-I					KinFaceW-II					UB			Cor
	F-S	F-D	M-S	M-D	Mean	F-S	F-D	M-S	M-D	Mean	0-1	0-2	Mean	-
CL	74.7	77.6	72.4	81.1	76.5	**85.8**	85.8	84.0	83.8	84.9	58.3	60.0	59.2	76.2
ML+CL	**77.3**	74.6	**78.0**	**83.6**	78.4	85.8	84.6	86.6	88.0	86.3	59.8	61.0	60.4	78.3
AdvKin	75.7	**78.3**	77.6	83.1	**78.7**	**88.4**	**85.8**	**88.0**	**89.8**	**88.0**	**75.0**	**75.0**	**75.0**	**81.4**

TABLE V
ACCURACY OF DIFFERENT METHODS WITH DEEP AND AUGMENTED (FUSED) FEATURES IN SMALL-SCALE KINSHIP VERIFICATION TASK

Methods	KinFaceW-I					KinFaceW-II				
	F-S	F-D	M-S	M-D	Mean	F-S	F-D	M-S	M-D	Mean
PDFL (LE) [32]	68.2	63.5	61.3	69.5	65.6	77.0	74.3	77.0	77.2	76.4
CNN-Basic [26]	75.7	70.8	73.4	79.4	74.8	84.9	79.6	**88.3**	88.5	85.3
DDML (LPQ) [56]	**83.8**	77.0	78.1	**86.6**	**81.4**	84.8	82.6	79.4	81.8	82.2
WGEML (CNN) [43]	77.0	69.1	**78.8**	78.7	75.9	83.4	75.2	80.2	79.9	79.7
AdvKin	75.7	**78.3**	77.6	83.1	78.7	**88.4**	**85.8**	88.0	**89.8**	**88.0**
MPDFL (Fusion) [32]	73.5	67.5	66.1	73.1	70.1	77.3	74.7	77.8	78.0	77.0
CNN-Points (Fusion) [26]	76.1	71.8	78.0	84.1	77.5	89.4	81.9	89.9	**92.4**	88.4
DDMML (Fusion) [56]	**86.4**	**79.1**	**81.4**	**87.0**	**83.5**	87.4	83.8	83.2	83.0	84.3
WGEML (Fusion) [43]	78.5	73.9	80.6	81.9	78.7	88.6	77.4	83.4	81.6	82.8
E-AdvKin (Fusion)	76.6	77.3	78.4	86.2	79.6	**91.6**	**85.2**	**90.2**	**92.4**	**89.9**

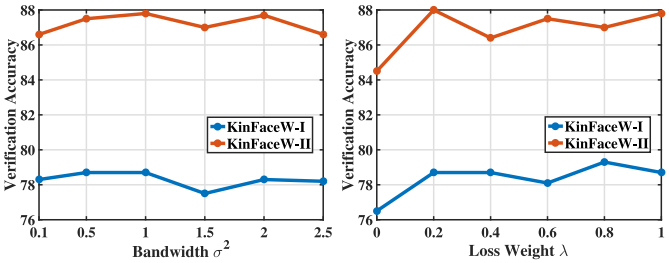

Fig. 9. Accuracy on KinFaceW-I and KinFaceW-II with different bandwidth σ^2 (left) and loss weight λ (right).

F. Hyperparameter Sensitivity Analysis of Model

There are two model parameters, that is, kernel parameter σ^2 and tradeoff parameter λ. Fig. 9 (left) shows the accuracy of KinFaceW-I and KinFaceW-II datasets with respect to different bandwidth σ^2. We see that the proposed AdvKin obtains the best performance when σ^2 is set as 1.0, which is then used throughout all experiments. After fixing σ^2, Fig. 9 (right) shows the accuracy on KinFaceW-I and KinFaceW-II datasets with respect to different loss weight λ. We see that the AdvKin method obtains the best performance when λ is set as 0.2.

