

# ArcFace: Additive Angular Margin Loss for Deep Face Recognition

Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou

**Abstract**—Recently, a popular line of research in face recognition is adopting margins in the well-established softmax loss function to maximize class separability. In this paper, we first introduce an Additive Angular Margin Loss (ArcFace), which not only has a clear geometric interpretation but also significantly enhances the discriminative power. Since ArcFace is susceptible to the massive label noise, we further propose sub-center ArcFace, in which each class contains  $K$  sub-centers and training samples only need to be close to any of the  $K$  positive sub-centers. Sub-center ArcFace encourages one dominant sub-class that contains the majority of clean faces and non-dominant sub-classes that include hard or noisy faces. Based on this self-propelled isolation, we boost the performance through automatically purifying raw web faces under massive real-world noise. Besides discriminative feature embedding, we also explore the inverse problem, mapping feature vectors to face images. Without training any additional generator or discriminator, the pre-trained ArcFace model can generate identity-preserved face images for both subjects inside and outside the training data only by using the network gradient and Batch Normalization (BN) priors. Extensive experiments demonstrate that ArcFace can enhance the discriminative feature embedding as well as strengthen the generative face synthesis.

**Index Terms**—Large-scale Face Recognition, Additive Angular Margin, Noisy Labels, Sub-class, Model Inversion

arXiv:1801.07698v4 [cs.CV] 4 Sep 2022

## 1 INTRODUCTION

FACE representation using DCNN embedding is the method of choice for face recognition [1], [2], [3], [4], [5], [6]. DCNNs map the face image, typically after a pose normalization step [7], [8], into a feature that should have small intra-class and large inter-class distance. There are two main lines of research to train DCNNs for face recognition. Some train a multi-class classifier which can separate different identities in the training set, such as by using a softmax classifier [2], [4], [9], [10], [11], and the others learn directly an embedding, such as the triplet loss [3]. Based on the large-scale training data and the elaborate DCNN architectures, both the softmax-loss-based methods [9] and the triplet-loss-based methods [3] can obtain excellent performance on face recognition. However, both the softmax loss and the triplet loss have some drawbacks. For the softmax loss: (1) the learned features are separable for the closed-set classification problem but not discriminative enough for the open-set face recognition problem; (2) the size of the linear transformation matrix  $W \in \mathbb{R}^{d \times N}$  increases linearly with the identities number  $N$ . For the triplet loss: (1) there is a combinatorial explosion in the number of face triplets especially for large-scale datasets, leading to a significant increase in the number of iteration steps; (2) semi-hard sample mining is a quite difficult problem for effective model training.

To adopt margin benefit but avoid the sampling problem in the Triplet loss [3], recent methods [13], [14], [15] focus on incorporating margin penalty into a more feasible framework, the softmax loss, which has global sample-to-class comparisons

within the multiplication step between the embedding feature and the linear transformation matrix. Naturally, each column of the linear transformation matrix is viewed as a class center representing a certain class. Sphereface [13] introduces the important idea of angular margin, however their loss function requires a series of approximations, which results in an unstable training of the network. In order to stabilize training, they propose a hybrid loss function which includes the standard softmax loss. Empirically, the softmax loss dominates the training process, because the integer-based multiplicative angular margin makes the target logit curve very precipitous and thus hinders convergence.

In this paper, we propose an Additive Angular Margin loss [16] to stabilize the training process and further improve the discriminative power of the face recognition model. More specifically, the dot product between the DCNN feature and the last fully connected layer is equal to the cosine distance after feature and center normalization. We utilize the arc-cosine function to calculate the angle between the current feature and the target center. Afterwards, we introduce an additive angular margin to the target angle, and we get the target logit back again by the cosine function. Then, we re-scale all logits by a fixed feature norm, and the subsequent steps are exactly the same as in the softmax loss. Due to the exact correspondence between the angle and arc in the normalized hypersphere, our method can directly optimize the geodesic distance margin, thus we call it ArcFace.

Even though impressive performance has been achieved by the margin-based softmax methods [17], [13], [14], [15], they all need to be trained on well-annotated clean datasets [18], which require intensive human efforts. Wang et al. [18] found that faces with label noise significantly degenerate the recognition accuracy and manually built a high-quality dataset including 1.7M images of 59K celebrities. However, it took 50 annotators to work continuously for one month to clean the dataset, which further demonstrates the difficulty of obtaining a large-scale clean dataset for face recognition. Since accurate manual annotations can be

- Corresponding author: Jiankang Deng, E-mail: j.deng16@imperial.ac.uk  
J. Deng, N. Xue and S. Zafeiriou are with the Department of Computing, Imperial College London, UK. J. Guo is with InsightFace. J. Yang is with Department of Computer Science, University of Nottingham. I. Kotsia is with Cogitat.

Manuscript received on November 28, 2020; revised on May 4, 2021; accepted on June 4, 2021.

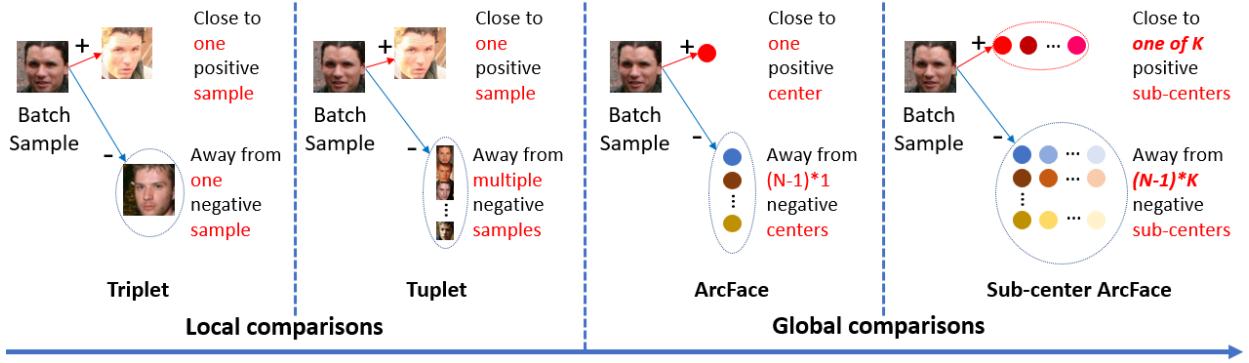


Fig. 1. Comparisons of Triplet [3], Tuplet [12], ArcFace and sub-center ArcFace. Triplet and Tuplet conduct local sample-to-sample comparisons with Euclidean margins within the mini-batch. By contrast, ArcFace and sub-center ArcFace conduct global sample-to-class and sample-to-subclass comparisons with angular margins.

expensive [18], learning with massive noisy data has recently attracted much attention [19], [20], [21]. However, computing time-varying weights for samples [19] or designing piece-wise loss functions [20] according to the current model’s predictions can only alleviate the influence from noisy data to some extent as the robustness and improvement depend on the initial performance of the model. Besides, the co-mining method [21] requires to train twin networks together thus it is less practical for training large models on large-scale datasets.

To improve the robustness under massive real-world noise, we relax the intra-class constraint of forcing all samples close to the corresponding positive centers by introducing sub-classes into ArcFace [22]. As illustrated in Figure 1, we design  $K$  sub-centers for each class and the training sample only needs to be close to any of the  $K$  positive sub-centers instead of the only one positive center. If a training face is a noisy sample, it does not belong to the corresponding positive class. In ArcFace, this noisy sample generates a large wrong loss value, which impairs the model training. In sub-center ArcFace, the intra-class constraint enforces the training sample to be close to one of the multiple positive sub-centers but not all of them. The noise is likely to form a non-dominant sub-class and will not be enforced into the dominant sub-class. Therefore, sub-center ArcFace is more robust to noise. In our experiments, we find the proposed sub-center ArcFace can encourage one dominant sub-class that contains the majority clean faces and multiple non-dominant sub-classes that include hard or noisy faces. This automatic isolation can be directly employed to clean the training data through dropping non-dominant sub-centers and high-confident noisy samples. Based on the proposed sub-center ArcFace, we can automatically obtain large-scale clean training data from raw web face images to further improve the discriminative power of the face recognition model.

In Figure 1, we compare the differences between Triplet [3], Tuplet [12], ArcFace and sub-center ArcFace. Triplet loss [3] only considers local sample-to-sample comparisons with Euclidean margins within the mini-batch. Tuplet loss [12] further enhances the comparisons by using all of the negative pairs within the mini-batch. By contrast, the proposed ArcFace and sub-center ArcFace conduct global sample-to-class and sample-to-subclass comparisons with angular margins.

As the proposed ArcFace is effective for the mapping from the face image to the discriminative feature embedding, we are also interested in the inverse problem: the mapping from a low-dimensional latent space to a highly nonlinear face space. Syn-

thesizing face images [23], [24], [25], [26], [27], [28], [29] has recently brought much attention from the community. DeepDream [30] is proposed to transform a random input to yield a high output activation for a chosen class by employing the gradient from the pre-trained classification model and some regularizers (e.g. total variance [31] for maintaining piece-wise constant patches). Even though DeepDream can keep the selected output response high to preserve identity, the resulting faces are not realistic, lacking natural face statistics. Inspired by the pioneer generative face recognition model (Eigenface [32]) and recent data-free methods [33], [34], [35] for restoring ImageNet images, we employ the statistic prior (e.g. mean and variance stored in the BN layers) to constrain the face generation. In this paper, we show that the proposed ArcFace can also enhance the generative power. Without training any additional generator or discriminator like in Generative Adversarial Networks (GANs) [36], the pre-trained ArcFace model can generate identity-preserved and visually reasonable face images only by using the gradient and BN priors.

The advantages of the proposed methods can be summarized as follows:

**Intuitive.** ArcFace directly optimizes the geodesic distance margin by virtue of the exact correspondence between the angle and arc in the normalized hypersphere. The proposed additive angular margin loss can intuitively enhance the intra-class compactness and inter-class discrepancy during discriminative learning of face feature embedding.

**Economical.** We introduce sub-class into ArcFace to improve its robustness under massive real-world noises. The proposed sub-center ArcFace can automatically clean the large-scale raw web faces (e.g. MS1MV0 [37] and Celeb500K [38]) without expensive and intensive human efforts. The automatically cleaned training data, named IBUG-500K, has been released to facilitate future research.

**Easy.** ArcFace only needs several lines of code and is extremely easy to implement in the computational-graph-based deep learning frameworks, e.g. MxNet [39], Pytorch [40] and Tensorflow [41]. Furthermore, contrary to the works in [13], [42], ArcFace does not need to be combined with other loss functions in order to have stable convergence.

**Efficient.** ArcFace only adds negligible computational complexity during training. The proposed center parallel strategy can easily support millions of identities for training on a single server (8 GPUs).

**Effective.** Using IBUG-500K as the training data, ArcFace

achieves state-of-the-art performance on ten face recognition benchmarks including large-scale image and video datasets collected by us. Impressively, our model reaches 97.27% TPR@FPR=1e-4 on IJB-C. Code and pre-trained models have been made available.

**Engaging.** ArcFace can not only enhance the discriminative power but also strengthen the generative power. By accessing the network gradient and employing the statistic priors stored in the BN layers, the pre-trained ArcFace model can restore identity-preserved and visually plausible face images for both subjects inside and outside the training data.

## 2 RELATED WORK

**Face Recognition with Margin Penalty.** As shown in Figure 1, the pioneering work [3] uses the Triplet loss to exploit triplet data such that faces from the same class are closer than faces from different classes by a clear Euclidean distance margin. Even though the Triplet loss makes perfect sense for face recognition, the sample-to-sample comparisons are local within mini-batch and the training procedure for the Triplet loss is very challenging as there is a combinatorial explosion in the number of triplets especially for large-scale datasets, requiring effective sampling strategies to select informative mini-batch [43], [3] and choose representative triplets within the mini-batch [44], [12]. As the Triplet loss trained with semi-hard negative mining converges slower due to the ignorance of too many examples, a double-margin contrastive loss is proposed in [45] to explore more informative and stable examples by distance weighted sampling, thus it converges faster and more accurately. Some other works tried to reduce the total number of triplets with proxies [46], [47], i.e., sample-to-sample comparison is changed into sample-to-proxy comparison. However, sampling and proxy methods only optimize the embedding of partial classes instead of all classes in one iteration step.

Margin-based softmax methods [13], [17], [14], [15] focused on incorporating margin penalty into a more feasible framework, softmax loss, which has extensive sample-to-class comparisons. Compared to deep metric learning methods (e.g., Triplet [3], Tuplet [44], [12]), margin-based softmax methods conduct global comparisons at the cost of memory consumption on holding the center of each class as illustrated in Figure 1. Sample-to-class comparison is more efficient and stable than sample-to-sample comparison as (1) the class number is much smaller than sample number, and (2) each class can be represented by a smoothed center vector which can be updated online during training. To further improve the margin-based softmax loss, recent works focus on the exploration of adaptive parameters [48], [49], [50], inter-class regularization [51], [52], mining [53], [54], grouping [55], etc.

