Evaluation-oriented Knowledge Distillation for Deep Face Recognition

Yuge Huang* Jiaxiang Wu* Xingkun Xu Shouhong Ding‡
Youtu Lab, Tencent
{yugehuang, willjxwu, xingkunxu, ericshding}@tencent.com
https://github.com/Tencent/TFace/tree/master/recognition/tasks/ekd

Abstract

Knowledge distillation (KD) is a widely-used technique that utilizes large networks to improve the performance of compact models. Previous KD approaches usually aim to guide the student to mimic the teacher's behavior completely in the representation space. However, such one-to-one corresponding constraints may lead to inflexible knowledge transfer from the teacher to the student, especially those with low model capacities. Inspired by the ultimate goal of KD methods, we propose a novel Evaluation-oriented KD method (EKD) for deep face recognition to directly reduce the performance gap between the teacher and student models during training. Specifically, we adopt the commonly used evaluation metrics in face recognition, i.e., False Positive Rate (FPR) and True Positive Rate (TPR) as the performance indicator. According to the evaluation protocol, the critical pair relations that cause the TPR and FPR difference between the teacher and student models are selected. Then, the critical relations in the student are constrained to approximate the corresponding ones in the teacher by a novel rank-based loss function, giving more flexibility to the student with low capacity. Extensive experimental results on popular benchmarks demonstrate the superiority of our EKD over state-of-the-art competitors.

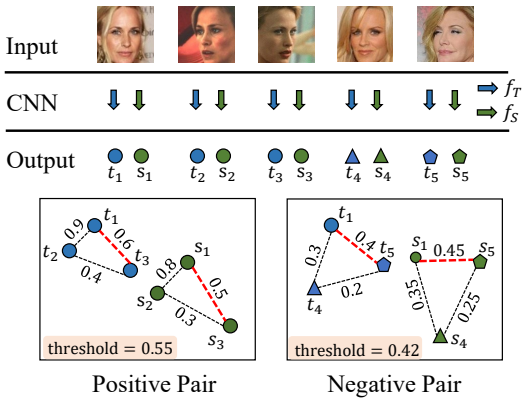


Figure 1. Illustration of critical relations of samples. Different colors indicate different models (Teacher T in blue and Student S in green). Different shapes indicate samples of different subjects. The numbers denote the cosine similarities of samples. The relation of the 1st and the 3rd samples is the only one whose similarities fall on the different side of the threshold in teacher and student models (*i.e.*, $0.6 > 0.55$ in teacher while $0.5 < 0.55$ in student), and thus leads to the TPR difference. Therefore, in order to pursue the same TPR of the teacher, the student which has limited model capability should pay more attention on the relation (in red) of the 1st and the 3rd samples which is the **critical relation**. Similarly, for the negative pairs, the relation of 1st and 5th samples leads to the FPR difference and should be paid more attention.

1. Introduction

With a large number of recognition systems deployed on mobile and edge devices, compact yet discriminative models are in increasingly high demand. Although some optimized neural network architectures for mobile devices [4, 25] are proposed in the recent years, there still exists an enormous performance gap between these compact networks and the resource-intensive networks which have millions of parameters. In order to narrow the gap, Knowledge Distillation (KD), which is a widely-used technique that utilizes the knowledge of a large network to improve the performance of the compact models, is proposed.

The seminal works [2, 10] introduced the original idea of KD, which targets on reducing the Kullback–Leibler (KL) divergence between each instance's probabilities at the output layers of the teacher and the student networks. In the past decade, work [13, 24, 33] has continued optimizing KD methods by extending such instance-wise constraints to the activation of the hidden layers. For example, attention transfer [33] aims to elicit similar response patterns in feature maps. FitNets [24] directly constrains intermediate representations by using regressions. However, such instance-based methods essentially require the teacher and student to share the same representation space, which is unrealistic

* equal contribution. ‡ corresponding author.

for student networks with low model capacities. As a result, these instance-wised methods bring limited improvement on the performance of student models. Recently, relation-based KD methods [20,23,31] are proposed. Different from the traditional instance-based methods, relation-based ones utilize the correlations between instances as knowledge. The students in these methods are not required to mimic the teacher's representation space, but rather to preserve the relations of samples in their own representation space. Thus, they can achieve relatively better performance comparing to the instance-based methods. However, the model performance trained with these methods are still far from perfect as they still have too strict constraint on knowledge transfer. In particular, they require the student to mimic all relations between samples in a mini-batch, which seriously limits the flexibility and efficiency of the knowledge transfer from the teacher to the student.

Unlike all the KD methods mentioned earlier, we propose a novel Evaluation-oriented Knowledge Distillation (EKD) method for deep face recognition, which draws inspiration from the ultimate goal of KD, that is, to reduce the performance gap between the teacher and student models. Specifically, we adopt the commonly used evaluation metrics in face recognition, *i.e.*, False Positive Rate (FPR), and True Positive Rate (TPR) as the performance indicator of a face recognition model. By performing these two evaluation metrics during the student model training, we can directly obtain the critical pair relations which cause the TPR and FPR difference between the teacher and student models. Naturally, these critical pairs should be mainly focused on during knowledge transfer. Thus, we adopt a novel rank-based loss function to constrain the critical relations in the student to approximate the teacher's corresponding ones. Fig. 1 gives a motivational example and illustrates how critical relations cause the difference of TPR and FPR between the teacher and student models. Generally, the thresholds of a face recognition model are determined by target FPRs from the similarities of whole negative pairs and are usually different for different models, even if corresponding to the same FPR. For clarity, we directly give 0.55 and 0.42, which roughly correspond to FPR=$1e$-5 and FPR=$1e$-4, as the thresholds of the student and teacher model.