VI. EXPERIMENTS FOR LARGE-SCALE TASK

A. Description of Network Architecture and Datasets

Consider the different size of the kinship dataset, the CNN architecture of AdvKin in the large-scale task is slightly different from that of small-scale in network depth. Specifically, a deeper AdvKin network is employed. Because of the excellent performance of ResNet [21] in image classification, the proposed AdvKin method follows a two-stream residual architecture with different depths. The input size of the deeper AdvKin network is 224×224. The details of the two-stream AdvKin networks for large-scale task are described in Table VI.

The large-scale kinship data, FIW [9], is used for large-scale kinship verification task. To the best of our knowledge, FIW is the largest and most comprehensive kinship face database for automatic kinship recognition, which contains over 12 000 family photos of 1001 families. The dataset comes from the first Large-Scale Kinship Recognition Data Challenge in ACM MM 2017.

B. Experimental Setup

In the challenge, we focus on the evaluation protocol of Kinship Verification (Track 1)-based on FIW that includes a total of 6 44 000 pairs, from which 5 38 518 pairs (i.e., over 1 million of face images) of seven different kin-relations are used. These datasets are partitioned into three disjoint sets referred to as Train, Validation, and Test sets. The ground truth for Train and Validation sets are provided, but the Test set is "blind" by the developers. Therefore, the Validation set is used for evaluation. Notably, due to the "blindness" of the Test set, the result of AdvKin is reported with the help of developers, and comparisons to others are unavailable for this dataset.

In the competition, seven different types of kinship: 1) Father–Daughter (F-D); 2) Father–Son (F-S); 3) Mother–Daughter (M-D); 4) Mother–Son (M-S); 5) Sister–Brother (SIBS); 6) Brother–Brother (B-B); and 7) Sister–Sister (S-S) are explored. Specifically, the sample distribution of each type of kinship relation in Train, Validation, and Test is shown as follows.

1) In the Train set, 2 82 186 kinship pairs are included, consisting of 42 458, 53 974, 34 828, 38 312, 40 846, 52 482, and 19 286 pairs for seven different types in order, respectively.
2) In the Validation set, 76 664 kinship pairs are included, consisting of 11 460, 13 696, 10 698, 9816, 7434, 17 342, and 6218 pairs for seven different types in order, respectively.

TABLE VI
TWO-STREAM ADVKIN NETWORK ARCHITECTURE FOR LARGE-SCALE KINSHIP VERIFICATION TASK. THE STACKED CONVOLUTION BLOCKS ARE SHOWN IN BRACKETS. DOWN-SAMPLING IS PERFORMED FROM CONV1_X TO CONV5_X LAYERS WITH A STRIDE OF 2

CNN	Conv1_x	Conv2_x	Conv3_x	Conv4_x	Conv5_x	Conv6_x	FC1	FC2	Softmax
AdvKin	$3\times3, 32$ $3\times3, 64$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times1$ $3\times3, 128$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$ $3\times3, 256$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times5$ $3\times3, 512$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	-	1024	512	300
AdvKin (deeper)	$3\times3, 32$ $3\times3, 64$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times1$ $3\times3, 128$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$ $3\times3, 256$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times5$ $3\times3, 512$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	1024	512	300

TABLE VII
ACCURACY OF ADVKIN WITH DIFFERENT METHODS IN LARGE-SCALE KINSHIP VERIFICATION TASK

Methods	M-D	M-S	S-S	B-B	SIBS	F-S	F-D	Mean
VGG-Face [40]	65.99	58.88	74.59	**71.99**	64.69	64.71	62.87	66.25
ResNet-29 [19]	59.55	59.08	51.74	64.81	59.39	58.21	56.54	58.47
ResNet-22(finetune) [11]	**71.09**	**68.63**	69.54	69.88	69.54	67.73	**68.15**	69.22
E-AdvKin	69.93	67.33	**77.44**	71.76	**69.80**	**68.77**	67.82	**70.41**

