DAM: Discrepancy Alignment Metric for Face Recognition

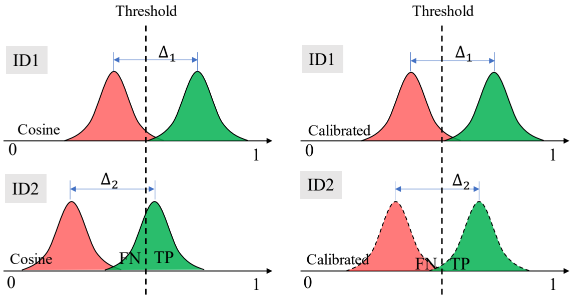
Jiaheng Liu*[1], Yudong Wu*[2], Yichao Wu†[2], Chuming Li[2], Xiaolin Hu[3], Ding Liang[2], Mengyu Wang[2]
[1]Beihang University, [2]SenseTime Group Limited, [3]Tsinghua University
liujiaheng@buaa.edu.cn, {wuyudong, wuyichao, lichuming, liangding}@sensetime.com,
1600017843@pku.edu.cn, xlhu@mail.tsinghua.edu.cn

Abstract

The field of face recognition (FR) has witnessed remarkable progress with the surge of deep learning. The effective loss functions play an important role for FR. In this paper, we observe that a majority of loss functions, including the widespread triplet loss and softmax-based cross-entropy loss, embed inter-class (negative) similarity s_n and intra-class (positive) similarity s_p into similarity pairs and optimize to reduce $(s_n - s_p)$ in the training process. However, in the verification process, existing metrics directly take the absolute similarity between two features as the confidence of belonging to the same identity, which inevitably causes a gap between the training and verification process. To bridge the gap, we propose a new metric called Discrepancy Alignment Metric (DAM) for verification, which introduces the Local Inter-class Discrepancy (LID) for each face image to normalize the absolute similarity score. To estimate the LID of each face image in the verification process, we propose two types of LID Estimation (LIDE) methods, which are reference-based and learning-based estimation methods, respectively. The proposed DAM is plug-and-play and can be easily applied to the most existing methods. Extensive experiments on multiple popular face recognition benchmark datasets demonstrate the effectiveness of our proposed method.

1. Introduction

Face recognition based on deep learning has been well investigated for decades [31, 35, 42]. Most of the progress depends on large-scale training data [9, 50, 16], deep neural network architectures [37, 10, 11], and effective loss function designs [26, 6, 41, 39, 53, 27, 4, 25, 13, 7]. Despite many efforts, most prior works use the sample-to-sample absolute similarity metric during inference. The identities are determined by directly thresholding cosine or L2 distance for each face image pair. It is inconsistent with the training process, which optimizes the relative margin between the intra-class and the inter-class similarities.



(a) (b)

Figure 1: The similarity distributions on different evaluation metrics of pairs from different IDs. The green histogram represents the positive pairs, and the red histogram represents the negative pairs. (a): Distribution of cosine similarity metric. ID1 and ID2 have the same margin between positive and negative pairs, whereas the overall similarity of pairs of ID2 is less than ID1, which leads to judge a large number of positive pairs of ID2 as false negatives. (b): Distribution of our proposed Discrepancy Alignment Metric (DAM). The scores of positive pairs of ID2 are calibrated to a higher level.

Specifically, the popular softmax-based loss functions (e.g., ArcFace [5], CosFace [41]) or metric learning based loss functions (e.g., Triplet loss [27]) seek to reduce $s_n - s_p$ or $s_n - s_p + m$ as the optimization target [32], where s_p is the intra-class (positive) similarity, s_n refers to inter-class (negative) similarity and m is the margin term to enhance the discriminative ability. Therefore, the relative score $s_p - s_n$ indicates the optimization degree for each face image during the training process. It works well for common close-set classification, which maintains category weights of the classifier by relative probabilities, i.e., $c_{\text{pred}} = \arg\max_i \{\frac{e^{z_i}}{\sum e^{z_j}}\}_{i=1}^{C}$. However, in open-set face

*Equal contribution
†Corresponding author

recognition[1], the absolute cosine similarity between the pair of features $s = \langle \boldsymbol{f}_1, \boldsymbol{f}_2 \rangle$ is considered as the probability of having the same identity (ID), which causes the gap between the training and verification process. The z_i and z_j are the logits predicted by the classifier, $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are the face embeddings extracted from the neural network, and c_{pred} is the predicted class label from C classes.