**Face Recognition under Noise.** Most of the face recognition datasets [56], [37], [9], [38] are downloaded from the Internet by searching a pre-defined celebrity list, and the original labels are likely to be ambiguous and inaccurate [18]. Learning with massive noisy data has recently drawn much attention in face recognition [57], [19], [20], [21] as accurate manual annotations can be expensive [18] or even unavailable.

Wu et al. [57] proposed a semantic bootstrap strategy, which re-labels the noisy samples according to the probabilities of the softmax function. However, automatic cleaning by the bootstrapping rule requires time-consuming iterations (e.g. twice refinement

steps are used in [57]) and the labelling quality is affected by the capacity of the original model. Hu et al. [19] found that the cleanliness possibility of a sample can be dynamically reflected by its position in the target logit distribution and presented a noise-tolerant end-to-end paradigm by employing the idea of weighting training samples. Zhong et al. [20] devised a noise-resistant loss by introducing a hypothetical training label, which is a convex combination of the original label with probability  $\rho$  and the predicted label by the current model with probability  $1 - \rho$ . However, computing time-varying fusion weight [19] and designing piece-wise loss [20] contain many hand-designed hyper-parameters. Besides, re-weighting methods are susceptible to the performance of the initial model. Wang et al. [21] proposed a co-mining strategy which uses the loss values as the cue to simultaneously detect noisy labels, exchange the high-confidence clean faces to alleviate the error accumulation caused by the sampling bias, and re-weight the predicted clean faces to make them dominate the discriminative model training. However, the co-mining method requires training twin networks simultaneously and it is challenging to train large networks (e.g. ResNet100 [58]) on a large-scale dataset (e.g. MS1MV0 [37] and Celeb500K [38]). **Face Recognition with Sub-classes.** Practices and theories that lead to “sub-class” have been studied for a long time [59], [60]. The concept of “sub-class” applied in face recognition was first introduced in [59], [60], where a mixture of Gaussians was used to approximate the underlying distribution of each class. For instance, a person’s face images may be frontal view or side view, resulting in different modalities when all images are represented in the same data space. In [59], [60], experimental results showed that subclass divisions can be used to effectively adapt to different face modalities thus improve the performance of face recognition. Wan et al. [61] further proposed a separability criterion to divide every class into sub-classes, which have much less overlaps. The new within-class scatter can represent multi-modality information, therefore optimizing this within-class scatter will separate different modalities more clearly and further increase the accuracy of face recognition. However, these work [59], [60], [61] only employed hand-designed feature descriptor on tiny under-controlled datasets.

Concurrent with our work, Softtriple [62] presents a multi-center softmax loss with class-wise regularizer. These multi-centers can depict the hidden distribution of the data [63] due to the fact that they can capture the complex geometry of the original data and help reduce the intra-class variance. On the fine-grained visual retrieval problem, the Softtriple [62] loss achieves better performance than the softmax loss as capturing local clusters is essential for this task. Even though the concept of “sub-class” has been employed in face recognition [59], [60], [61] and fine-grained visual retrieval [62], none of these work has considered the large-scale (e.g. 0.5 million classes) face recognition problem under massive noise (e.g. around 50% noisy samples within the training data).

**Face Synthesis by Model Inversion.** Identity-preserving face generation [64], [65], [66], [29] has been extensively explored under the framework of GAN [36]. Even though GAN models can yield high-fidelity images [67], [68], training a GAN’s generator requires access to the original data. Due to the emerging concern of data privacy, an alternative line of work in security focuses on model inversion, that is, image synthesis from a single CNN. Model inversion can not only help researchers to visualize neural networks to understand their properties [69] but also can be used

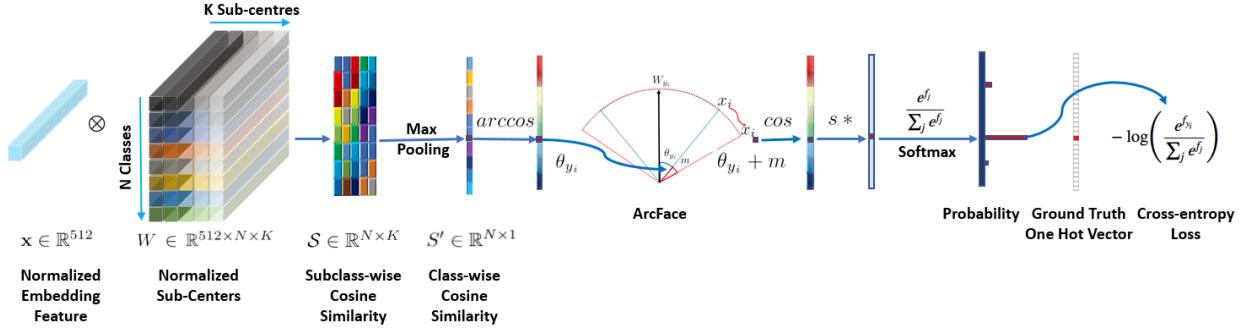


Fig. 2. Training the deep face recognition model by the proposed ArcFace loss ( $K=1$ ) and sub-center ArcFace loss (e.g.  $K=3$ ). Based on a  $\ell_2$  normalization step on both embedding feature  $x_i \in \mathbb{R}^{512}$  and all sub-centers  $W \in \mathbb{R}^{512 \times N \times K}$ , we get the subclass-wise similarity score  $S \in \mathbb{R}^{N \times K}$  by a matrix multiplication  $W^T x_i$ . After a max pooling step, we can easily get the class-wise similarity score  $S' \in \mathbb{R}^{N \times 1}$ . Afterwards, we calculate the  $\arccos \theta_{y_i}$  and get the angle between the feature  $x_i$  and the ground truth center  $W_{y_i}$ . Then, we add an angular margin penalty  $m$  on the target (ground truth) angle  $\theta_{y_i}$ . After that, we calculate  $\cos(\theta_{y_i} + m)$  and multiply all logits by the feature scale  $s$ . Finally, the logits go through the softmax function and contribute to the cross entropy loss.

for data-free distillation, quantization and pruning [33], [34], [35]. Fredrikson et al. [70] propose the model inversion attack to obtain class images from a network through a gradient descent on the input. As the pixel space is so large compared to the feature space, optimizing the image pixels by gradient descent [31] requires heavy regularization terms, such as total variation [31] or Gaussian blur [71]. Even though previous model inversion methods [70], [30] can transform an input image (random noise or a natural image) to yield a high output activation for a chosen class, it leaves intermediate representations constraint-free. Therefore, the resulting images are not realistic, lacking natural image statistics.

The pioneer generative face recognition model is Eigenface [32], which can project a training face image or a new face image (mean-subtracted) on the eigenfaces and thereby record how that face differs from the mean face. The eigenvalue associated with each eigenface represents how much the image vary from the mean image in that direction. The recognition process with the eigenface method is to project query images into the face-space spanned by eigenfaces calculated, and to find the closest match to a face class in that face-space. Even though raw pixel features used in Eigenface are substituted by the deep convolutional features, the procedure of employing the statistic prior (e.g. mean and variance) to reconstruct face images can be an inspiration. Recently, [33], [34], [35] have proposed a data-free method employing the statistics (e.g. mean and variance) stored in the BN layers to restore ImageNet images. Inspired by these works, we synthesize face images by inverting the pre-trained ArcFace model and considering the face prior (e.g. mean and variance) stored in the BN layers.

### 3 PROPOSED APPROACH

#### 3.1 ArcFace

The most widely used classification loss function, softmax loss, is presented as follows:

$$L_1 = -\log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^N e^{W_j^T x_i + b_j}}, \quad (1)$$

where  $x_i \in \mathbb{R}^d$  denotes the deep feature of the  $i$ -th sample, belonging to the  $y_i$ -th class. The embedding feature dimension  $d$  is set to 512 in this paper following [72], [73], [13], [14].  $W_j \in \mathbb{R}^d$  denotes the  $j$ -th column of the weight  $W \in \mathbb{R}^{d \times N}$ ,  $b_j \in \mathbb{R}^N$  is the bias term, and the class number is  $N$ . Traditional

softmax loss is widely used in deep face recognition [4], [9]. However, the softmax loss function does not explicitly optimize the feature embedding to enforce higher similarity for intra-class samples and diversity for inter-class samples, which results in a performance degeneration for deep face recognition under large intra-class appearance variations (e.g. pose variations [74], [75] and age gaps [76], [77]) and large-scale test scenarios [78], [79], [80].

For simplicity, we fix the bias  $b_j = 0$  as in [13]. Then, we transform the logit [81] as  $W_j^T x_i = \|W_j\| \|x_i\| \cos \theta_j$ , where  $\theta_j$  is the angle between the weight  $W_j$  and the feature  $x_i$ . Following [13], [14], [82], we fix the individual weight  $\|W_j\| = 1$  by  $\ell_2$  normalization. Following [83], [14], [82], [15], we also fix the embedding feature  $\|x_i\|$  by  $\ell_2$  normalization and re-scale it to  $s$ . The normalization step on features and weights makes the predictions only depend on the angle between the feature and the weight. The learned embedding features are thus distributed on a hypersphere with a radius of  $s$ .

$$L_2 = -\log \frac{e^{s \cos \theta_{y_i}}}{e^{s \cos \theta_{y_i}} + \sum_{j=1, j \neq y_i}^N e^{s \cos \theta_j}}. \quad (2)$$

Since the embedding features are distributed around each feature center on the hypersphere, we employ an additive angular margin penalty  $m$  between  $x_i$  and  $W_{y_i}$  to simultaneously enhance the intra-class compactness and inter-class discrepancy as illustrated in Figure 2. Since the proposed additive angular margin penalty is equal to the geodesic distance margin penalty in the normalized hypersphere, we name our method as ArcFace.

$$L_3 = -\log \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j=1, j \neq y_i}^N e^{s \cos \theta_j}}. \quad (3)$$

We select face images from 8 different identities containing enough samples (around 1,500 images/class) to train 2-D feature embedding networks with the Norm-Softmax and ArcFace loss, respectively. As illustrated in Figure 3, all face features are pushed to the arc space with a fixed radius based on the feature normalization. The Norm-Softmax loss provides roughly separable feature embedding but produces noticeable ambiguity in decision boundaries, while the proposed ArcFace loss can obviously enforce a more evident margin between the nearest classes.

**Numerical Similarity.** In SphereFace [13], [42], ArcFace, and CosFace [14], [15], three different kinds of margin penalty are

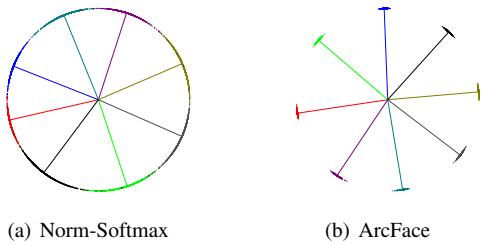


Fig. 3. Toy examples under the Norm-Softmax and ArcFace loss on 8 identities with 2D features. Dots indicate samples and lines refer to the center direction of each identity. Based on the feature normalization, all face features are pushed to the arc space with a fixed radius. The geodesic distance margin between closest classes becomes evident as the additive angular margin penalty is incorporated.

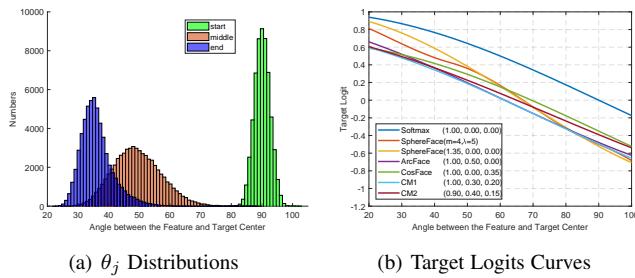


Fig. 4. Target logit analysis. (a)  $\theta_j$  distributions from start to end during ArcFace training. (2) Target logit curves for softmax, SphereFace, ArcFace, CosFace and combined margin penalty ( $\cos(m_1\theta + m_2) - m_3$ ).

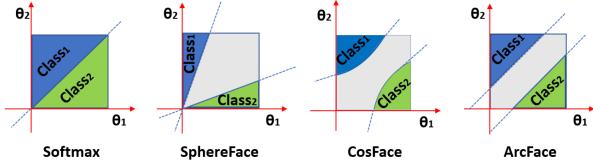


Fig. 5. Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.

proposed, e.g. multiplicative angular margin  $m_1$ , additive angular margin  $m_2$ , and additive cosine margin  $m_3$ , respectively. From the view of numerical analysis, different margin penalties, no matter add on the angle [13] or cosine space [14], all enforce the intra-class compactness and inter-class diversity by penalizing the target logit [81]. In Figure 4(b), we plot the target logit curves of SphereFace, ArcFace and CosFace under their best margin settings. We only show these target logit curves within  $[20^\circ, 100^\circ]$  because the angles between  $W_{y_i}$  and  $x_i$  start from around  $90^\circ$  (random initialization) and end at around  $30^\circ$  during ArcFace training as shown in Figure 4(a). Intuitively, there are three numerical factors in the target logit curves that affect the performance, i.e. the starting point, the end point and the slope.