TABLE VIII
ACCURACY OF DIFFERENT MODEL, LOSS AND FEATURE AUGMENTATION IN LARGE-SCALE KINSHIP VERIFICATION TASK. NOTE THAT CL MEANS CONTRASTIVE LOSS, 2L MEANS CL PLUS AL, 3L MEANS THE JOINT LOSS OF CL, AL, AND SL

Index	Loss	Model	M-D	M-S	S-S	B-B	SIBS	F-S	F-D	Mean
0	CL	AdvKin (CL)	61.06	61.95	62.45	65.35	62.05	61.33	59.18	61.91
1	2L	AdvKin (2L)	60.50	64.07	64.17	63.76	61.99	62.23	60.53	62.46
2	3L	AdvKin (3L)	**64.11**	65.65	64.53	**65.80**	64.82	63.42	63.18	64.50
3	3L	AdvKin (3L deeper)	63.56	**66.80**	**65.48**	65.77	**65.35**	**64.14**	**63.59**	**64.97**
4	SL	VGG-Face [40]	65.99	58.88	74.59	**71.99**	64.69	64.71	62.87	66.25
1+2+3	Joint	E-AdvKin	64.20	**67.55**	65.71	66.82	66.45	64.78	64.04	65.65
2+3+4	Joint	E-AdvKin	**70.07**	65.60	**77.52**	71.88	69.72	**68.79**	67.56	70.16
1+2+3+4	Joint	E-AdvKin	69.93	67.33	77.44	71.76	**69.80**	68.77	**67.82**	**70.41**

3) In the *blind* Test set, 1 79 668 kinship pairs are included, consisting of 23 506, 45 988, 20 674, 47 954, 15 076, 19 946, and 6524 pairs for seven different types in order, respectively.

In experiments, the proposed AdvKin with different loss is trained from scratch on the Train set, and finally, Euclidean distance is used for kinship verification on the Validation set. In model optimization, the mini-batch SGD-based back-propagation algorithm is used for training, with an initial learning rate of 10^{-2}, and the margin of CL is set as 1. The batch size is set as 22 for large-scale kinship verification task. The deeper AdvKin model for larger-scale kinship verification task is trained on three pieces of NVIDIA 1080Ti GPUs for about 20 h.

C. Comparison With Deep Kinship Verification Models

The verification results of the state-of-the-art deep methods (e.g., VGG-Face, ResNet) on large-scale kinship verification task (i.e., FIW challenge) are shown in Table VII. VGG-Face [40] is deployed with VGG-16, which is pretrained on 2.6 million of face images from 2622 different celebrities. ResNet-29 [19] is a 29-layered residual CNN trained on CASIA-WebFace [59]. Both of them are state-of-the-art methods for face verification. In addition, the results of fine-tuned ResNet-22 on FIW kinship faces are also presented in Table VII, that is, ResNet-22(finetune). It is demonstrated that the proposed method outperforms state-of-the-art deep learning-based methods on large-scale kinship verification task. Therefore, it can be concluded that the kin-related characteristic information can be exploited more effectively through the proposed adversarial learning mechanism in this article.

D. Ablation Analysis of Different Losses in AdvKin

In order to present the ablation analysis of loss functions, the joint loss function formulated in (5) with AL and CL is simplified as 2L in short for convenience. The joint loss formulated in (10) with the 2L loss and the SL is simplified as 3L in short. The loss weight is set as 1. As can be seen from Table VIII, the results of 2L outperform the CL, which denotes that the AL can improve the discrimination of the kin-relation features. In addition, the results of 3L outperform the 2L by feeding the family ID supervised SL into our network, which demonstrates that the SL can effectively improve the separability of kinship features. The results fully confirm that the superior performance of the proposed AdvKin model is reasonable.