Although both the proposed EKD and the relation-based KD methods optimize the relations between samples, they differ in two aspects. First, the previous relation-based KD methods require the student to mimic all the relations of the teacher to indirectly reduce the performance gap between the teacher and student models, while our EKD introduces the commonly used evaluation protocol, *i.e.*, TPR and FPR, into the training process and optimizes the critical relations that cause the TPR and FPR difference in the student model to reduce these two metrics gap. Second, the previous relation-based KD methods usually constrain the absolute similarity of the corresponding pair between the teacher and student models, while our EKD relaxes the constraint by a novel rank-based loss function, which only requires the similarities of the corresponding pairs on the same side of the thresholds in the teacher and student models.

The contributions of this paper are summarized as follows:

- We propose a novel Evaluation-oriented KD method for deep face recognition. To our best knowledge, EKD is the first KD method to directly reduce the evaluation metric difference between the teacher and student model during training.
- We propose a novel rank-based loss function to optimize the student model's critical relations that cause the TPR and FPR difference between the teacher and student models. By only constraining the similarities of the corresponding pairs are on the same side of the thresholds in the teacher and student models, it gives more flexibility to the student, thereby alleviating the student's low capacity problem.
- We conduct extensive experiments on popular facial benchmarks, which demonstrate the superiority of the proposed EKD over the SOTA competitors.

2. Related Work

Loss Function on Face Recognition. Designing a suitable loss function plays a vital role in deep face recognition. The commonly used loss function can be categorized into two types: metric loss and classification loss. Metric losses such as the contrastive [28] and the triplet [21,26] loss are designed to increase the margin in the Euclidean distance space. Current SOTA deep face recognition methods mostly adopt softmax-based classification loss [6,12,16,30]. Though such margin-based loss functions equipped with large neural networks are verified to obtain satisfactory performance [6], they do not always perform well with a mobile neural network [7]. The performance gap between the large and compact model motivates us to explore the knowledge distillation method.

Knowledge Distillation. Knowledge distillation has been actively investigated and widely used in many computer vision tasks. The basic idea proposed by Hinton et al. [10] minimizes the KL divergence of soften class probabilities between the teacher and student. Later, several variants of distillation strategies are proposed to make better use of the teacher network's information. They mainly fall into two categories, *i.e.*, instance-based methods, and relation-based

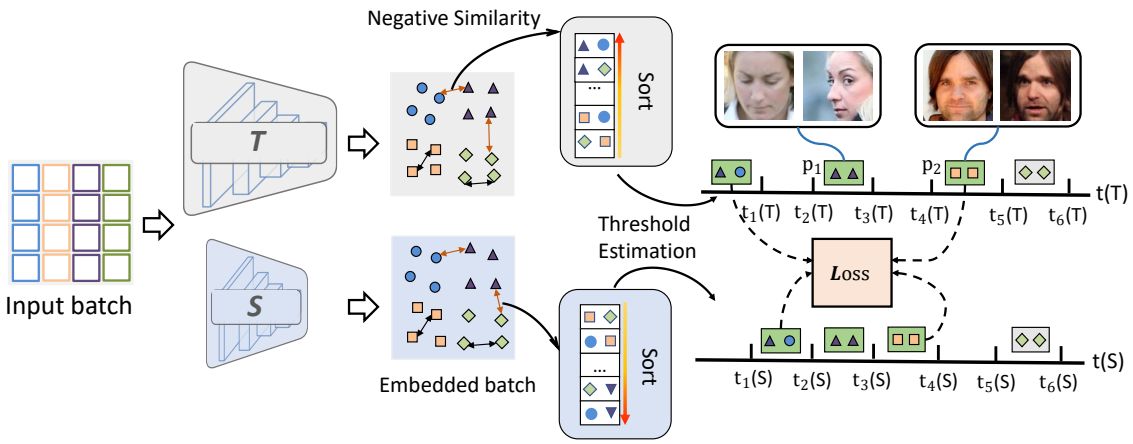


Figure 2. Illustration of EKD. T and S denote the teacher and student network, p_1 and p_2 denote the two positive pair relations, respectively. The critical pair-wise relations that cause the TPR and FPR difference between the teacher and student model are selected and constrained by the loss function.

methods. Instance-based methods transfer individual outputs from a teacher model to a student model point-wise. For example, FitNets [24] use the intermediate representations of a teacher network to guide the feature activation of a student network. KD methods especially proposed for face recognition are also mainly in this category. ShrinkTeaNet [8] minimizes the angle of each face sample between teacher and student embedding vectors. TripletDistillation [9] improves the triplet loss with dynamic margins by utilizing the similarity structures among different identities in the teacher network. MarginDistillation [29] uses class centers from the teacher network for the student network. Unlike the instance-based methods, relation-based methods [5, 20, 23, 31] transfer relations of the samples in a batch. RKD [20] utilizes two concrete relations, *i.e.*, pairwise and ternary relations of examples. SP [31] and CCKD [20] adopt the pairwise similarities of the outputs. Darkrank [5] transfers similarity ranks between data examples. Although the model performance trained with the two types of KD methods is better than direct training, it is still far from perfect as these methods have too strict constraints on knowledge transfer. In particular, instance-based methods require the teacher and student to share the same representation space, while relation-based methods require the student to mimic all relations between samples in a mini-batch.