By combining all of the margin penalties, we implement SphereFace, ArcFace and CosFace in a united framework with  $m_1$ ,  $m_2$  and  $m_3$  as the hyper-parameters.

$$L_4 = -\log \frac{e^{s(\cos(m_1\theta_{y_i} + m_2) - m_3)}}{e^{s(\cos(m_1\theta_{y_i} + m_2) - m_3)} + \sum_{j=1, j \neq y_i}^N e^{s \cos \theta_j}}. \quad (4)$$

As shown in Figure 4(b), by combining all of the above-mentioned margins ( $\cos(m_1\theta + m_2) - m_3$ ), we can easily get some other target logit curves which also achieve high performance.

**Geometric Difference.** Despite the numerical similarity between ArcFace and previous works, the proposed additive angular margin

has a better geometric attribute as the angular margin has the exact correspondence to the geodesic distance. As illustrated in Figure 5, we compare the decision boundaries under the binary classification case. The proposed ArcFace has a constant linear angular margin throughout the whole interval. By contrast, SphereFace and CosFace only have a nonlinear angular margin.

The minor difference in margin designs can have a significant influence on model training. For example, the original SphereFace [13] employs an annealing optimization strategy. To avoid divergence at the beginning of training, joint supervision from softmax is used in SphereFace to weaken the multiplicative integer margin penalty. We implement a new version of SphereFace without the integer requirement on the margin by employing the arc-cosine function instead of using the complex double angle formula. In our implementation, we find that  $m = 1.35$  can obtain similar performance compared to the original SphereFace without any convergence difficulty.

**Other Intra and Inter Losses.** Other loss functions can be designed based on the angular representation of features and centers. For examples, we can design a loss to enforce intra-class compactness and inter-class discrepancy on the hypersphere.

Intra-Loss is designed to improve the intra-class compactness by decreasing the angle/arc between the sample and the ground truth center.

$$L_5 = L_2 + \frac{1}{\pi} \theta_{y_i}. \quad (5)$$

Inter-Loss targets at enhancing inter-class discrepancy by increasing the angle/arc between different centers.

$$L_6 = L_2 - \frac{1}{\pi(N-1)} \sum_{j=1, j \neq y_i}^N \arccos(W_{y_i}^T W_j). \quad (6)$$

To enhance inter-class separability, RegularFace [51] explicitly distances identities by penalizing the angle between an identity and its nearest neighbor, while Minimum Hyper-spherical Energy (MHE) [84] encourages the angular diversity of neuron weights inspired by the Thomson problem. Recently, fixed classifier methods [85], [86], [87] exhibit little or no reduction in classification performance while allowing a noticeable reduction in computational complexity, trainable parameters and communication cost. In these methods, inter-class separability is not learned but inherited from a pre-defined high-dimensional geometry [87].

Triplet-loss aims at enlarging the angle/arc margin between triplet samples. In FaceNet [3], Euclidean margin is applied on the normalized features. Here, we employ the triplet-loss by the angular representation of our features as  $\arccos(x_i^{pos} x_i) + m \leq \arccos(x_i^{neg} x_i)$ .

### 3.2 Sub-center ArcFace

Even though ArcFace has shown its power in efficient and effective face feature embedding, this method assumes that training data are clean. However, this is not true especially when the dataset is in large scale. How to enable the margin-based softmax loss to be robust to noise is one of the main challenges impeding the development of face recognition [18]. In this paper, we address this problem by proposing the idea of using sub-classes for each identity, which can be directly adopted by ArcFace and will significantly increase its robustness.

As illustrated in Figure 2, we set  $K$  sub-centers for each identity. Based on a  $\ell_2$  normalization step on both embedding feature  $\mathbf{x}_i \in \mathbb{R}^{512}$  and all sub-centers  $W \in \mathbb{R}^{512 \times N \times K}$ , we

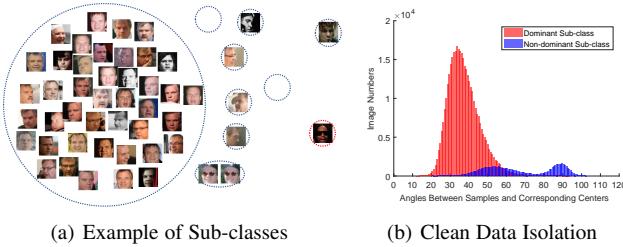


Fig. 6. (a) The sub-classes of one identity from the CASIA dataset [56] after using the sub-center ArcFace loss ( $K = 10$ ). Noisy samples and hard samples (e.g. profile and occluded faces) are automatically separated from the majority of clean samples. (b) Angle distribution of samples from the dominant and non-dominant sub-classes. Clean data are automatically isolated by the sub-center ArcFace.

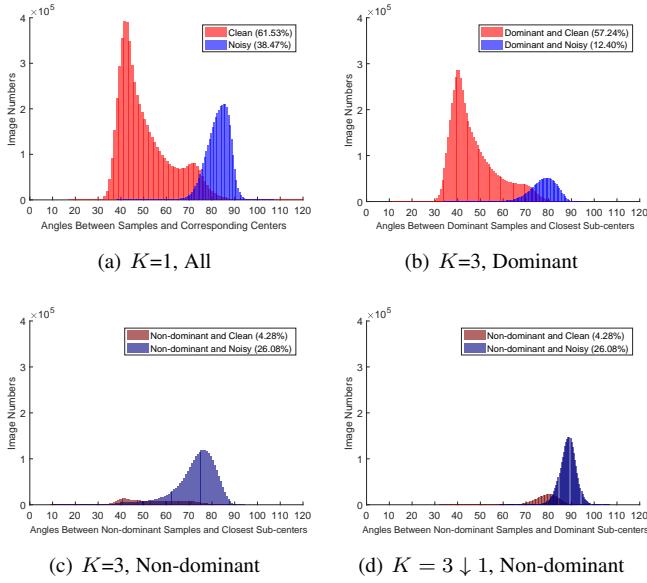


Fig. 7. Data distribution of ArcFace ( $K=1$ ) and the proposed sub-center ArcFace ( $K=3$ ) before and after dropping non-dominant sub-centers. MS1MV0 [37] is used here.  $K = 3 \downarrow 1$  denotes sub-center ArcFace with non-dominant sub-centers dropping.

get the subclass-wise similarity scores  $\mathcal{S} \in \mathbb{R}^{N \times K}$  by a matrix multiplication  $W^T \mathbf{x}_i$ . Then, we employ a max pooling step on the subclass-wise similarity score  $\mathcal{S} \in \mathbb{R}^{N \times K}$  to get the class-wise similarity score  $\mathcal{S}' \in \mathbb{R}^{N \times 1}$ . The proposed sub-center ArcFace loss can be formulated as:

$$L_7 = -\log \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j=1, j \neq y_i}^N e^{s \cos \theta_j}}, \quad (7)$$

where  $\theta_j = \arccos \left( \max_k (W_{j_k}^T \mathbf{x}_i) \right)$ ,  $k \in \{1, \dots, K\}$ .

In Figure 6(a), we have visualized the clustering results of one identity from the CASIA dataset [56] after employing the sub-center ArcFace loss ( $K = 10$ ) for training. It is obvious that the proposed sub-center ArcFace loss can automatically cluster faces such that hard samples and noisy samples are separated away from the dominant clean samples. Note that some sub-classes are empty as  $K = 10$  is too large for a particular identity. In Figure 6(b), we show the angle distribution on the CASIA dataset [56]. We use the pre-trained ArcFace model to predict the feature center of each identity and then calculate the angle between the sample and its corresponding feature center. As we can see from Figure 6(b), most of the samples are close to their centers, however, there are some noisy samples which are far away from their centers. This

observation on the CASIA dataset matches the noise percentage estimation ( $9.3\% \sim 13.0\%$ ) in [18]. To automatically obtain a clean training dataset, the noisy tail is usually removed by a hard threshold (e.g. angle  $\geq 77^\circ$  or cosine  $\leq 0.225$ ). Since sub-center ArcFace can automatically divide the training samples into dominant sub-classes and non-dominant sub-classes, clean samples (in red) can be separated from hard and noisy samples (in blue). More specifically, the majority of clean faces (85.6%) go to the dominant sub-class, while the rest hard and noisy faces go to the non-dominant sub-classes.

Even though using sub-classes can improve the robustness under noise, it undermines the intra-class compactness as hard samples are also kept away as shown in Figure 6(b). In [37], MS1MV0 (around 10M images of 100K identities) is released with the estimated noise percentage around 47.1%  $\sim$  54.4% [18]. In [88], MS1MV0 is refined by a semi-automatic approach into a clean dataset named MS1MV3 (around 5.1M images of 93K identities). Based on these two datasets, we can get the clean and noisy labels on MS1MV0. In Figure 7(b) and Figure 7(c), we show the angle distributions of samples to their closest sub-centers (training settings: [MS1MV0, ResNet50, Sub-center ArcFace  $K=3$ ]). In general, there are four categories of samples: (1) easy clean samples belonging to dominant sub-classes (57.24%), (2) hard noisy samples belonging to dominant sub-classes (12.40%), (3) hard clean samples belonging to non-dominant sub-classes (4.28%), and (4) easy noisy samples belonging to non-dominant sub-classes (26.08%). In Figure 7(a), we show the angle distribution of samples to their corresponding centers from the ArcFace model (training settings: [MS1MV0, ResNet50, ArcFace  $K=1$ ]). By comparing the percentages of noisy samples in Figure 7(b) and Figure 7(a), we find that sub-center ArcFace can significantly decrease the noise rate to around one third (from 38.47% to 12.40%) and this is the reason why sub-center ArcFace is more robust under noise. During the training of sub-center ArcFace, samples belonging to non-dominant sub-classes are pushed to be close to these non-dominant sub-centers as shown in Figure 7(c). Since we have not set any constraint on sub-centers, the sub-centers of each identity can be quite different and even orthogonal. In Figure 7(d), we show the angle distributions of non-dominant samples to their dominant sub-centers. By combining Figure 7(b) and Figure 7(d), we find that the clean and noisy data have some overlaps but a constant angle threshold (between  $70^\circ$  and  $80^\circ$ ) can be easily searched to drop most of the high-confident noisy samples.

Based on the above observations, we propose a straightforward approach to recapture intra-class compactness. We directly drop non-dominant sub-centers after the network has enough discriminative power. Meanwhile, we introduce a constant angle threshold to drop high-confident noisy data. After that, we retrain the ArcFace model from scratch on the automatically cleaned dataset.

### 3.3 Inversion of ArcFace

In the above sections, we have explored how the proposed ArcFace can enhance the discriminative power of a face recognition model. In this section, we take a pre-trained ArcFace model as a white-box and reconstruct identity preserved as well as visually plausible face images only using the gradient of the ArcFace loss and the face statistic priors (e.g. mean and variance) stored in the BN layers. As shown in Figure 8 and illustrated in Algorithm 1, the pre-trained ArcFace model has encoded substantial information of

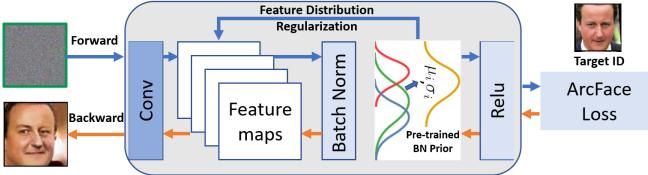


Fig. 8. ArcFace is not only a discriminative model but also a generative model. Given a pre-trained ArcFace model, a random input tensor can be gradually updated into a pre-defined identity by using the gradient of the ArcFace loss as well as the face statistic priors stored in the Batch Normalization layers.

#### Algorithm 1 Face Image Generation from the ArcFace Model

```

Input: model  $\mathcal{M}$  with  $L$  BN layers, class label  $y_i$ 
Output: a batch of generated face images:  $I^r$ 
Generate random data  $I^r$  from Gaussian ( $\mu = 0, \sigma = 1$ )
Get  $\mu_i, \sigma_i$  from BN layers of  $\mathcal{M}, i \in 0, \dots, L$ 
for  $j = 1, 2, \dots, T$  do
    Forward propagate  $\mathcal{M}(I^r)$  and calculate ArcFace loss
    Get  $\tilde{\mu}_i$  and  $\tilde{\sigma}_i$  from intermediate activations,  $i \in 0, \dots, L$ 
    Compute statistic loss  $\min \sum_{i=0}^L \|\tilde{\mu}_i^r - \mu_i\|_2^2 + \|\tilde{\sigma}_i^r - \sigma_i\|_2^2$ 
    Backward propagate and update  $I^r$ 
end for

```

the training distribution. The distribution, stored in BN layers via running mean and running variance, can be effectively employed to generate visually plausible face images, avoiding convergence outside natural faces with high confidence.

Besides the ArcFace loss (Eq. 3) to preserve identity, we also consider the following statistic priors during face generation:

$$L_8 = \sum_{i=0}^L \|\tilde{\mu}_i^r - \mu_i\|_2^2 + \|\tilde{\sigma}_i^r - \sigma_i\|_2^2, \quad (8)$$

where  $\mu_i^r/\sigma_i^r$  are the mean/standard deviation of the distribution at layer  $i$ , and  $\mu_i/\sigma_i$  are the corresponding mean/standard deviation parameters stored in the  $i$ -th BN layer of a pre-trained ArcFace model. After jointly optimizing Eq. 3 and Eq. 8 ( $L_3 + \lambda L_8, \lambda = 0.05$ ) for  $T$  steps as in Algorithm 1, we can generate faces, when fed into the network, not only have same identity as the pre-defined identity but also have a statistical distribution that closely matches the original data set.