Depth is a very important factor of the CNN model for classification performance [60]. In order to demonstrate the impact of depth in AdvKin, under different depth, the results of AdvKin and AdvKin(deeper) are listed in Table VIII. It can be seen that the deeper AdvKin has a slight improvement of 1.5% in average accuracy, which shows the impact of depth.

E. Comparison Between AdvKin and E-AdvKin

The performance comparison of the single AdvKin model and multimodel E-AdvKin are shown in Table VIII, in which the features from index 0, 1, 2, 3, and 4 represent the single

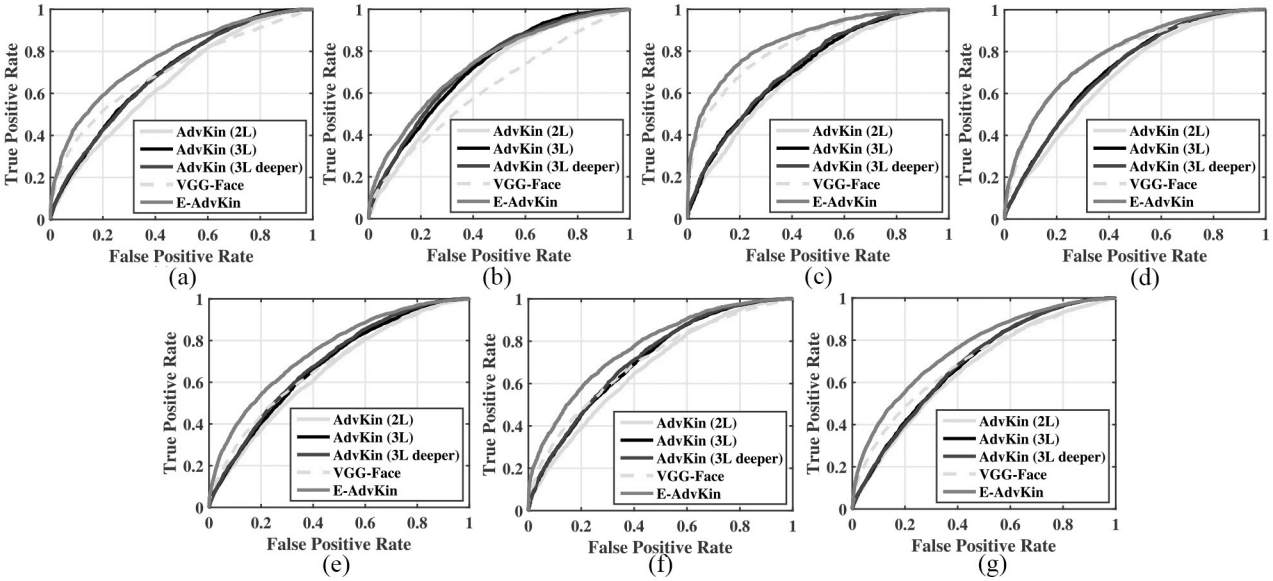

Fig. 10. ROC curves of different models on seven types of kin-relation. (a) M-D. (b) M-S. (c) S-S. (d) B-B. (e) F-D. (f) SIBS. (g) F-S.