Our method is related to relation-based methods, but there are several key differences. Compared with RKD [20] and SP [31], our method improves in two aspects: 1) EKD focuses on the critical relations that cause the TPR and FPR difference between the teacher and student models, while RKD and SP treat all the possible relations equally. 2) EKD constrains the critical relations by a novel rank-based loss function to give more flexibility to the student with low capacity, while RKD and SP directly constrain the corresponding similarities by L2 loss. Our EKD and DarkRank [5] differ in two aspects: 1) EKD adopts the rank between a certain similarity and the thresholds estimated from the total negative pairs in a mini-batch, while DarkRank uses the rank based on the similarity score between the candidate samples and a query sample. 2) EKD calculates the rank with an indicator function, which can be simply approximated by a sigmoid function, while DarkRank uses the way introduced by classical list-wise learning to rank methods [3]. Thus, our method is far simpler to implement. Besides, the critical relation selection of our method is different from the common hard sample mining strategies in previous methods [9, 17]. As illustrated in Fig. 2, the positive pair p_1 is more likely to be mined in previous hard sample mining methods. On the contrary, the positive pair p_2 is mined in our approach since it leads to the TPR difference between the teacher and student models.

3. The Proposed Method

Fig. 2 illustrates the framework of the proposed EKD. Given a teacher model $\mathbf{T}$ and a student model $\mathbf{S}$, we let $f_{\mathcal{T}}$ and $f_{\mathcal{S}}$ be functions of the teacher and the student, respectively. We follow batch construction from RKD [20] and sample q positive images per category in a mini-batch. Thus, the features extracted by T and S can be used to construct the positive and negative pairs, respectively. Then, according to the commonly used evaluation protocol TPR and FPR in face recognition, the critical pair-wise relations that cause the two metrics difference between the teacher and student models are chosen (see Sec. 3.1). Finally, we constrain the critical relations by a novel rank-based loss function (see Sec. 3.2), giving more flexibility to the stu-

dent and alleviating the student's low capacity problem.

3.1. Critical Relation Selection

Positive Pairs and Negative Pairs. First, we introduce the details of constructing the positive and negative pairs in one mini-batch during training. A balanced mini-batch consists of p classes, each class with q images. Therefore, there are $B = p*q$ samples in each mini-batch. The number of total pairs are $B*(B-1)/2$, where $p*q*(q-1)/2$ are number of positive pairs and $p*q*(p-1)*q/2$ are negative pairs. Following prior art [6, 12] in face recognition, we adopt the cosine similarity to denote the pair-wise relation:

$$s_{i,j} = \langle f(x_i), f(x_j) \rangle, i \neq j \tag{1}$$

where $f(x_i)$ denotes the representation of a sample.

FPR and TPR Calculation. Our method's motivation is directly taking reducing the performance gap between the teacher and the student model as the training constraint. Thus, the critical problem is to select a suitable evaluation metric as the performance indicator of the model. In face recognition, TPR and FPR are the most commonly used evaluation metrics. Thus, we adopt these two evaluation metrics as the model's performance indicator in this work. We first briefly describe the evaluation protocol of the two metrics. Given a vector of M similarities v from all the negative pairs, the FPR is computed as the proportion above t.

$$FPR(t) = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}(v_i > t) \tag{2}$$

where t is a chosen threshold, $\mathbb{1}(x)$ is the discrete Indicator function and v_i denotes the similarity of i relation. Similarly, given a vector of N genuine scores u from all the positive pairs, the TPR is computed as the proportion above a threshold t as follows.

$$TPR(t) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(u_i > t) \tag{3}$$

In practice, the typical way to assess two face recognition models is to fix their FPRs and compare their TPRs. Specifically, the thresholds corresponding to each FPR are determined by the quantiles of all the negative pair similarities, and the TPRs can be calculated from the positive pair similarities based on the obtained thresholds. The higher the TPRs, the better the model. The concerned FPR range depends on the deployment scenario of the face recognition system. For example, the FPR is usually set to be $1e$-5 or $1e$-6 in a face access control system to balance security and user experience. In the popular public face benchmarks, the FPR often ranges from $1e$-1 to $1e$-6 [14, 18, 32]. Thus, we choose [$1e$-1, $1e$-6] as the target FPR range. Correspondingly, a vector of 6 thresholds corresponding to the FPR range evenly spaced on a logarithmic scale can be obtained. Since the number of negative pairs from one training mini-batch is not large enough, the threshold corresponding to a small FPR value, *e.g.*, $1e$-6 may has a large variance. We follow [15] to utilize Exponential Moving Average (EMA) to address this issue. Let e_k^n be the estimated k-th threshold of the n-th batch for the specific FPR and therefore we have:

$$t_k = \alpha t_k + (1-\alpha) e_k^n, \tag{4}$$

where t_k is the k-th threshold and initialized with 0; α is the momentum parameter and set to 0.99.