The above approach exploits the relationship between an input image and its class label for the reconstruction process. As the output similarity score is fixed according to predefined  $N$  classes, the reconstruction is limited on images of training subjects. To solve open-set face generation from the embedding feature, the constraints on predefined classes need to be removed. Therefore, we substitute the classification loss to the  $\ell_2$  loss between feature pairs. Open-set face generation can restore the face image from any embedding feature, while close-set face generation only reconstructs face images from the class centers stored in the linear weight.

Concurrent with our work, [33], [34], [35] have proposed a data-free method employing the BN priors to restore ImageNet images for distillation, quantization and pruning. Their model inversion results contain obvious artifact in the background due to the translation augmentation during training. By contrast, our ArcFace model is trained on normalized face crops without background, thus the restored faces exhibit less artifact. Besides, these

TABLE 1  
Face datasets for training and testing. "(D)" refers to the distractors. IBUG-500K is the training data automatically refined by the proposed sub-center ArcFace. LFR2019-Image and LFR2019-Video are the proposed large-scale image and video test sets.

Datasets	#Identity	#Image/Video
CASIA [56]	10K	0.5M
VGG2 [9]	9.1K	3.3M
MS1MV0 [37]	100K	10M
MS1MV3 [88]	93K	5.1M
Celeb500K [38]	500K	50M
<b>IBUG-500K</b>	493K	11.96M
LFW [89]	5,749	13,233
YTF [90]	1,595	3,425
CFP-FP [74]	500	7,000
CPLFW [75]	5,749	11,652
AgeDB [76]	568	16,488
CALFW [77]	5,749	12,174
MegaFace [78]	530	1M (D)
IJB-B [79]	1,845	76.8K
IJB-C [80]	3,531	148.8K
<b>LFR2019-Image</b> [88]	5.7K	1.58M(D)
<b>LFR2019-Video</b> [88]	10K	200K

data-free methods only considered close-set image generation but ArcFace can freely restore both close-set and open-set subjects. In this paper, we show that the proposed additive angular margin loss can also improve face generation.

## 4 EXPERIMENTS

### 4.1 Implementation Details

**Training Datasets.** As given in Table 1, we separately employ CASIA [56], VGG2 [9], MS1MV0 [37] and Celeb500K [38] as our training data in order to conduct fair comparison with other methods. MS1MV0 (loose cropped version) [37] is a raw data with the estimated noise percentage around 47.1%  $\sim$  54.4% [18]. MS1MV3 [88] is cleaned from MS1MV0 [37] by a semi-automatic approach. We employ ethnicity-specific annotators (e.g. African, Caucasian, Indian and Asian) for large-scale face image annotations, as the boundary cases (e.g. hard samples and noisy samples) are very hard to distinguish if the annotator is not familiar with the identity. Celeb500K [38] is collected in the same way as MS1MV0 [37], using the celebrity name list [37] to search identities from Google and download the top-ranked face images. We download 25M images of 500K identities, and employ RetinaFace [8] to detect faces larger than  $50 \times 50$  from the original images. By employing the proposed sub-center ArcFace, we can automatically clean MS1MV0 [37] and Celeb500K [38]. After removing the overlap identities (about 50K) through the ID string, we combine the automatically cleaned MS1MV0 and Celeb500K and obtain a large-scale face image dataset, named IBUG-500K, including 11.96 million images of 493K identities. Figure 9 illustrates the gender, race, pose, age and image number distributions of the proposed IBUG-500K dataset.

**Test Datasets.** During training, we explore efficient face verification datasets (e.g. LFW [89], CFP-FP [74], AgeDB [76]) to check the convergence status of the model. Besides the most widely used LFW [89] and YTF [90] datasets, we also report the performance of ArcFace on the recent datasets (e.g. CPLFW [75] and CALFW [77]) with large pose and age variations. We also extensively test the proposed ArcFace on large-scale image datasets (e.g. MegaFace [78], IJB-B [79], IJB-C [80] and LFR2019-Image [88])

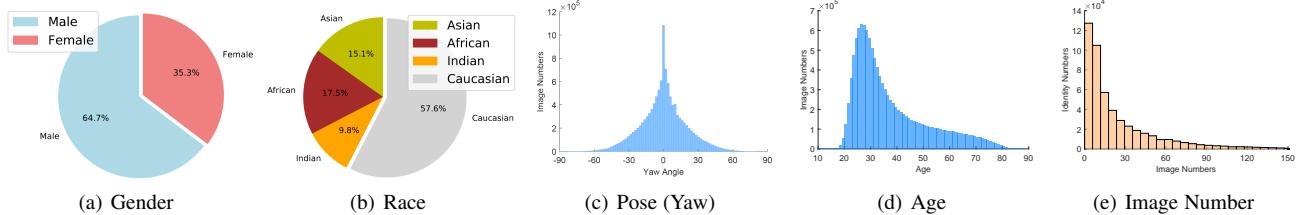


Fig. 9. IBUG-500K statistics. We show the (a) gender, (b) race, (c) yaw pose, (d) age and (e) image number distributions of the proposed large-scale training dataset.

and large-scale video datasets (LFR2019-Video [88]). Detailed dataset statistics are presented in Table 1. For the LFR2019-Image dataset, there are 274K images from the 5.7K LFW identities [89] and 1.58M distractors downloaded from Flickr. For the LFR2019-Video dataset, there are 200K videos of 10K identities collected from various shows, films and television dramas. The length of each video ranges from 1 to 30 seconds. Both the LFR2019-Image dataset and the LFR2019-Video dataset are manually cleaned to ensure the unbiased evaluation of different face recognition models.

**Experimental Settings.** For data prepossessing, we follow the recent papers [13], [14] to generate the normalized face crops ( $112 \times 112$ ) by utilizing five facial points predicted by RetinaFace [8]. For the embedding network, we employ the widely used CNN architectures, ResNet50 and ResNet100 [58], [91] without the bottleneck structure. After the last convolutional layer, we explore the BN [92]-Dropout [93]-FC-BN structure to get the final 512- $D$  embedding feature. In this paper, we use ([training dataset, network structure, loss]) to facilitate understanding of different experimental settings.

We follow [14] to set the feature scale  $s$  to 64 and choose the angular margin  $m$  of ArcFace at 0.5. All recognition experiments in this paper are implemented by MXNet [39]. We set the batch size to 512 and train models on eight NVIDIA Tesla P40 (24GB) GPUs. We set the momentum to 0.9 and weight decay to  $5e-4$ . For the ArcFace training, we employ the SGD optimizer and follow [14], [9] to design the learning rate schedules for different datasets. On CASIA, the learning rate starts from 0.1 and is divided by 10 at 20, 28 epochs. The training process is finished at 32 epochs. On VGG2, the learning rate is decreased at 6, 9 epochs and we finish training at 12 epochs. On MS1MV3 and IBUG-500K, we refer to the verification accuracy on CFP-FP and AgeDB to reduce the learning rate at 8, 14 epochs and terminate at 18 epochs.

For the training of the proposed sub-center ArcFace on MS1MV0 [37], we also employ the same learning rate schedule as on MS1MV3 to train the first round of model ( $K=3$ ). Then, we drop non-dominant sub-centers ( $K = 3 \downarrow 1$ ) and high-confident noisy data ( $> 75^\circ$ ) by using the first round model through an off-line way. Finally, we retrain the model from scratch using the automatically cleaned data. For the experiments of the sub-center ArcFace on Celeb500K [38], the only difference is the learning rate schedule, which is same as on IBUG-500K.

During testing of the face recognition models, we only keep the feature embedding network without the fully connected layer (160MB for ResNet50 and 250MB for ResNet100) and extract the 512- $D$  features (8.9 ms/face for ResNet50 and 15.4 ms/face for ResNet100) for each normalized face. To get the embedding features for templates (e.g. IJB-B and IJB-C) or videos (e.g. YTF and LFR2019-Video), we simply calculate the feature center of all

TABLE 2  
Verification results (%) of different loss functions ([CASIA, ResNet50, Loss\*]).

Loss Functions	LFW	CFP-FP	AgeDB
ArcFace (0.4)	99.53	95.41	94.98
ArcFace (0.45)	99.46	95.47	94.93
ArcFace (0.5)	<b>99.53</b>	<b>95.56</b>	<b>95.15</b>
ArcFace (0.55)	99.41	95.32	95.05
SphereFace [13]	99.42	-	-
SphereFace (1.35)	99.11	94.38	91.70
CosFace [14]	99.33	-	-
CosFace (0.35)	99.51	95.44	94.56
CM1 (1, 0.3, 0.2)	99.48	95.12	94.38
CM2 (0.9, 0.4, 0.15)	99.50	95.24	94.86
Softmax	99.08	94.39	92.33
Norm-Softmax ( $s = 64$ )	98.56	89.79	88.72
Norm-Softmax ( $s = 20$ )	99.20	94.61	92.65
Norm-Softmax+Intra	99.30	94.85	93.58
Norm-Softmax+Inter	99.22	94.73	92.94
Norm-Softmax+Intra+Inter	99.31	94.88	93.76
Triplet (0.35)	98.98	91.90	89.98
ArcFace+Intra	99.45	95.37	94.73
ArcFace+Inter	99.43	95.25	94.55
ArcFace+Intra+Inter	99.43	95.42	95.10
ArcFace+Triplet	99.50	95.51	94.40

images from the template or all frames from the video.

## 4.2 Ablation Study on ArcFace

In Table 2, we first explore the angular margin setting for ArcFace on the CASIA dataset with ResNet50. The best margin observed in our experiments is 0.5. Using the proposed combined margin framework in Eq. 4, it is easier to set the margin of SphereFace and CosFace which we find to have optimal performance when setting at 1.35 and 0.35, respectively. Our implementations for both SphereFace and CosFace can lead to excellent performance without observing any difficulty in convergence. The proposed ArcFace achieves the highest verification accuracy on all three test sets. In addition, we perform extensive experiments with the combined margin framework (some of the best performance is observed for CM1 (1, 0.3, 0.2) and CM2 (0.9, 0.4, 0.15)) guided by the target logit curves in Figure 4(b). The combined margin framework leads to better performance than individual SphereFace and CosFace but upper-bounded by the performance of ArcFace.

Besides the comparison with margin-based methods, we conduct a further comparison between ArcFace and other losses which aim at enforcing intra-class compactness (Eq. 5) and inter-class discrepancy (Eq. 6). As the baseline, we choose the softmax loss. After weight and feature normalization, we have observed obvious performance drops on CFP-FP and AgeDB with the feature re-scale parameter  $s$  set as 64. To obtain comparable performance as the softmax loss, we have searched the best scale

TABLE 3

Ablation experiments of different settings of the proposed sub-center ArcFace on MS1MV0, MS1MV3 and Celeb500K. The 1:1 verification accuracy ( $TPR@FPR=1e-4$ ) is reported on the IJB-B and IJB-C datasets. ([MS1MV0 / MS1MV3 / Celeb500K, ResNet50, Sub-center ArcFace])

Settings	IJB-B	IJB-C
(1) MS1MV0, $K=1$	87.87	90.27
(2) MS1MV0, $K=3$	91.70	93.72
(3) MS1MV0, $K=3$ , softmax pooling [62]	91.53	93.55
(4) MS1MV0, $K=5$	91.47	93.62
(5) MS1MV0, $K=10$	63.84	67.94
(6) MS1MV0, $K = 3 \downarrow 1$ , drop $> 70^\circ$	94.44	95.91
(7) MS1MV0, $K = 3 \downarrow 1$ , drop $> 75^\circ$	94.56	95.92
(8) MS1MV0, $K = 3 \downarrow 1$ , drop $> 80^\circ$	94.04	95.74
(9) MS1MV0, $K = 3 \downarrow 1$ , drop $> 85^\circ$	93.33	95.10
(10) MS1MV0, $K=3$ , regularizer [62]	91.53	93.64
(11) MS1MV0,Co-mining [21]	91.80	93.82
(12) MS1MV0,NT [19]	91.57	93.65
(13) MS1MV0,NR [20]	91.58	93.60
(14) MS1MV3, $K=1$	95.13	96.50
(15) MS1MV3, $K=3$	94.84	96.35
(16) MS1MV3, $K = 3 \downarrow 1$	94.87	96.43
(17) Celeb500K, $K=1$	90.96	92.15
(18) Celeb500K, $K=3$	93.76	94.90
(19) Celeb500K, $K = 3 \downarrow 1$	<b>95.65</b>	<b>96.91</b>

parameter  $s = 20$  for Norm-Softmax. By combining the Norm-Softmax with the intra-class loss, the performance improves on CFP-FP and AgeDB. However, combining the Norm-Softmax with the inter-class loss only slightly improves the accuracy. Employing margin penalty within triplet samples is less effective than inserting margin between samples and centers as in ArcFace, indicating local comparisons in the Triplet loss are not as effective as global comparisons in ArcFace. Finally, we incorporate the Intra-loss, Inter-loss and Triplet-loss into ArcFace, but no obvious improvement is observed, which leads us to believe that ArcFace is already enforcing intra-class compactness, inter-class discrepancy and classification margin.