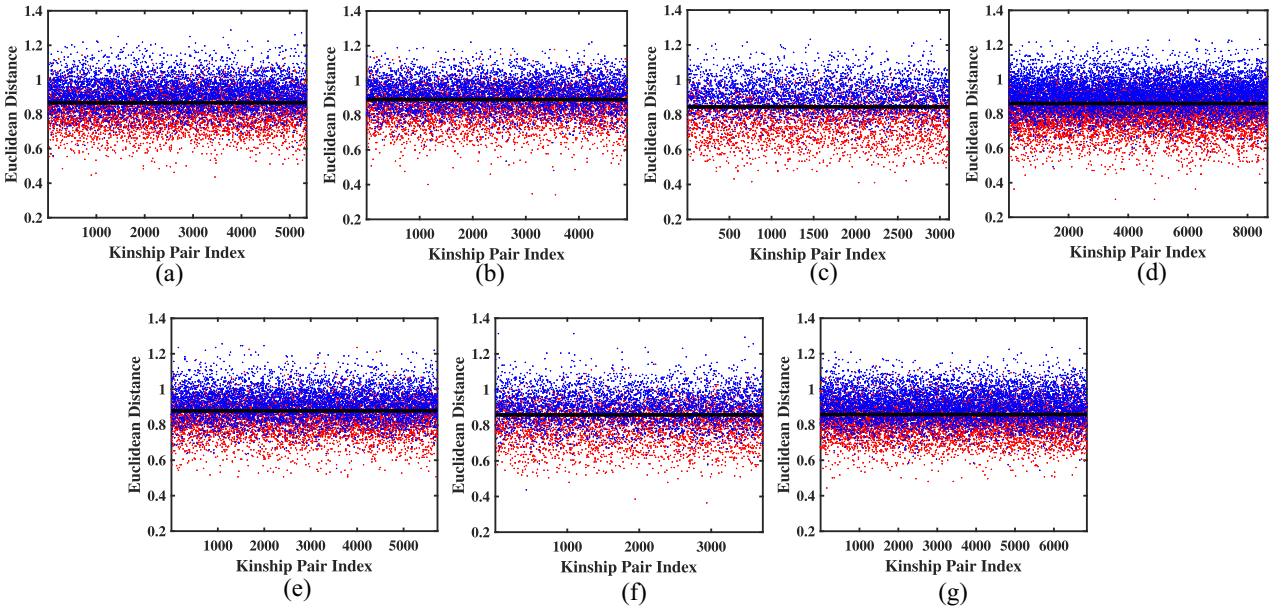

Fig. 11. $\mathcal{L}_2$-distances of kinship pairs on seven types of kin-relation. The points in red and blue represent the distance between the kinship pairs and between the nonkinship pairs, respectively. The black line denotes the searched threshold for verification. (a) M-D. (b) M-S. (c) S-S. (d) B-B. (e) F-D. (f) SIBS. (g) F-S.

feature (without augmentation) and the last three rows denote the performance after feature augmentation by network fusion, that is, E-AdvKin, which concatenates the features from each model together. The dimension of the augmented feature (e.g., $1+2+3$) is 1536 (512×3). In addition, consider the excellent performance of the VGG-Face model, it is used as the feature extractor of FIW faces in this article, and the dimension of features extracted from the VGG-Face model is 4096. After the ensemble of the four networks (e.g., $1+2+3+4$), we can observe significant performance improvement of 5% in average accuracy. Notably, the $\mathcal{L}_2$-normalization is used twice before and after feature augmentation. It is noteworthy that, although the performance of the VGG-Face model is slightly better than AdvKin, the number of training data of AdvKin (i.e., 0.01 million of faces) is 200 times less than the VGG-Face model (i.e., 2.6 millions of faces). Therefore, a direct comparison between AdvKin and VGG-Face is unfair.

To better visualize the performance of different methods, the ROC curves of different methods are shown in Fig. 10, in which Fig. 10(a)–(g) describe the ROC curves for seven types of kinship relation. We can observe that the proposed ensemble model (E-AdvKin) can yield the best verification performance for all the tasks.