Critical Relation Selection. According to the above evaluation process, once the thresholds are chosen according to the target FPR ranges, the positive pair relations that cause the TPR difference between the teacher and student model can be obtained. Though the FPR has been fixed when estimating the corresponding threshold, the difference of the negative pairs in teacher and student models that cause the false positive cases is also instructive during the knowledge transfer. Thus, the critical relations that cause the difference between the teacher and student models can be defined as follows:

$$\mathbb{1}(s_{i,j}(T) > t_k(T)) \neq \mathbb{1}(s_{i,j}(S) > t_k(S)) \tag{5}$$

where $s_{i,j}(T)$ and $s_{i,j}(S)$ denote the similarities between the i and j samples, and $t_k(T)$ and $t_k(S)$ are k-th thresholds in the teacher and student models, respectively. The relation between the i and j samples can be positive and negative pairs.

3.2. Evaluation-oriented Knowledge Distillation

Let $s_{i,j}(T)$ and $s_{i,j}(S)$ denote the similarities between the i and j samples in the teacher and student, respectively. For brevity, the i and j indexes are omitted. To constrain the critical relations in the student to approximate the corresponding ones in the teacher model, a common loss can be defined as:

$$\mathcal{L}_k = \|s(T) - t_k(T) - (s(S) - t_k(S))\| \tag{6}$$

where $t_k(T)$ and $t_k(S)$ are the k-th thresholds of the teacher and student model, respectively. Assuming there are K thresholds and N critical relations, the loss function can be formulated as follows:

$$\mathcal{L}_{hard} = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \|s_n(T) - t_k(T) - (s_n(S) - t_k(S))\| \tag{7}$$

This formula can be considered as a general loss form used in previous methods like RKD [20] and SP [31]. If the thresholds of the teacher and student are set to be equal, the loss can be simplified as the common L2 loss.

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \|s_n(T) - s_n(S)\| \tag{8}$$

Algorithm 1: Evaluation-oriented KD

Input: The balanced input mini-batch X, the pre-trained teacher network T, the student network with random initialized parameters S, the FPR range $[FPR_L, FPR_U]$, the number of thresholds k, learning rate λ.

teacher thresholds $t(T) = [t_1(T), t_2(T), \cdots, t_k(T)] \leftarrow [0, 0, \cdots, 0]$;

student thresholds $t(S) = [t_1(S), t_2(S), \cdots, t_k(S)] \leftarrow [0, 0, \cdots, 0]$;

iteration number $i \leftarrow 0$;

while *not converged* **do**

- Obtain the features by T and S;
- Construct all the possible positive and negative pairs by Eq. 1;
- Sort the negative pair similarities and obtain the thresholds corresponding to predefined FPR range in the current mini-batch;
- Update thresholds $t(T)$ and $t(S)$ by Eq. 4;
- Compute our EKD loss $\mathcal{L}$ by Eq. 11 for positive and negative pairs, respectively;
- Compute the total loss by Eq. 12;
- Compute the gradients of S;
- Update the parameters S;
- $i \leftarrow i + 1$;

end

Output: S

However, the formulation of Eq. 7 may still be inflexible due to the absolute distance constraint of each critical relation between the teacher and student models. Given a positive or negative similarity and a chosen threshold, the comparative relations influence the TPR or FPR rather than the absolute distance. That is, if a relation meets the condition that $\mathbb{1}(s(T) - t_k(T)) = \mathbb{1}(s(S) - t_k(S))$, it will not cause the metric difference between the teacher and student models. Thus, we can directly adopt this condition to optimize the student model. Since the Indicator function is a step function whose value is 0 or 1 and the thresholds are monotonic, the loss can be formulated as follows.

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}\left\|(\sum_{k=1}^{K}\mathbb{1}(s_n(T) - t_k(T)) - \sum_{k=1}^{K}\mathbb{1}(s_n(S) - t_k(S)))\right\| \tag{9}$$

The above formulation can be considered as a constraint for the rank between a certain similarity and the thresholds. However, the Indicator function cannot be optimized with gradient-based methods. Inspired by [1], a sigmoid function $G(\cdot;\tau)$ is used to approximate the Indicator function:

$$\mathcal{G}(x_{nk}, \tau) = \frac{1}{1 + e^{\frac{-x_{nk}}{\tau}}} \tag{10}$$

where τ refers to the temperature adjusting the sharpness, and $x_{nk} = s_n - t_k$ refers to the distance between the n-th similarity and the k-th threshold. Substituting $G(\cdot;\tau)$ into Eq. 9, the loss can be approximated as:

$$\mathcal{L}_{ekd} = \frac{1}{N}\sum_{n=1}^{N}\left\|(\sum_{k=1}^{K}\mathcal{G}(x_{nk}(T), \tau) - \sum_{k=1}^{K}\mathcal{G}(x_{nk}(S), \tau))\right\| \tag{11}$$

where $x_{nk}(T) = s_n(T) - t_k(T)$ and $x_{nk}(S) = s_n(S) - t(S)$. In addition, as described in Sec. 3.1, since the number of negative pairs is much larger than the one of positive pairs, we handle the two relations separately and reduce the number of negative pairs via hard negative mining. In summary, the entire formulation of our EKD is: $\mathcal{L}_{EKD} = \lambda_1\mathcal{L}_{pos} + \lambda_2\mathcal{L}_{neg}$, where λ_1 and λ_2 are the weight parameters. Furthermore, to maintain the class discriminability, we incorporate the loss function of Arcface [6], and thus the final loss becomes:

$$\mathcal{L}(\Theta) = \mathcal{L}_{EKD} + \mathcal{L}_{Arcface}, \tag{12}$$

where Θ denotes the parameter set. The entire training process is summarized in Algorithm 1.