### 4.3 Ablation Study on Sub-center ArcFace

In Table 3, we conduct extensive experiments to investigate the proposed sub-center ArcFace on noisy training data (e.g. MS1MV0 [37] and Celeb500K [38]). Models trained on the manually cleaned MS1MV3 [88] are taken as the reference. We train ResNet50 networks under different settings and evaluate the performance by adopting  $TPR@FPR=1e-4$  on IJB-C, which is more objective and less affected by the noise within the test data [94].

From Table 3, we have the following observations:

- ArcFace has an obvious performance drop (from (14) 96.50% to (1) 90.27%) when the training data is changed from the clean MS1MV3 to the noisy MS1MV0. By contrast, sub-center ArcFace is more robust ((2) 93.72%) under massive noise.
- Too many sub-centers (too large  $K$ ) can obviously undermine the intra-class compactness and decrease the accuracy (from (2) 93.72% to (5) 67.94%). This observation indicates that noise tolerance and intra-class compactness should be balanced during training. Considering the GPU memory consumption, we select  $K=3$  in this paper.
- The nearest sub-center assignment by the max pooling is slightly better than the softmax pooling [62] ((2) 93.72%

vs. (3) 93.55%). Thus, we choose the more efficient max pooling operator in the following experiments.

- Dropping non-dominant sub-centers and high-confident noisy samples can achieve better performance than adding regularization [62] to enforce compactness between sub-centers ((7) 95.92% vs. (10) 93.64%). Besides, the performance of our method is not very sensitive to the constant threshold ((6) 95.91%, (7) 95.92% and (8) 95.74%), and we select  $75^\circ$  as the threshold for dropping high-confident noisy samples in the following experiments.
- Co-mining [21] and re-weighting methods [19], [20] can also improve the robustness under massive noise, but sub-center ArcFace can do better through automatic clean and noisy data isolation during training ((7) 95.92% vs. (11) 93.82%, (12) 93.65% and (13) 93.60%).
- On the clean dataset (MS1MV3), sub-center ArcFace achieves similar performance as ArcFace ((16) 96.43% vs. (14) 96.50%). By employing the threshold of  $75^\circ$  on MS1MV3, 4.18% hard samples are removed, but the performance only slightly decreases, thus we estimate MS1MV3 still contains some noises.
- The proposed sub-center ArcFace trained on noisy MS1MV0 can achieve comparable performance compared to ArcFace trained on manually cleaned MS1MV3 ((7) 95.92% vs. (14) 96.50%).
- By enlarging the training data, sub-center ArcFace can easily achieve better performance even though it is trained from noisy web faces ((19) 96.91% vs. (13) 96.50%).

### 4.4 Benchmark Results

**Results on LFW, YTF, CFP-FP, CPLFW, AgeDB, CALFW.** LFW [89] and YTF [90] datasets are the most widely used benchmark for unconstrained face verification on images and videos. In this paper, we follow the *unrestricted with labelled outside data* protocol to report the performance. As reported in Table 4, ArcFace models trained on MS1MV3 and IBUG-500K with ResNet100 beat the baselines (e.g. SphereFace [13] and CosFace [14]) on both LFW and YTF, which shows that the additive angular margin penalty can notably enhance the discriminative power of deeply learned features, demonstrating the effectiveness of ArcFace. As the margin-based softmax loss has been widely used in recent methods, the performance begins to be saturated around 99.8% and 98.0% on LFW and YTF, respectively. However, the proposed ArcFace is still among the most competitive face recognition methods.

Besides on LFW and YTF datasets, we also report the performance of ArcFace on the recently introduced datasets (e.g. CFP-FP [74], CPLFW [75], AgeDB [76] and CALFW [77]) which show large pose and age variations. Among all of the recent face recognition models, our ArcFace models trained on MS1MV3 and IBUG-500K are evaluated as the top-ranked face recognition models as shown in Table 5, outperforming counterparts by an obvious margin on the pose-invariant and age-invariant face recognition. In Figure 10, we show the results of ArcFace model trained on IBUG-500K by illustrating the angle distributions of both positive and negative pairs on LFW, YTF, CFP-FP, CPLFW, AgeDB and CALFW. We can clearly find that the intra-variance due to pose and age gaps significantly increases the angles between positive pairs thus making the best threshold for face verification increasing and generating more confusion regions on the histogram.

TABLE 4

Verification performance (%) of different methods on LFW and YTF. ([Dataset\*, ResNet100, ArcFace])

Method	#Image	LFW	YTF
DeepID [1]	0.2M	99.47	93.20
Deep Face [2]	4.4M	97.35	91.4
VGG Face [4]	2.6M	98.95	97.30
FaceNet [3]	200M	99.63	95.10
Baidu [95]	1.3M	99.13	-
Center Loss [72]	0.7M	99.28	94.9
Range Loss [73]	5M	99.52	93.70
Marginal Loss [17]	3.8M	99.48	95.98
SphereFace [13]	0.5M	99.42	95.0
SphereFace+ [84]	0.5M	99.47	-
CosFace [14]	5M	99.73	97.6
RegularFace [51]	3.1M	99.61	96.7
UniformFace [52]	6.1M	99.8	97.7
DAL [96]	0.5M	99.47	-
FTL [97]	5M	99.55	-
Fair Loss [98]	0.5M	99.57	96.2
Unequal-training [20]	0.55M	99.53	96.04
Noise-Tolerant [19]	1M noisy	99.72	97.36
AdaptiveFace [50]	5M	99.62	-
AFRN [99]	3.1M	<b>99.85</b>	97.1
PFE [100]	4.4M	99.82	97.36
DUL [101]	3.6M	99.78	96.78
RDCFace [102]	1.7M	99.80	97.10
HPDA [103]	5M	99.80	-
URFace [104]	5M	99.78	-
CircleLoss [105]	3.6M	99.73	96.38
GroupFace [55]	5.8M	<b>99.85</b>	97.8
BioMetricNet [106]	3.8M	99.80	<b>98.06</b>
BroadFace [107]	5.8M	<b>99.85</b>	98.0
IBUG500K, R100, BroadFace	11.96M	99.83	98.03
MS1MV3, R100, ArcFace	5.1M	99.83	98.02
IBUG500K, R100, ArcFace	11.96M	99.83	98.01

TABLE 5

Verification performance (%) of different methods on CFP-FP, CPLFW, AgeDB and CALFW. ([Dataset\*, ResNet100, ArcFace])

Method	CFP-FP	CPLFW	AgeDB	CALFW
Center Loss [72]	-	77.48	-	85.48
SphereFace [13]	-	81.40	-	90.30
VGGFace2 [9]	-	84.00	-	90.57
MV-Softmax [53]	98.28	92.83	97.95	<b>96.10</b>
Search-Softmax [108]	95.64	89.50	97.75	95.40
FaceGraph [109]	96.90	92.27	97.92	95.67
CurricularFace [54]	98.36	93.13	98.37	96.05
MS1MV3, R100, ArcFace	98.79	93.21	98.23	96.02
IBUG500K, R100, ArcFace	<b>98.87</b>	<b>93.43</b>	<b>98.38</b>	<b>96.10</b>

**Results on MegaFace.** The MegaFace dataset [78] includes 1M images of 690K different individuals as the gallery set and 100K photos of 530 unique individuals from FaceScrub [112] as the probe set. As we observed an obvious performance gap between identification and verification in the previous work (e.g. CosFace [14]), we performed a thorough manual check in the whole MegaFace dataset and found many face images with wrong labels, which significantly affects the performance. Therefore, we manually refined the whole MegaFace dataset and report the correct performance of ArcFace on MegaFace. In Table 6, we use “R” to denote the refined version of MegaFace and the performance comparisons also focus on the refined version.

On MegaFace, there are two testing scenarios (identification and verification) under two protocols (large or small training set). The training set is defined as large if it contains more than 0.5M images. For the fair comparison, we train ArcFace on CASIA

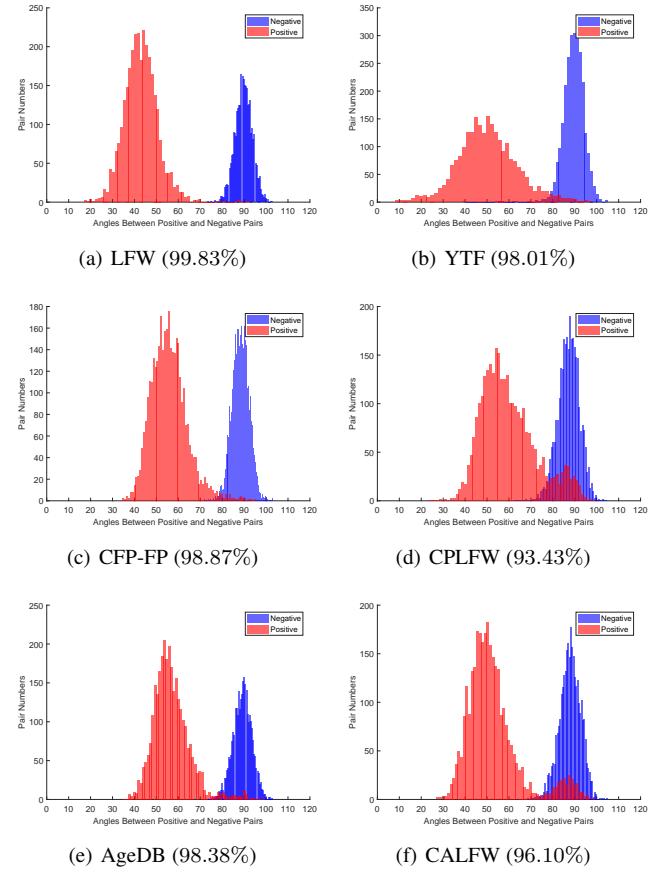


Fig. 10. Angle distributions of both positive and negative pairs on LFW, YTF, CFP-FP, CPLFW, AgeDB and CALFW. The red histogram indicates positive pairs while the blue histogram indicates negative pairs. All angles are represented in degree. ([IBUG-500K, ResNet100, ArcFace])

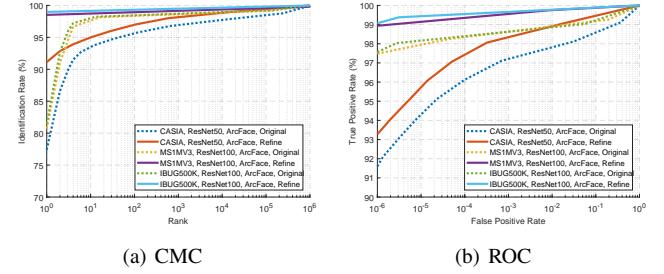


Fig. 11. CMC and ROC curves of different models on MegaFace. Results are evaluated on both original and refined MegaFace dataset.

and IBUG-500K under the small protocol and large protocol, respectively. In Table 6, ArcFace trained on CASIA achieves the best single-model identification and verification performance, not only surpassing the strong baselines (e.g. SphereFace [13] and CosFace [14]) but also outperforming other published methods [72], [84].

Under the large protocol, ArcFace trained on IBUG-500K surpasses ArcFace trained on MS1MV3 by a clear margin (0.47% improvement on identification), which indicates that large-scale training data is very beneficial and the proposed sub-center ArcFace is effective for automatic data cleaning under different data scales. As shown in Figure 11, ArcFace trained on IBUG-500K forms an upper envelope of other models under both identification and verification scenarios. Compared to MC-FaceGraph [109], ArcFace trained on IBUG-500K obtains comparable results on

TABLE 6

Face identification and verification evaluation of different methods on MegaFace Challenge1 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  $10^{-6}$  FPR. “R” refers to data refinement on both probe set and 1M distractors of MegaFace. ArcFace obtains state-of-the-art performance under both small and large protocols.

Methods	Id (%)	Ver (%)
Softmax [13]	54.85	65.92
Contrastive Loss[13], [1]	65.21	78.86
Triplet [13], [3]	64.79	78.32
Center Loss[72]	65.49	80.14
SphereFace [13]	72.729	85.561
CosFace [14]	77.11	89.88
AM-Softmax [15]	72.47	84.44
SphereFace+ [84]	73.03	-
RegularFace [51]	70.23	84.07
CASIA, R50, ArcFace	77.42	91.69
CASIA, R50, ArcFace, R	91.12	93.56
FaceNet [3]	70.49	86.47
CosFace [14]	82.72	96.65
UniformFace [52]	79.98	95.36
RegularFace [51]	75.61	91.13
AdaptiveFace, R [50]	95.02	95.61
MV-Softmax, R [53]	98.00	98.31
P2SGrad,R [48]	97.25	-
Adocos, R [49]	97.41	-
PFE [100]	78.95	92.51
Fair Loss [98]	77.45	92.87
Search-Softmax, R [108]	96.97	97.84
Domain Balancing, R [110]	96.35	96.56
URFace [104]	78.60	95.04
DUL, R [101]	98.60	-
CircleLoss, R [105]	98.50	98.73
CurricularFace, R [54]	98.25	98.44
GroupFace, R [55]	98.74	98.79
MC-FaceGraph, R [109]	<b>99.02</b>	98.94
SST, R [111]	96.27	96.96
BroadFace, R [107]	98.70	98.95
MS1MV3, R100, ArcFace	80.71	97.46
MS1MV3, R100, ArcFace, R	98.51	98.74
IBUG-500K, R100, ArcFace	81.43	97.63
IBUG-500K, R100, ArcFace,R	98.98	<b>99.08</b>

identification and better results on verification. Considering 18.8M images of 636K identities are used in MC-FaceGraph [109], the performance of our method is very impressive, as we only use images automatically cleaned from noisy web data. Similar to LFW, the identification results on MegaFace are also saturated (around 99%). Therefore, the performance gap of 0.04% on identification is negligible and our model is among the most competitive face recognition methods.