In addition, for better insight of the augmented features, the Euclidean distances of kinship pairs based on the augmented features are visualized in Fig. 11. We observe that most of

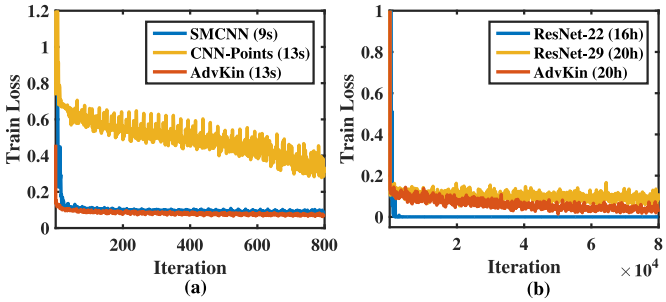

Fig. 12. Comparison of convergence and training time on (a) small-scale and (b) large-scale kinship verification tasks.

the kin pairs and nonkin pairs can be easily distinguished via an appropriate threshold, which is indicated by a black line. However, there are still many incorrectly recognized pairs with $\mathcal{L}_2$-distance. In the future, metric-learning models can be further exploited on the deep representational features for jointly learning more effective distance similarity metrics instead of the Euclidean distance metric.

F. Competition Results on the Blind Test Set

For competition on the blind test set (the label of the test set is unavailable), the proposed E-AdvKin and VGG-Face model are finally used. With the help of the developers of this competition, the final verification accuracies on the test set are 70.66%, 65.22%, 72.10%, 63.59%, 66.51%, 63.38%, and 64.60% for M-D, M-S, S-S, B-B, SIBS, F-S, and F-D, respectively. The average accuracy of the seven kinship verification tasks is 66.58% and ranks the third position. Notably, since the labels of the test set are blind and unavailable, comparisons with other methods are not presented in this article.

G. Convergence and Training Time

The convergence and training time of the proposed AdvKin and other methods are presented. For the small-scale dataset, the convergence and training time (second) are shown in Fig. 12(a). For the large-scale dataset, the convergence and training time (hour) are shown in Fig. 12(b). We observe that the convergence speed and training time of our model are comparable to others, even with an adversarial mechanism in AdvKin.

VII. Conclusion

In this article, we proposed a two-stream family ID-based AdvKin network model for small-scale and large-scale kinship verification tasks, which is motivated by a self-adversarial learning idea. The self-adversarial learning mechanism is achieved by proposing an AL that works jointly with the family ID-based CL and SL. In order to further promote the performance of our AdvKin method, an E-AdvKin is then proposed with two types of feature augmentation (i.e., patch level fusion and network level fusion). Extensive kinship verification experiments on the small-scale benchmarks and the large-scale benchmark show the superiority of our proposed methods over many state-of-the-art algorithms. In our future work, we will consider more self-adversarial layers in convolution modules instead of fully connected layer with the triplet network architecture, so that the discrimination of kin-relation features can be better improved through multiple self-adversarial training strategy. In addition, more challenging backbones can be exploited in self-adversarial learning.