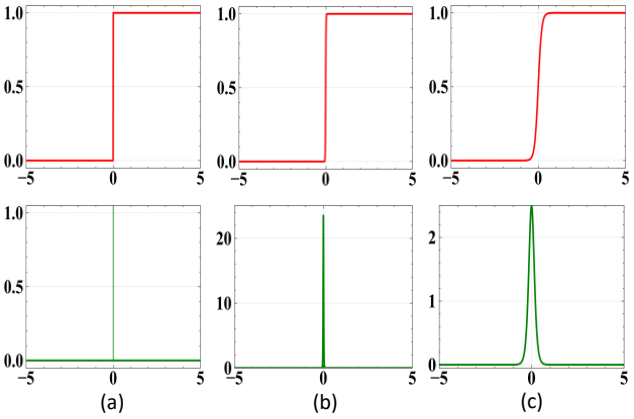


Figure 3. (Top) The Indicator function and sigmoid functions with different temperature τ as different approximations. (Bottom) The corresponding derivatives of each function. (a) Indicator function (b) sigmoid function with $\tau = 0.01$ (c) sigmoid function with $\tau = 0.1$.

Indicator Function Approximation. The derivative of the Indicator function is defined as Dirac delta function $\delta(x)$, which is either flat everywhere, with zero gradient, or discontinuous, and hence cannot be optimized with gradient based method [1]. The derivative of the sigmoid function $\mathcal{G}(x, \tau)$ is as follows:

$$\frac{\partial\mathcal{G}(x,\tau)}{\partial x} = \frac{\mathcal{G}(x,\tau)(1 - \mathcal{G}(x,\tau))}{\tau} \tag{13}$$

As shown in Fig. 3, the temperature governs the approximation tightness and the operating region to provide gradients.

4. Experiments

4.1. Datasets

Training Set. We employ refined MS1MV2 [6] as our training data for fair comparisons with other methods. MS1MV2 contains about 5.8M images of 85K individuals.

Test Set. We extensively test our method on several popular face benchmarks, including LFW [11], CFP-FP [27], CPLFW [36], AgeDB [19], CALFW [35], IJB-B [32], IJB-C [18], and MegaFace [14]. LFW is the most commonly used face verification test dataset, which contains 13233 web-collected images from 5749 different identities. The other four datasets are standard benchmarks with two variations, *i.e.*, CFP and CPLFW on pose, and AgeDB and CALFW on age. MegaFace aims at evaluating the face recognition performance at the million scales of distractors. The gallery set of MegaFace includes 1M images of 690K subjects, and the probe set includes 100K photos of 530 unique subjects from FaceScrub. The IJB-B and IJB-C are two challenging public template-based benchmarks for face recognition. The IJB-B dataset contains $1,845$ subjects with 21.8K still images and 55K frames from $7,011$ videos. The IJB-C dataset is a further extension of IJB-B, which contains about $3,500$ identities with a total of $31,334$ images and $117,542$ unconstrained video frames.

4.2. Experimental Settings

Data Processing. We follow [6] to crop the 112×112 faces with five landmarks detected by MTCNN [34]. The RGB images are first normalized by subtracting 127.5 and divided by 128, then feeding into the embedding network.

Teacher. We use Resnet50 as the teacher model, which is trained by ArcFace [6]. For all the experiments in this paper, the teacher model is pre-trained and frozen.

Student. To show our method's generality, we use two neural network structures, *e.g.*, MobileFaceNet [4] and Resnet18 [6] as the student models, respectively.

Training. We conducted all the experiments on 16 NVIDIA Tesla V100 GPU with Pytorch [22] framework. All student models are trained from scratch using the SGD algorithm for 28 epochs. The learning rate starts at 0.1 and is divided by 10 at the 10, 18, 24 epochs. The momentum is 0.9, and the weight decay is $5e-4$. The weights λ_1 and λ_2 are set to 0.02 and 0.01, respectively. For ArcFace, we follow the common setting as [6] to set scale $s = 64$ and margin $m = 0.5$. The batch size for ArcFace is set to be 512. The balanced batch size is also set to be 512, and 4 images are randomly sampled per category. To increase the number of negative pairs, we merge the two inputs when constructing the negative pairs. All the training images are horizontally flipped with a probability of 0.5 as the only data augmentation strategy.

Testing. We follow the evaluation protocol [11] to report the performance on LFW, CFP-FP, CPLFW, AgeDB and CALFW. On Megaface, both face identification and verification performance are reported. On IJB-B and IJB-C, we follow the 1:1 verification protocol in ArcFace [6] and take the *average of the image features* as the corresponding template representation without bells and whistles.

4.3. Ablation Study

Effects of Student Network Structure. We investigate the generalization capability of our method for different student network structures. Tab. 1 (Student Structure) shows the results of two structures, *i.e.*, IR18 and MobileFaceNet. Though the performance improvement on the two network structures is different, our method generally performs better than directly training the student network from scratch. Our method can bring more improvement for a student with a lower capacity (MobileFaceNet).