**Results on IJB-B and IJB-C.** The IJB-B dataset [79] contains 1,845 subjects with 21.8K still images and 55K frames from 7,011 videos. The IJB-C dataset [79] is a further extension of IJB-B, having 3,531 subjects with 31.3K still images and 117.5K frames from 11,779 videos. On IJB-B and IJB-C datasets, there are two evaluation protocols, 1:1 verification and 1:N identification.

For the widely used 1:1 verification protocol, there are 12,115 templates with 10,270 genuine matches and 8M impostor matches on IJB-B, and there are 23,124 templates with 19,557 genuine matches and 15,639K impostor matches on IJB-C. In Table 7, we compare the TPR (@FPR=1e-4) of ArcFace with the previous state-of-the-art models. We first employ the VGG2 [9] dataset as

TABLE 7  
1:1 verification (TPR@FPR=1e-4) on IJB-B and IJB-C.

Method	IJB-B (%)	IJB-C (%)
ResNet50 [9]	78.4	82.5
SENet50 [9]	80.0	84.0
MN-vc [113]	83.1	86.2
DCN [94]	84.9	88.5
Crystal Loss [114]	-	92.29
AIM [115]	-	89.5
P2SGrad [48]	-	92.25
Adocos [49]	-	92.4
PFE [100]	-	93.3
MV-Softmax [53]	93.6	95.2
AFRN [99]	88.5	93.1
PFE [100]	-	93.25
DUL [101]	-	94.61
URFace [104]	-	96.6
CircleLoss [105]	-	93.95
CurricularFace [54]	94.86	96.15
GroupFace [55]	94.93	96.26
BroadFace [107]	94.61	96.03
VGG2, R50, ArcFace	89.8	92.79
MS1MV3, R100, ArcFace	95.42	96.83
IBUG-500K, R100, ArcFace	<b>96.02</b>	<b>97.27</b>

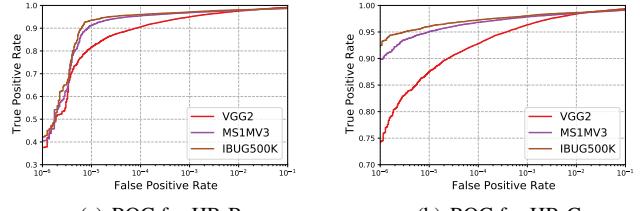


Fig. 12. ROC curves of 1:1 verification protocol on IJB-B and IJB-C. ([Dataset\*, ResNet100, ArcFace])

the training data and the ResNet50 as the embedding network to train ArcFace for the fair comparison with the most recent softmax-based methods [9], [113], [94]. As we can see from the results, the proposed additive angular margin can obviously boost the performance on both IJB-B and IJB-C compared to the softmax loss (about 3 ~ 5%, which is a significant reduction in the error).

Drawing support from more training data (IBUG-500K) and deeper neural network (ResNet100), ArcFace can further improve the TPR (@FPR=1e-4) to 96.02% and 97.27% on IJB-B and IJB-C, respectively. Compared to the joint margin-based and mining-based method (e.g. CurricularFace [54]), our method further decreases the error rate by 22.57% and 29.09% on IJB-B and IJB-C, which indicates that the automatically cleaned data

TABLE 8  
1:1 verification (TPR@FPR=1e-5) and 1:N identification (Rank-1) on IJB-B and IJB-C. ([Dataset\*, ResNet100, ArcFace])

Training Datasets	IJB-B		IJB-C	
	Ver.(%)	Id.(%)	Ver.(%)	Id.(%)
CASIA [56]	62.42	86.70	69.61	88.05
IMDB-Face [18]	64.87	93.41	66.85	94.52
VGG2 [9]	41.64	93.20	59.33	94.44
MS1MV1 [17]	80.27	92.19	88.16	93.54
MS1MV2 [16]	89.33	94.50	93.15	95.72
MC-FaceGraph [109]	92.82	95.76	95.62	96.93
MS1MV3	91.27	95.04	95.56	96.94
IBUG-500K	<b>93.48</b>	<b>95.94</b>	<b>96.07</b>	<b>97.21</b>

TABLE 9

Verification results (%) on the LFR2019-Image ( $\text{TPR@FPR}=1\text{e-}8$ ) and LFR2019-Video ( $\text{TPR@FPR}=1\text{e-}4$ ) datasets. ([Dataset\*, Network\*, ArcFace])

Methods	Image	Video
YMJ <sup>1</sup> [116]	<b>88.78</b>	-
count <sup>2</sup> [117]	88.42	-
NothingLC <sup>3</sup>	88.14	-
NothingLC <sup>1</sup>	-	63.23
Rhapsody <sup>2</sup>	-	61.87
xfr <sup>3</sup>	-	61.05
Our Method	88.65	<b>63.60</b>
MS1MV3, EfficientNet-B0, ArcFace	86.44	61.47
MS1MV3, R100, ArcFace	92.75	64.89

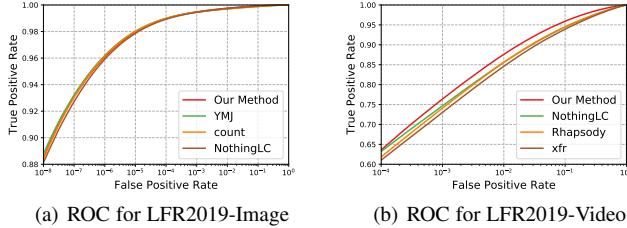


Fig. 13. ROC curves of 1:1 verification protocol on the LFR2019-Image and LFR2019-Video datasets. ([MS1MV3, EfficientNet-B0, ArcFace])

by the proposed sub-center ArcFace are effective to boost the performance. In Table 8, we compare the proposed sub-center ArcFace with FaceGraph [109] on large-scale cleansing. In FaceGraph [109], one million celebrities (87.0M face images) [37] are cleaned into a noise-free dataset named MC-FaceGraph (including 18.8M face images of 636.2K identities) by employing a global-local graph convolutional network. Even though the proposed sub-center ArcFace is only applied to half million identities, the cleaned dataset, IBUG-500K (including 11.96M face images of 493K identities), still outperforms MC-FaceGraph [109]. Under the evaluation metric of  $\text{TPR@FPR}=1\text{e-}5$ , the ArcFace model trained on IBUG-500K surpasses the counterpart trained on MC-FaceGraph by 0.66% and 0.45% on IJB-B and IJB-C, respectively. In Figure 12, we show the full ROC curves of the proposed ArcFace on IJB-B and IJB-C, and ArcFace achieves impressive performance even at  $\text{FPR}=1\text{e-}6$  setting a new baseline.

For the 1:N end-to-end mixed protocol, there are 10,270 probe templates containing 60,758 still images and video frames on IJB-B, and there are 19,593 probe templates containing 127,152 still images and video frames on IJB-C. In Table 8, we report the Rank-1 identification accuracy of our method compared to baseline models. ArcFace trained on IBUG-500K achieves impressive performance on both IJB-B (95.94%) and IJB-C (97.21%), setting a new record on this benchmark.

**Results on LFR2019-Image and LFR2019-Video.** Lightweight Face Recognition (LFR) Challenge [88] targets on bench-marking face recognition methods under strict computation constraints (i.e. computational complexity  $< 1.0 \text{ GFlops}$ ). For a fair comparison, all participants in the challenge must use MS1MV3 [88] as the training data. On LFR2019-Image, trillion-level pairs between gallery and probe set are used for evaluation and  $\text{TPR@FPR}=1\text{e-}8$  is selected as the main evaluation metric. On LFR2019-Video, billion-level pairs between all videos are used for evaluation and  $\text{TPR@FPR}=1\text{e-}4$  is employed as the main evaluation metric.

In Table 9, we compare the performance of ArcFace with the top-ranked competition solutions [88]. For the design of

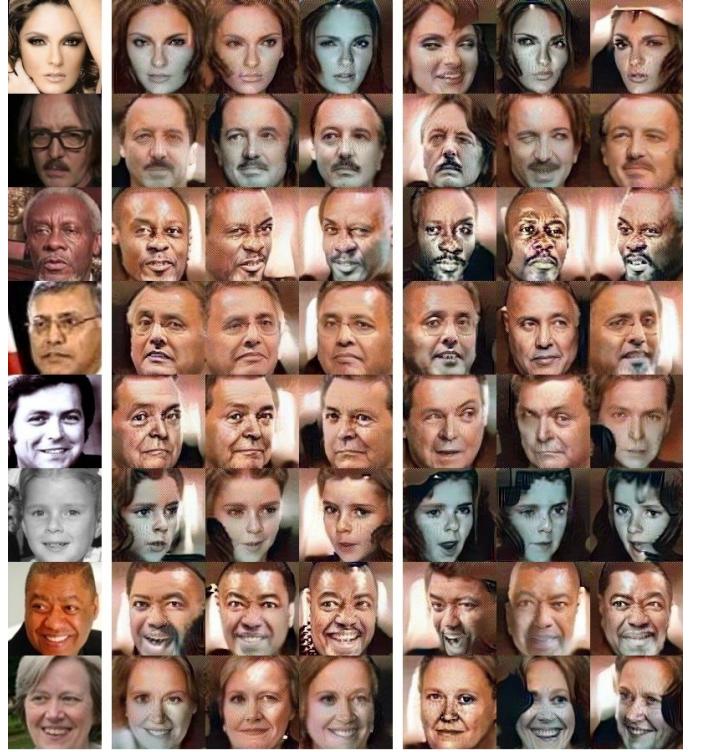


Fig. 14. Close-set face generation. ArcFace can generate identity-preserved face images only by using the model parameters without training any additional discriminator and generator like in GAN. The first column is the identity from the training data. Column 2 to 4 are the outputs from our ArcFace model. Column 5 to 7 are the outputs from the baseline CosFace model.

our lightweight model, we explore EfficientNet-B0 [118] as the backbone. When training from scratch with the proposed ArcFace loss, EfficientNet-B0 can obtain 86.44% on LFR2019-Image and 61.47% on LFR2019-Video, respectively. Following the top-ranked solutions, we also employ knowledge distillation [119] to boost the performance of our lightweight model. ArcFace trained on MS1MV3 with ResNet100 provides a high-performance teacher network, achieving 92.75% on LFR2019-Image and 64.89% on LFR2019-Video. With the assistance of the teacher network, our lightweight model is trained by minimizing (1) the ArcFace loss (2) the  $\ell_2$  regression loss between 512-D features of the teacher and student networks, and (3) the KL loss [119] between class-wise similarities predicted by the teacher and student networks. The weights of the  $\ell_2$  regression loss and the KL loss is set to 1.0 and 0.1, respectively. With knowledge distillation, our method finally achieves 88.65% on LFR2019-Image and 63.60% on LFR2019-Video. As shown in Figure 13, our method obtains comparable performance with the champion of the LFR2019-Image track and envelopes the ROC curves of all top-ranked challenge solutions in the LFR2019-Video track, surpassing the champion by 0.37%.

#### 4.5 Inversion of ArcFace

This section demonstrates the capability of the proposed ArcFace model in terms of effectively synthesizing identity-preserved face images from subject’s centers (the close-set setting) or features (the open-set setting).