References

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, 2015, pp. 815–823.
[2] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, "Neighborhood repulsed metric learning for kinship verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 331–345, Feb. 2014.
[3] M. Shao, S.-Y. Xia, and Y. Fu, "Genealogical face recognition based on UB KinFace database," in *Proc. CVPRW*, 2011, pp. 60–65.
[4] X. Zhou, J. Hu, J. Lu, Y. Shang, and Y. Guan, "Kinship verification from facial images under uncontrolled conditions," in *Proc. ACM MM*, 2011, pp. 953–956.
[5] N. Kohli, R. Singh, and M. Vatsa, "Self-similarity representation of Weber faces for kinship classification," in *Proc. ICB*, 2012, pp. 245–250.
[6] R. Fang, K. D. Tang, N. Snavely, and T. Chen, "Towards computational models of kinship verification," in *Proc. ICIP*, 2010, pp. 1577–1580.
[7] S. Xia, M. Shao, and Y. Fu, "Kinship verification through transfer learning," in *Proc. IJCAI*, 2011, pp. 2539–2544.
[8] N. Kohli, M. Vatsa, R. Singh, A. Noore, and A. Majumdar, "Hierarchical representation learning for kinship verification," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 289–302, Jan. 2017.
[9] J. P. Robinson, M. Shao, Y. Wu, and Y. Fu, "Families in the wild (FIW): Large-scale kinship image database and benchmarks," in *Proc. ACM MM*, 2016, pp. 242–246.
[10] H. Dibeklioglu, A. A. Salah, and T. Gevers, "Like father, like son: Facial expression dynamics for kinship verification," in *Proc. ICCV*, 2013, pp. 1497–1504.
[11] J. P. Robinson, M. Shao, Y. Wu, H. Liu, T. Gillis, and Y. Fu, "Visual kinship recognition of families in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2624–2637, Nov. 2018.
[12] H. Yan and J. Hu, "Video-based kinship verification using distance metric learning," *Pattern Recognit.*, vol. 75, pp. 15–24, Mar. 2018.
[13] Y.-G. Zhao, Z. Song, F. Zheng, and L. Shao, "Learning a multiple kernel similarity metric for kinship verification," *Inf. Sci.*, vols. 430–431, pp. 247–260, Mar. 2018.
[14] J. Hu, J. Lu, Y.-P. Tan, J. Yuan, and J. Zhou, "Local large-margin multi-metric learning for face and kinship verification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1875–1891, Aug. 2018.
[15] L. Li, X. Feng, X. Wu, Z. Xia, and A. Hadid, "Kinship verification from faces via similarity metric based convolutional neural network," in *Proc. ICIAR*, 2016, pp. 539–548.
[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[18] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. ICML*, 2011, pp. 513–520.
[19] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. ECCV*, 2016, pp. 499–515.
[20] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. CVPR*, 2017, pp. 6738–6746.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2015, pp. 770–778.
[22] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, 2017, pp. 2261–2269.
[23] Y.-M. Zhang, K. Huang, X. Hou, and C. Liu, "Learning locality preserving graph from data," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2088–2098, Nov. 2014.
[24] L. Zhang, X. Wang, G.-B. Huang, T. Liu, and X. Tan, "Taste recognition in e-tongue using local discriminant preservation projection," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 947–960, Mar. 2019.
[25] M. Wang, Z. Li, X. Shu, and J. Wang, "Deep kinship verification," in *Proc. IEEE Int. Workshop MSP*, 2015, pp. 1–6.