Effects of the Temperature τ. As described in Sec. 3.2, the temperature τ governs the smoothing of the sigmoid function used to approximate the Indicator function. Tab. 1 (Temperature τ) shows that a value of 0.01 achieves the best performance, which shares similar conclusion with [1]. As shown in Fig. 3, the value 0.01 gives a better approximation to the Indicator function than 0.1 and corresponds to a small operating region to provide gradients. Though the value of 0.001 gives a tighter approximation, it cannot provide enough large regions with the gradients.

Effects of Hard Negative Mining. As describe in Sec. 3.2, we adopt the hard negative mining strategy to reduce the number of negative pair similarities. Firstly, to investigate the influence of the negative pair numbers, we train models with the corresponding strategy (1000, 2000, 5000 negative pairs with the largest similarity are selected). The number of positive pairs in a mini-batch is about 800. Thus we try these values to keep the number of positive pairs and negative pairs comparable. The comparative results are reported in Tab. 1 (Hard negative mining). We have two observations: 1) all the strategies perform better than directly training the student (the row of MobileFaceNet), demonstrating our method's effectiveness. 2) The performance of 1000 and 2000 is similar, and 5000 is inferior to the other two. The reason may be that with the number of negative pairs increasing, the positive pairs' relative weight decreases. We choose 2000 as the default value since it achieves the best average performance. Second, we also investigate the effect of the hard negative mining strategy by replacing it with random negative selection. Comparing the results between "Random negative selection" and "Hard negative mining" in Tab. 1, our hard negative mining versions generally perform better than the random selection versions.

Table 1. Extensive ablation studies on MS1Mv2. We report the results of five small test datasets and a large scale test dataset (IJB-C). The default student network is MobileFaceNet. N denotes the number of selected negative pairs. K denotes the threshold number. TPR@FPR=$1e$-4 and TPR@FPR=$1e$-5 on IJB-C are reported.

Ablation Type	Methods (%)	LFW	CFP-FP	CPLFW	AgeDB	CALFW	IJB-C	IJB-C
	ResNet50 (Teacher)	99.80	97.63	92.50	97.92	96.05	95.16	92.66
Student Structure	MobileFaceNet	99.52	91.66	87.93	95.82	95.12	89.13	81.65
	MobileFaceNet + Ours	**99.60**	**94.33**	**89.35**	**96.48**	**95.37**	**90.48**	**84.00**
	IR18	99.67	94.60	89.97	97.33	95.70	91.96	86.01
	IR18 + Ours	**99.68**	**95.31**	**90.82**	**97.48**	**95.85**	**92.74**	**88.84**
Temperature τ	$\tau = 0.1$	99.62	93.33	88.55	96.20	95.20	89.51	82.04
	$\tau = 0.01$	99.60	**94.33**	**89.35**	**96.48**	**95.37**	**90.48**	**84.00**
	$\tau = 0.001$	**99.65**	93.29	89.07	96.17	95.28	88.63	79.07
Hard negative mining	$N = 1000$	99.57	93.66	89.28	95.94	95.33	90.29	**84.56**
	$N = 2000$	**99.60**	**94.33**	**89.35**	**96.48**	**95.37**	**90.48**	84.00
	$N = 5000$	99.58	93.74	88.93	96.35	95.30	89.85	82.93
Random negative selection	$N = 1000$	99.53	94.04	89.00	96.36	95.10	89.71	83.09
	$N = 2000$	99.55	94.19	89.00	96.27	95.33	89.41	82.35
	$N = 5000$	99.53	94.17	89.38	96.15	95.51	89.73	83.02
Threshold Number	$K = 3$	99.53	93.57	88.93	96.05	**95.47**	89.71	83.22
	$K = 6$	**99.60**	**94.33**	**89.35**	**96.48**	95.37	**90.48**	**84.00**
Loss function	$Eq.$ 7	99.53	91.99	88.23	96.17	94.88	89.35	81.68
	$Eq.$ 11	**99.60**	**94.33**	**89.35**	**96.48**	**95.37**	**90.48**	**84.00**

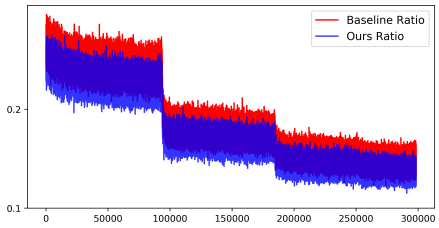


Figure 4. Ratio change between the number of critical relations and total relations during training in baseline and ours.

Effects of Thresholds Number. Given a concerned FPR range $[FPR_L, FPR_U]$, the number of thresholds depends on how the FPR is spaced. In general, a vector of thresholds corresponding to FPR evenly spaced on a logarithmic scale is chosen. For an FPR range [$1e$-1, $1e$-6], the typical number of thresholds is 6. Here, we compare two values, *i.e.*, 3, 6. The 3 thresholds are set to be as $1e$-1, $1e$-3, and $1e$-6, respectively. As shown in Tab. 1, the results of 6 thresholds are better than 3, since more thresholds can more finely describe the relations between similarities.