We adopt the ArcFace (ResNet50) trained on MS1MV3 to conduct the inversion experiments, which include two settings,

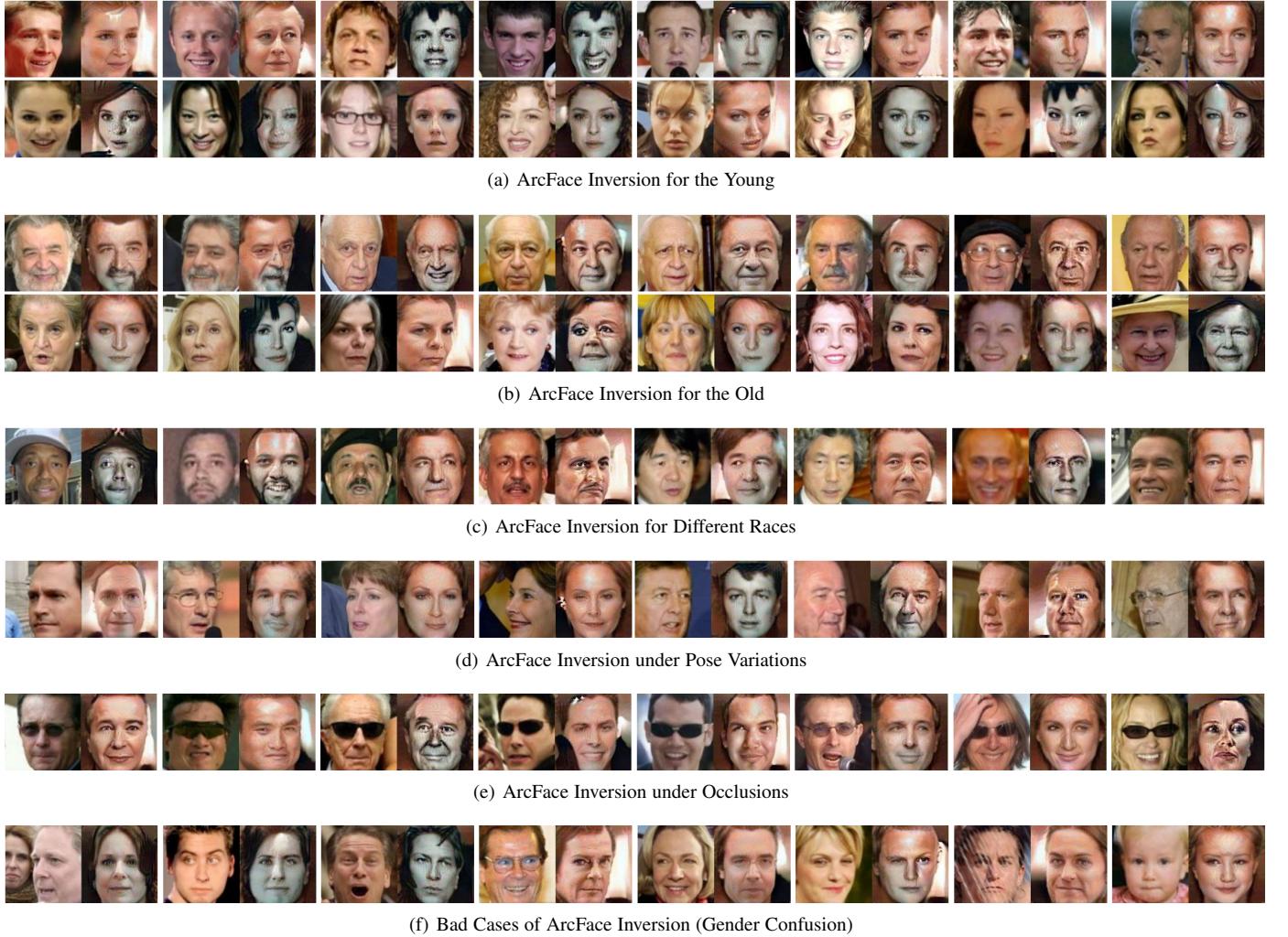


Fig. 15. Open-set face generation from the pre-trained ArcFace model. We show the ArcFace inversion results (right) under age, gender, race, pose and occlusion variations by only using the embedding features from LFW [89] test samples (left). In the bottom, we show some bad cases (e.g. gender confusion) generated from the ArcFace inversion.



Fig. 16. Open-set face generation without and with BN constraints. The first row is the original LFW [89] samples. The second row is the ArcFace inversion results without BN constraints, and the third row is the ArcFace inversion results with BN constraints.

TABLE 10

FID and cosine similarity of different model inversion results. ArcFace model (ResNet50) for inversion is trained on MS1MV3, but the generated face images also exhibit high similarity from the view of the more powerful ArcFace model (ResNet100) trained on IBUG-500K. The margin parameter for each method is given in the bracket.

Method	FID	Cosine Similarity
Softmax	75.59	0.5612
SphereFace (1.35)	73.18	0.5919
CosFace (0.35)	71.64	0.6176
ArcFace (0.5)	<b>70.39</b>	<b>0.6248</b>

i.e. close-set and open-set. In the close-set mode, centers stored in the linear layer are selected as the targets to generate face

TABLE 11  
FID, cosine similarity and verification accuracy on LFW of different model inversion results. The cosine similarity and the verification accuracy are tested by the ArcFace model (ResNet100) trained on IBUG-500K. The margin parameter for each method is given in the bracket.

Method	FID	Cosine Sim	LFW Acc (%)
Softmax	77.85	0.5504	90.14
SphereFace (1.35)	75.16	0.5687	92.05
CosFace (0.35)	74.02	0.5762	92.69
ArcFace (0.5)	<b>73.16</b>	<b>0.5849</b>	<b>93.30</b>

images. Identity preservation is constrained by a classification loss (e.g. Softmax, SphereFace, CosFace and ArcFace). In the open-set

mode, embedding features predicted by the pre-trained models are used as the targets to generate face images. Identity preservation is constrained by a  $\ell_2$  loss. For each time, we synthesize 256 face images of different identities at the resolution of  $112 \times 112$  in one mini-batch using one NVIDIA V100 GPU. We employ Adam optimizer [120] at a learning rate of 0.25 and the iteration lasts 20K steps. Regularization parameters [30] for total variance and  $\ell_2$  norm of the generated faces are set as  $1e - 3$  and  $1e - 4$ , respectively.

In order to quantitatively validate how well the proposed method can preserve the identity of the subject and how visually plausible the reconstructed face image is, three metrics are adopted: (1) Frechet Inception Distance (FID) [121]; (2) cosine similarity from a third-party model ([IBUG-500K, ResNet100, ArcFace]); and (3) face verification accuracy on LFW for open-set experiments.

**Close-set Face Generation.** In Table 10, we quantify the realism and identity preservation of the reconstructed faces from different face recognition models. For each model, we synthesize training identities by using the 5K randomly selected class indexes. For each identity, different random inputs are gradually updated by the network gradient into identity-preserved face images. The proposed ArcFace model obviously outperforms the baseline methods (e.g. softmax, SphereFace and CosFace) in the image quality, achieving the FID score of 70.39. By employing the powerful ArcFace model trained on IBUG-500K, we calculate all cosine similarities between real training faces and corresponding generated faces. The average cosine similarity of ArcFace is 0.6248, surpassing all the baseline models by a clear margin.

In Figure 14, we show the synthesized faces from the proposed ArcFace in comparison with the baseline CosFace model. As can be seen, ArcFace is able to reconstruct identity-preserved faces only by using the model parameters without training any additional discriminator and generator like in GAN [36]. Considering the image quality is only constrained by the classification loss and the BN priors, it is quite understandable that there exist some identity-unrelated artifacts in the generation results. Besides, there are many grey images in MS1MV3 and this statistic information is also stored in the BN parameters, thus some generated faces are not colorful. Compared to the baseline CosFace model, our ArcFace can depict better facial features of the real faces in terms of identity preservation and image quality.

**Open-set Face Generation.** In Table 11, we compare inversion results of different models on LFW. For each pre-trained model, we first calculate the embedding features of 13,233 face images from LFW, and then we generate faces constrained to these target features through a  $\ell_2$  loss. As we can see, ArcFace maintains best reconstruction quality and identity preservation, consistently outperforming the baseline models in both FID and average cosine similarity metrics. On the real faces of LFW, the ArcFace model (ResNet50) achieves 99.81% verification accuracy. On the generated faces, the verification accuracy slightly drops to 97.75% by using the same model ([MS1MV3, ResNet50, ArcFace]) for testing. For unbiased evaluation, we report the matching accuracy on LFW by employing the powerful ArcFace model (ResNet100) trained on IBUG-500K and this model is more susceptible to artifacts in the generated results. Even though there is a further drop in the verification accuracy (93.30%), the results compared to the baseline models further demonstrate the advantages of ArcFace in the inversion problem.

Figure 15 illustrates our synthesis from features of LFW faces

that contain appearance variations (e.g. age, gender, race, pose and occlusion). Similar to the previous experiment, our ArcFace model robustly depicts identity-preserved faces. The success of robustly handling with those challenging factors comes from two properties: (1) the ArcFace network was trained to ignore those facial variations in its embedding features, and (2) real face distributions stored in the BN layers can be effectively exploited for face image synthesis. Even though ArcFace can inverse most of the faces with realism and identity preservation, there exist some confusions during generation. In Figure 15(f), we show some inversion results from ArcFace containing gender confusions. Even though these confusions can be easily distinguished by human eyes, they exhibit high similarity from the view of the machine. In Figure 16, we further conduct an ablation study about ArcFace inversion without BN constraints. As we can see from these results, constraints from the BN layers can enforce the generated face more visually plausible. Without the BN constraints, the resulting face images lack natural image statistics and can be quite easily identified as unnatural.

## 5 CONCLUSIONS

In this paper, we first propose an Additive Angular Margin Loss function, named ArcFace, which can effectively enhance the discriminative power of deep feature embedding for face recognition. We further introduce sub-class into ArcFace to relax the intra-class constraint under massive real-world noises. The proposed sub-center ArcFace encourages one dominant sub-class that contains the majority of clean faces and non-dominant subclasses that include hard or noisy faces. This automatic isolation can be employed to clean large-scale web faces and we demonstrate that our method consistently outperforms the state of the art through the most comprehensive experiments. Apart from enhancing discriminative power, ArcFace can also strengthen the model’s generative power, mapping feature vectors to face images. The pre-trained ArcFace model can generate identity-preserved face images for both subjects inside and outside the training data only by using the network gradient and BN priors. As the proposed ArcFace inversion only focuses on approximating the target identity feature, the facial poses and expressions are not controllable. In the future, we will explore controlling intermediate neuron activations to target specific facial poses and expressions during inversion. In addition, we will also explore how to make the face recognition model not invertible so that face images cannot be easily reconstructed from model weights to protect privacy.

## ACKNOWLEDGMENT

We are thankful to NVIDIA for the hardware donation and Amazon Web Services for the cloud credits. The work of Jiankang Deng was partially funded by Imperial President’s PhD Scholarship. The work of Jing Yang was partially funded by the Vice-Chancellor’s PhD Scholarship from University of Nottingham. The work of Stefanos Zafeiriou was partially funded by the EPSRC Fellowship DEFORM: Large Scale Shape Analysis of Deformable Models of Humans (EP/S010203/1), FACER2VM: Face Matching for Automatic Identity Retrieval, Recognition, Verification and Management (EP/N007743/1), and a Google Faculty Award.



**Jiankang Deng** obtained his PhD degree from Imperial College London (ICL), supervised by Prof. Stefanos Zafeiriou and funded by the Imperial President's PhD Scholarships. His research topic is deep learning-based face analysis, including detection, alignment, reconstruction, recognition and generation etc. He is a reviewer in prestigious computer vision journals and conferences including T-PAMI, IJCV, CVPR, ICCV and ECCV. He is one of the main contributors to the widely used open-source platform Insightface. He is a student member of the IEEE.



**Irene Kotsia** received the PhD degree from the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2008. From 2008 to 2009, she was a research associate and teaching assistant with the Department of Informatics, Aristotle University of Thessaloniki. From 2009 to 2011, she was a research associate with the Department of Electronic Engineering and Computer Science, Queen Mary University of London, while from 2012 to 2014, she was a senior research associate with the Department of Computing, Imperial College London. From 2013 to 2020, she was a lecturer in creative technology and digital creativity with the Department of Computing Science, Middlesex University of London. She has been a guest editor of two journal special issues dealing with face analysis topics. She has co-authored more than 40 journal and conference publications in the most prestigious journals and conferences of her field (e.g., the IEEE Transactions on Image Processing, the IEEE Transactions on Neural Networks and Learning Systems, CVPR, ICCV). She has published one of the most influential works in facial expression recognition in the IEEE Transactions on Image Processing which has received around 700 citations. She is a member of the IEEE.



**Jia Guo** is an active contributor to the non-profit Github project InsightFace (2D and 3D face analysis).



**Jing Yang** is a Ph.D. candidate from Department of Computer Science, University of Nottingham. She is funded by the Vice-Chancellor's PhD Scholarship. Her research interest is deep face analysis and model compression.



**Niannan Xue** received the BA degree (first class) in theoretical physics from Cambridge University in 2013, and the MMath degree in applied mathematics from Cambridge University in 2014. He is currently working toward the PhD degree at Imperial College London. He was a visiting student in the Biological and Soft Systems Sector of the Cavendish Laboratory. He received St Catharine's Skerne Prize for three consecutive times. His research interests include data mining, machine learning and artificial intelligence. He is a student member of the IEEE.



**Stefanos Zafeiriou** is currently a Professor in Machine Learning and Computer Vision with the Department of Computing, Imperial College London, London, U.K, and an EPSRC Early Career Research Fellow. He served Associate Editor and Guest Editor in various journals including TPAMI, IJCV, TAC, CVIU, and IVC. He has been a Guest Editor of 8+ journal special issues and co-organised over 16 workshops/special sessions on specialised computer vision topics in top venues. He has co-authored 70+ journal papers mainly on novel statistical machine learning methodologies applied to computer vision problems, such as 2-D/3-D face analysis, deformable object fitting and tracking, shape from shading, and human behaviour analysis, published in the most prestigious journals in his field of research, such as TPAMI, IJCV, and many papers in top conferences, such as CVPR, ICCV, ECCV, ICML. His students are frequent recipients of very prestigious and highly competitive fellowships, such as the Google Fellowship x2, the Intel Fellowship, and the Qualcomm Fellowship x4. He has more than 20K+ citations to his work, h-index 64. He was the General Chair of BMVC 2017. He is a member of the IEEE.