[26] K. Zhang, Y. Huang, C. Song, H. Wu, and L. Wang, "Kinship verification with deep convolutional neural networks," in *Proc. BMVC*, 2015, pp. 1–12.
[27] M. Dawson, A. Zisserman, and C. Nellaker, "From same photo: Cheating on visual kinship challenges," in *Proc. ACCV*, 2018, pp. 654–668.
[28] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 3730–3738.
[29] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, 2015, pp. 97–105.
[30] Q. Duan, L. Zhang, and W. Jia, "Adv-kin: An adversarial convolutional network for kinship verification," in *Proc. CCBR*, 2017, pp. 48–57.
[31] Q. Duan and L. Zhang, "AdvNet: Adversarial contrastive residual net for 1 million kinship recognition," in *Proc. ACM MMW*, 2017, pp. 21–29.
[32] H. Yan, J. Lu, and X. Zhou, "Prototype-based discriminative feature learning for kinship verification," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2535–2545, Nov. 2015.
[33] X. Zhou, Y. Shang, H. Yan, and G. Guo, "Ensemble similarity learning for kinship verification from facial images in the wild," *Inf. Fusion*, vol. 32, pp. 40–48, Nov. 2016.
[34] X. Zhou, J. Lu, J. Hu, and Y. Shang, "Gabor-based gradient orientation pyramid for kinship verification under uncontrolled environments," in *Proc. ACM MM*, 2012, pp. 725–728.
[35] H. Yan, "Learning discriminative compact binary face descriptor for kinship verification," *Pattern Recognit. Lett.*, vol. 117, pp. 146–152, Jan. 2019.
[36] S. Xia, M. Shao, J. Luo, and Y. Fu, "Understanding kin relationships in a photo," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1046–1056, Aug. 2012.
[37] H. Yan, J. Lu, W. Deng, and X. Zhou, "Discriminative multimetric learning for kinship verification," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 7, pp. 1169–1178, Jul. 2014.
[38] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. CVPR*, 2014, pp. 1891–1898.
[39] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. CVPR*, 2014, pp. 1701–1708.
[40] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. BMVC*, 2015, pp. 1–12.
[41] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[42] H. Dibeklioglu, "Visual transformation aided contrastive learning for video-based kinship verification," in *Proc. ICCV*, 2017, pp. 2478–2487.
[43] J. Liang, Q. Hu, C. Dang, and W. Zuo, "Weighted graph embedding-based metric learning for kinship verification," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1149–1162, Mar. 2019.
[44] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. NIPS*, 2015, pp. 1486–1494.
[45] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. CVPR*, 2017, pp. 105–114.
[46] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint:11605.05396*, 2016.
[47] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv*, 2015.
[48] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv*, 2015.
[49] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint:1703.10593*, 2017.
[50] T. Luan, X. Yin, and X. Liu, "Disentangled representation learning GAN For pose-invariant face recognition," in *Proc. CVPR*, 2017, pp. 1283–1292.
[51] K. M. Borgwardt *et al.*, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, 2006.
[52] L. Zhang, W. Zuo, and D. Zhang, "LSDT: Latent sparse domain transfer learning for visual adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1177–1191, Mar. 2016.
[53] L. Zhang and D. Zhang, "Robust visual knowledge transfer via extreme learning machine based domain adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4959–4973, Oct. 2016.
[54] L. Zhang, J. Yang, and D. Zhang, "Domain class consistency based transfer learning for image classification across domains," *Inf. Sci.*, vols. 418–419, pp. 242–257, Dec. 2017.
[55] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.
[56] J. Lu, J. Hu, and Y.-P. Tan, "Discriminative deep metric learning for face and kinship verification," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4269–4282, Sep. 2017.
[57] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[58] Q. Duan, L. Zhang, and W. Zuo, "From face recognition to kinship verification: An adaptation approach," in *Proc. ICCVW*, 2017, pp. 1590–1598.
[59] D. Yi, Z. Lei, S. Liao, and Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.
[60] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, 2015, pp. 1–9.

Lei Zhang (Senior Member, IEEE) received the Ph.D. degree in circuits and systems from the College of Communication Engineering, Chongqing University, Chongqing, China, in 2013.

He worked as a Postdoctoral Fellow with the Hong Kong Polytechnic University, Hong Kong, from 2013 to 2015. He is currently a Professor/Distinguished Research Fellow with Chongqing University. He has authored more than 90 scientific papers in top journals, such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS; the IEEE TRANSACTIONS ON IMAGE PROCESSING; the IEEE TRANSACTIONS ON MULTIMEDIA; the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT; the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS; and top conferences, such as ICCV, AAAI, ACM MM, and ACCV. His current research interests include machine learning, pattern recognition, computer vision, and intelligent systems.

Prof. Zhang was a recipient of the Best Paper Award of CCBR2017, the Outstanding Doctoral Dissertation Award of Chongqing, China, in 2015; the Hong Kong Scholar Award in 2014; and the New Academic Researcher Award for Doctoral Candidates from the Ministry of Education, China, in 2012. He serves as an Associate Editor for the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, and IEEE TRANSACTIONS ON NEURAL NETWORKS.

Qingyan Duan (Student Member, IEEE) received the graduation degree from the Hefei University of Technology, Hefei, China, in 2012, and the M.Sc. degree from Chongqing University, Chongqing, China, in 2016, where she is currently pursuing the Ph.D. degree.

Her current research interests include deep learning, pattern recognition, and computer vision.

David Zhang, photograph and biography not available at the time of publication.

Wei Jia, photograph and biography not available at the time of publication.

Xizhao Wang, photograph and biography not available at the time of publication.