Effects of Loss Function. To investigate the effect of our relaxed loss function, we train models with Eq. 7 and Eq. 11, respectively. Comparing the results in Tab. 1 (Loss Function), the model trained with Eq. 11 outperforms the version with Eq. 7, which demonstrates that giving more flexibility to the student is beneficial.

Ratio between the Number of Critical Relations and Total Relations. Fig. 4 shows the ratios between the number of critical relations and total relations, which are calculated during training in the baseline and our method, respectively. Since the student network is trained from scratch, the ratios at the very early training steps fluctuate wildly, and thus we remove the beginning training steps to make the figure clear. The number of critical relations trained by our method is smaller than the baseline, demonstrating that our approach does reduce the performance gap between the teacher and student models during training.

4.4. Comparisons with SOTA Methods

We compare with a wide variety of SOTA KD methods, including the methods proposed for other tasks (FitNet [24], KD [10], DarkRank [5], SP [31], CCKD [20] and RKD [20]) and specifically designed methods for face recognition (ShrinkTeaNet [8], Triplet Distillation [9] and MarginDistillation [29]). Since the former six methods do not conduct complete experiments on face recognition, we re-implement them following their original papers. We cite the results of the latter three methods from [29].

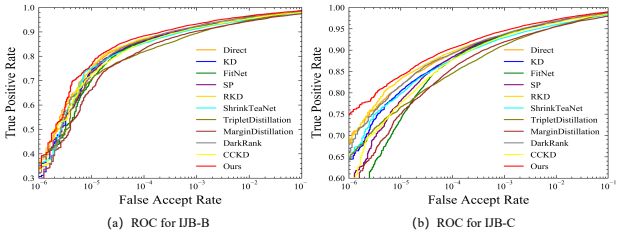
Results on LFW, CFP-FP, CPLFW, AgeDB and CALFW. Tab. 2 shows the results compare with the SOTA competitors on five common small benchmarks. From the Tab. 2, most of the knowledge distillation methods are better than directly training the student network from scratch (*i.e.*, MobileFaceNet), but the performance improvement is limited. Among all the competitors, relation-based methods seem to show better performance than the instance-based methods, while are inferior to MarginDistillation. Although we cannot beat the competitors on each test set, we achieve the best average performance on theses test sets.

Table 2. Verification comparison with SOTA methods on LFW, two pose benchmarks: CFP-FP and CPLFW, and two age benchmarks: AgeDB and CALFW.

Methods (%)	LFW	CFP-FP	CPLFW	AgeDB	CALFW
ResNet50	99.80	97.63	92.50	97.92	96.05
MobileFaceNet	99.52	91.66	87.93	95.82	95.12
FitNet (arxiv'14)	99.47	91.30	88.30	96.18	95.12
KD (NIPSW'14)	99.50	91.71	87.85	95.93	95.03
DarkRank (AAAI'18)	99.55	91.84	87.77	95.60	95.07
SP (ICCV' 19)	99.53	92.33	88.45	96.17	95.07
CCKD (ICCV' 19)	99.47	91.90	88.48	95.83	95.22
RKD (CVPR'19)	99.58	92.13	87.97	96.18	95.25
ShrinkTeaNet (arxiv'19)	99.47	91.97	88.52	96.00	94.98
TripletDistillation (ICIP'20)	99.55	93.14	88.03	95.53	94.97
MarginDistillation (arxiv'20)	**99.61**	92.01	88.03	**96.55**	95.13
EKD (Ours)	99.60	**94.33**	**89.35**	96.48	**95.37**

Table 3. 1:1 verification performance (TPR) on the IJB-B and IJB-C datasets.

Methods (%)	IJB-C (FPR)		IJB-B (FPR)	
	$1e{-}4$	$1e{-}5$	$1e{-}4$	$1e{-}5$
ResNet50	95.16	92.66	93.45	88.65
MobileFaceNet	89.13	81.65	87.07	74.63
FitNet (arxiv'14)	87.76	73.71	86.35	70.19
KD (NIPSW'14)	88.37	80.39	86.08	74.30
DarkRank (AAAI'18)	89.28	81.62	86.76	73.75
SP (ICCV' 19)	88.43	78.13	86.34	72.85
CCKD (ICCV' 19)	87.99	78.75	85.63	72.38
RKD (CVPR'19)	89.65	83.21	87.27	75.17
ShrinkTeaNet (arxiv'19)	87.80	79.78	85.31	75.23
TripletDistillation (ICIP'20)	84.57	76.65	81.88	70.51
MarginDistillation (arxiv'20)	85.71	75.00	82.97	66.25
EKD (Ours)	**90.48**	**84.00**	**88.35**	**76.60**



(a) ROC for IJB-B (b) ROC for IJB-C

Figure 5. ROC curves of 1:1 verification protocol on the IJB-B and IJB-C dataset.

Results on IJB-B and IJB-C. In Tab. 3, we compare the 1:1 verification TPR@FPR=$1e$-4 and TPR@FPR=$1e$-5 with the previous SOTA methods on the IJB-B and IJB-C datasets. Surprisingly, unlike the small test dataset results, most of the knowledge distillation methods bring a little performance improvement or even worse than the baseline on these two large scale test datasets. Though RKD shows better generalization ability than others, our method again achieves the best performance. Fig. 5 shows the full ROC curves of our method and other SOTA competitors, and it is clear that our method performs best.