## REFERENCES

- [1] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *NeurIPS*, 2014.
- [2] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014.
- [3] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.
- [4] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC*, 2015.
- [5] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *SIBGRAPI*, 2018.
- [6] G. Guo and N. Zhang, "A survey on deep learning based face recognition," in *CVIU*, 2019.
- [7] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," in *SPL*, 2016.
- [8] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *CVPR*, 2020.
- [9] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *FG*, 2018.
- [10] I. Masi, A. T. Tran, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?" in *ECCV*, 2016.
- [11] I. Masi, A. T. Tran, T. Hassner, G. Sahin, and G. Medioni, "Face-specific data augmentation for unconstrained face recognition," in *IJCVA*, 2019.
- [12] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NeurIPS*, 2016.
- [13] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017.
- [14] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *CVPR*, 2018.
- [15] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," in *SPL*, 2018.
- [16] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019.
- [17] J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," in *CVPR Workshop*, 2017.
- [18] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy, "The devil of face recognition is in the noise," in *ECCV*, 2018.
- [19] W. Hu, Y. Huang, F. Zhang, and R. Li, "Noise-tolerant paradigm for training face recognition cnns," in *CVPR*, 2019.
- [20] Y. Zhong, W. Deng, M. Wang, J. Hu, J. Peng, X. Tao, and Y. Huang, "Unequal-training for deep face recognition with long-tailed noisy data," in *CVPR*, 2019.
- [21] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei, "Co-mining: Deep face recognition with noisy labels," in *ICCV*, 2019.
- [22] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center arcface: Boosting face recognition by large-scale noisy web faces," in *ECCV*, 2020.
- [23] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman, "Synthesizing normalized faces from facial identity features," in *CVPR*, 2017.
- [24] C. Nhan Duong, K. Luu, K. Gia Quach, and T. D. Bui, "Beyond principal components: Deep boltzmann machines for face modeling," in *CVPR*, 2015.
- [25] C. N. Duong, K. Luu, K. G. Quach, and T. D. Bui, "Deep appearance models: A deep boltzmann machine approach for face modeling," in *IJCVA*, 2019.
- [26] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *CVPR*, 2016.
- [27] A. Zhmoginov and M. Sandler, "Inverting face embeddings with convolutional neural networks," *arXiv:1606.04189*, 2016.
- [28] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the reconstruction of face images from deep face templates," in *TPAMI*, 2018.
- [29] C. N. Duong, T.-D. Truong, K. Luu, K. G. Quach, H. Bui, and K. Roy, "Vec2face: Unveil human faces from their blackbox features in face recognition," in *CVPR*, 2020.
- [30] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," 2015.
- [31] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *CVPR*, 2015.
- [32] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *CVPR*, 1991.
- [33] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," in *CVPR*, 2020.
- [34] M. Haroush, I. Hubara, E. Hoffer, and D. Soudry, "The knowledge within: Methods for data-free model compression," in *CVPR*, 2020.
- [35] Y. Cai, Z. Yao, Z. Dong, A. Gholami, M. W. Mahoney, and K. Keutzer, "Zeroq: A novel zero shot quantization framework," in *CVPR*, 2020.
- [36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.
- [37] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *ECCV*, 2016.
- [38] J. Cao, Y. Li, and Z. Zhang, "Celeb-500k: A large training dataset for face recognition," in *ICIP*, 2018.
- [39] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv:1512.01274*, 2015.
- [40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NeurIPS Workshop*, 2017.
- [41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016.
- [42] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *ICML*, 2016.
- [43] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, "Metric learning with adaptive density discrimination," in *ICLR*, 2016.
- [44] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016.
- [45] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *ICCV*, 2017.
- [46] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *ICCV*, 2017.
- [47] Q. Qian, J. Tang, H. Li, S. Zhu, and R. Jin, "Large-scale distance metric learning with uncertainty," in *CVPR*, 2018.
- [48] X. Zhang, R. Zhao, J. Yan, M. Gao, Y. Qiao, X. Wang, and H. Li, "P2grad: Refined gradients for optimizing deep face models," in *CVPR*, 2019.
- [49] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "Adacos: Adaptively scaling cosine logits for effectively learning deep face representations," in *CVPR*, 2019.
- [50] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, "Adaptiveface: Adaptive margin and sampling for face recognition," in *CVPR*, 2019.
- [51] K. Zhao, J. Xu, and M.-M. Cheng, "Regularface: Deep face recognition via exclusive regularization," in *CVPR*, 2019.
- [52] Y. Duan, J. Lu, and J. Zhou, "Uniformface: Learning deep equidistributed representation for face recognition," in *CVPR*, 2019.
- [53] X. Wang, S. Zhang, S. Wang, T. Fu, H. Shi, and T. Mei, "Mis-classified vector guided softmax loss for face recognition," in *AAAI*, 2020.
- [54] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang, "Curricularface: adaptive curriculum learning loss for deep face recognition," in *CVPR*, 2020.
- [55] Y. Kim, W. Park, M.-C. Roh, and J. Shin, "Groupface: Learning latent groups and constructing group-based representations for face recognition," in *CVPR*, 2020.
- [56] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv:1411.7923*, 2014.
- [57] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," in *TIFS*, 2018.
- [58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [59] M. Zhu and A. M. Martínez, "Optimal subclass discovery for discriminant analysis," in *CVPR Workshops*, 2004.
- [60] M. Zhu and A. M. Martinez, "Subclass discriminant analysis," in *TPAMI*, 2006.
- [61] H. Wan, H. Wang, G. Guo, and X. Wei, "Separability-oriented subclass discriminant analysis," in *TPAMI*, 2017.
- [62] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, "Softtriple loss: Deep metric learning without triplet sampling," in *ICCV*, 2019.
- [63] R. Müller, S. Kornblith, and G. Hinton, "Subclass distillation," *arXiv:2002.03936*, 2020.
- [64] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang, "Faceid-gan: Learning a symmetry three-player gan for identity-preserving face synthesis," in *CVPR*, 2018.
- [65] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Towards open-set identity preserving face synthesis," in *CVPR*, 2018.
- [66] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Advancing high fidelity identity swapping for forgery detection," in *CVPR*, 2020.

- [67] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *ICLR*, 2019.
- [68] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *CVPR*, 2019.
- [69] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer, 2019.
- [70] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *CCCS*, 2015.
- [71] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” *arXiv:1506.06579*, 2015.
- [72] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *ECCV*, 2016.
- [73] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, “Range loss for deep face recognition with long-tail,” in *ICCV*, 2017.
- [74] 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.
- [75] T. Zheng and W. Deng, “Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments,” Tech. Rep., 2018.
- [76] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, “Agedb: The first manually collected in-the-wild age database,” in *CVPR Workshop*, 2017.
- [77] T. Zheng, W. Deng, and J. Hu, “Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments,” *arXiv:1708.08197*, 2017.
- [78] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, “The megaface benchmark: 1 million faces for recognition at scale,” in *CVPR*, 2016.
- [79] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. C. Adams, T. Miller, N. D. Kalka, A. K. Jain, J. A. Duncan, and K. Allen, “Iarpa janus benchmark-b face dataset,” in *CVPR Workshop*, 2017.
- [80] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, and J. Cheney, “Iarpa janus benchmark-c: Face dataset and protocol,” in *ICB*, 2018.
- [81] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, “Regularizing neural networks by penalizing confident output distributions,” *arXiv:1701.06548*, 2017.
- [82] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, “Normface:  $l_2$  hypersphere embedding for face verification,” *arXiv:1704.06369*, 2017.
- [83] R. Ranjan, C. D. Castillo, and R. Chellappa, “ $L_2$ -constrained softmax loss for discriminative face verification,” *arXiv:1703.09507*, 2017.
- [84] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song, “Learning towards minimum hyperspherical energy,” in *NeurIPS*, 2018.
- [85] M. Hardt and T. Ma, “Identity matters in deep learning,” *arXiv:1611.04231*, 2016.
- [86] E. Hoffer, I. Hubara, and D. Soudry, “Fix your classifier: the marginal value of training the last weight layer,” *arXiv:1801.04540*, 2018.
- [87] F. Pernici, M. Bruni, C. Baechi, and A. Del Bimbo, “Maximally compact and separated features with regular polytope networks,” in *CVPR Workshops*, 2019.
- [88] J. Deng, J. Guo, D. Zhang, Y. Deng, X. Lu, and S. Shi, “Lightweight face recognition challenge,” in *ICCV Workshop*, 2019.
- [89] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Tech. Rep., 2007.
- [90] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *CVPR*, 2011.
- [91] D. Han, J. Kim, and J. Kim, “Deep pyramidal residual networks,” *arXiv:1610.02915*, 2016.
- [92] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *ICML*, 2015.
- [93] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *JML*, 2014.
- [94] W. Xie, S. Li, and A. Zisserman, “Comparator networks,” in *ECCV*, 2018.
- [95] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, “Targeting ultimate accuracy: Face recognition via deep embedding,” *arXiv:1506.07310*, 2015.
- [96] H. Wang, D. Gong, Z. Li, and W. Liu, “Decorrelated adversarial learning for age-invariant face recognition,” in *CVPR*, 2019.
- [97] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Feature transfer learning for face recognition with under-represented data,” in *CVPR*, 2019.
- [98] B. Liu, W. Deng, Y. Zhong, M. Wang, J. Hu, X. Tao, and Y. Huang, “Fair loss: margin-aware reinforcement learning for deep face recognition,” in *ICCV*, 2019.
- [99] B.-N. Kang, Y. Kim, B. Jun, and D. Kim, “Attentional feature-pair relation networks for accurate face recognition,” in *ICCV*, 2019.
- [100] Y. Shi and A. K. Jain, “Probabilistic face embeddings,” in *ICCV*, 2019.
- [101] J. Chang, Z. Lan, C. Cheng, and Y. Wei, “Data uncertainty learning in face recognition,” in *CVPR*, 2020.
- [102] H. Zhao, X. Ying, Y. Shi, X. Tong, J. Wen, and H. Zha, “Rdcface: Radial distortion correction for face recognition,” in *CVPR*, 2020.
- [103] Q. Wang, T. Wu, H. Zheng, and G. Guo, “Hierarchical pyramid diverse attention networks for face recognition,” in *CVPR*, 2020.
- [104] Y. Shi, X. Yu, K. Sohn, M. Chandraker, and A. K. Jain, “Towards universal representation learning for deep face recognition,” in *CVPR*, 2020.
- [105] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, “Circle loss: A unified perspective of pair similarity optimization,” in *CVPR*, 2020.
- [106] A. Ali, M. Testa, T. Bianchi, and E. Magli, “Biometricnet: deep unconstrained face verification through learning of metrics regularized onto gaussian distributions,” in *ECCV*, 2020.
- [107] Y. Kim, W. Park, and J. Shin, “Broadface: Looking at tens of thousands of people at once for face recognition,” in *ECCV*, 2020.
- [108] X. Wang, S. Wang, C. Chi, S. Zhang, and T. Mei, “Loss function search for face recognition,” in *ICML*, 2020.
- [109] Y. Zhang, W. Deng, M. Wang, J. Hu, X. Li, D. Zhao, and D. Wen, “Global-local gcn: Large-scale label noise cleansing for face recognition,” in *CVPR*, 2020.
- [110] D. Cao, X. Zhu, X. Huang, J. Guo, and Z. Lei, “Domain balancing: Face recognition on long-tailed domains,” in *CVPR*, 2020.
- [111] H. Du, H. Shi, Y. Liu, J. Wang, Z. Lei, D. Zeng, and T. Mei, “Semi-siamese training for shallow face learning,” in *ECCV*, 2020.
- [112] H.-W. Ng and S. Winkler, “A data-driven approach to cleaning large face datasets,” in *ICIP*, 2014.
- [113] W. Xie and A. Zisserman, “Multicolumn networks for face recognition,” in *BMVC*, 2018.
- [114] R. Ranjan, A. Bansal, H. Xu, S. Sankaranarayanan, J.-C. Chen, C. D. Castillo, and R. Chellappa, “Crystal loss and quality pooling for unconstrained face verification and recognition,” *arXiv:1804.01159*, 2018.
- [115] J. Zhao, Y. Cheng, Y. Cheng, Y. Yang, F. Zhao, J. Li, H. Liu, S. Yan, and J. Feng, “Look across elapse: Disentangled representation learning and photorealistic cross-age face synthesis for age-invariant face recognition,” in *AAAI*, 2019.
- [116] Q. Zhang, J. Li, M. Yao, L. Song, H. Zhou, Z. Li, W. Meng, X. Zhang, and G. Wang, “Vargnet: Variable group convolutional neural network for efficient embedded computing,” *arXiv:1907.05653*, 2019.
- [117] X. Li, F. Wang, Q. Hu, and C. Leng, “Airface: lightweight and efficient model for face recognition,” in *ICCV Workshops*, 2019.
- [118] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *ICML*, 2019.
- [119] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv:1503.02531*, 2015.
- [120] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980*, 2014.
- [121] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *NeurIPS*, 2017.