Results on MegaFace. Finally, we evaluate the performance on the MegaFace Challenge. We also report the results following the ArcFace testing protocol, which refines both the probe set and the gallery set. As shown in Tab. 4, most of the competitors achieve better performance than baseline, and our method achieves the best verification performance, surpassing all the other strong competitors. Our method performs slightly inferior to the others on the rank-1 metric. The reason may be that our method adopts the TPR and FPR as the performance indicator during training and overlooks the top1 performance.

Table 4. Verification comparison with SOTA methods on MegaFace Challenge 1 using FaceScrub as the probe set. "Id" refers to the rank-1 face identification accuracy with 1M distractors, and "Ver" refers to the face verification TPR at $1e$-6 FPR. The column "R" refers to data refinement on both probe set and 1M distractors.

Methods (%)	Id (R)	Ver (R)	Id	Ver
ResNet50	98.14	98.34	80.62	96.83
MobileFaceNet	90.91	92.71	75.52	90.80
FitNet (arxiv'14)	91.16	92.34	75.88	90.64
KD (NIPSW'14)	90.40	92.00	75.81	90.07
DarkRank (AAAI'18)	90.76	92.41	75.80	90.66
SP (ICCV' 19)	91.25	92.41	75.37	90.62
CCKD (ICCV' 19)	91.17	92.76	75.73	90.63
RKD (CVPR'19)	91.44	92.92	75.73	91.21
ShrinkTeaNet (arxiv'19)	90.73	92.32	75.55	90.56
Triplet Distillation (ICIP'20)	86.52	88.75	71.93	91.35
MarginDistillation (arxiv'20)	**91.70**	92.96	**76.34**	91.31
EKD (Ours)	91.02	**93.08**	75.54	**91.42**

Table 5. Training time for each batch under the same experiment setting.

Methods	Baseline	RKD	Ours
Time (s)	0.068	0.147	0.129

Time Complexity. As shown in Tab. 5, though our method brings some burden on training complexity compared with directly training the small network without the teacher, our method has lower complexity compared with RKD, which is also a relation-based KD method.

5. Conclusions

In this paper, we propose a novel evaluation-oriented KD method for deep face recognition. Different from previous KD methods requiring the student to mimic the teacher's behavior completely in the representation space, our EKD optimizes the student to directly reduce the performance gap between the teacher and student models during training. The most commonly used evaluation metrics in face recognition, *i.e.*, False Positive Rate (FPR), and True Positive Rate (TPR), are adopted as the performance indicator. Extensive experiments on popular face recognition benchmarks have demonstrated our method's effectiveness and generalization capability. In subsequent work, we can try to improve the TOP1 performance of our method with more suitable performance evaluation metrics.

References

[1] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. *arXiv preprint arXiv:2007.12163*, 2020. 5, 6

[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 1

[3] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Int. Conf. Machine. Learning.*, pages 129–136, 2007. 3

[4] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Chinese Conf. on Biometric Recog.*, 2018. 1, 6

[5] Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. *arXiv preprint arXiv:1707.01220*, 2017. 3, 7

[6] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 2, 4, 5, 6

[7] Jiankang Deng, Jia Guo, Debing Zhang, Yafeng Deng, Xiangju Lu, and Song Shi. Lightweight face recognition challenge. In *Int. Conf. Comput. Vis. Worksh.*, pages 0–0, 2019. 2

[8] Chi Nhan Duong, Khoa Luu, Kha Gia Quach, and Ngan Le. Shrinkteanet: Million-scale lightweight face recognition via shrinking teacher-student networks. *arXiv preprint arXiv:1905.10620*, 2019. 3, 7

[9] Yushu Feng, Huan Wang, Haoji Roland Hu, Lu Yu, Wei Wang, and Shiyan Wang. Triplet distillation for deep face recognition. In *ICIP*, pages 808–812. IEEE, 2020. 3, 7

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Adv. Neural Inform. Process. Syst. Worksh.*, 2014. 1, 2, 7

[11] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 6

[12] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *CVPR*, pages 5901–5910, 2020. 2, 4

[13] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017. 1

[14] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016. 4, 6

[15] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *AAAI*, 2019. 4

[16] Weiyang Li, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 2

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 3

[18] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *ICB*, 2018. 4, 6

[19] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *CVPRW*, 2017. 6

[20] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019. 2, 3, 4, 7

[21] O.M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015. 2

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Adv. Neural Inform. Process. Syst. Worksh.*, 2017. 6

[23] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *CVPR*, pages 5007–5016, 2019. 2, 3

[24] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 1, 3, 7

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 1

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[27] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D.W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016. 6

[28] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NeurIPS*, 2014. 2

[29] David Svitov and Sergey Alyamkin. Margindistillation: distillation for margin-based softmax. *arXiv preprint arXiv:2003.02586*, 2020. 3, 7

[30] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014. 2

[31] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *CVPR*, pages 1365–1374, 2019. 2, 3, 4, 7

[32] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka,

Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *CVPRW*, 2017. 4, 6

[33] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 1

[34] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Sign. Process. Letters*, 23(10):1499–1503, 2016. 6

[35] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv:1708.08197*, 2017. 6

[36] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Technical Report 18-01, Beijing University of Posts and Telecommunications, February 2018. 6