A snapshot of a typical example is also shown in Fig. 1a, where two IDs have the same margin between the positive pairs and the negative pairs (i.e., $\Delta_1 = \Delta_2$). It means that they are optimized to the same degree during the training process. In contrast, the overall absolute cosine similarities of pairs of ID2 are less than ID1. When applying the optimized model in practice, a large number of positive pairs of ID2 are judged as false negatives. In addition, the long-tail or non-uniform distributions usually exist in the training data of real-world face recognition. Therefore, the optimized model is biased to different domains [1], which exacerbates the gap. Consequently, a more accurate metric, which is more consistent with the existing loss functions, is needed to compensate for the gap between the training and inference for face recognition.

Motivated by the above analysis, we propose a new metric called **Discrepancy Alignment Metric (DAM)**, which aims to bridge the gap between the training and verification process for face recognition. First, we analyze the aforementioned gap and introduce the DAM, which incorporates the Local Inter-class Discrepancy (LID) of each feature to normalize the absolute similarity score, and is more consistent with the current popular loss functions. Then, we introduce two types of **Local Inter-class Discrepancy Estimation (LIDE)** methods, which are the reference-based LIDE and learning-based LIDE methods, respectively. In our LIDE, we propose to randomly sample from the training set or employ GAN to generate a set of images from diverse identities to build the anchor image set. For the reference-based LIDE method, the neighbors from the anchor image set are searched in the feature space to estimate the LID for each face image, which is flexible without tuning the optimized models. For the learning-based LIDE method, we directly leverage a learnable regression module to regress the LID for each face image, which avoids the need of an anchor image set during the verification process.

The contributions of our paper are as follows:

1) We are the first to investigate the gap between the training and verification process of face recognition, and propose a new metric called Discrepancy Alignment Metric, which is plug-and-play and can be readily integrated into existing face recognition methods.

2) The Local Inter-class Discrepancy (LID) of each feature is incorporated into the new metric to normalize the similarity, and two types of Local Inter-class Discrepancy Estimation (LIDE) methods are introduced, including the reference-based and learning-based LIDE methods.

3) Extensive experiments on multiple benchmark datasets show that our proposed DAM significantly improves the performance of face recognition.

2. Related works

Overview of Face Recognition. There are three essential factors for face recognition including *network architecture* [37, 34, 31, 35, 30, 36, 33], *large-scale dataset* [9, 50, 16] and *effective loss function* [27, 46, 51, 20, 19, 41, 39, 6, 25, 13, 49]. First, with the process of neural network architecture, many hand-designed networks (e.g., VGGNet [30], GoogleNet [36] and ResNet [10]) also achieves promising performance for face recognition. Meanwhile, Neural Architecture Search (NAS) was proposed to relieve the burden from the hand-crafted network design process, and its effectiveness has been demonstrated in many computer vision tasks [56, 18]. For the large-scale datasets, many widely-used face recognition datasets are proposed to improve the generalization ability of face recognition. For the loss function, as face recognition is usually under the open-set protocol in the real-world scenarios, most face recognition approaches adopt the metric learning based loss functions. For example, Triplet loss [27] utilizes Euclidean distance to measure similarity score for each face image pair. Center loss [46] and range loss [51] are proposed to reduce intra-class variations via minimizing distances within each class. However, constraining margin in Euclidean space is insufficient to achieve optimal generalization. Therefore, many angular-margin based loss functions are proposed to tackle the problem, where angular constraints are integrated into the softmax cross-entropy loss function to improve the learned face representation in L-softmax [20] and A-softmax [19]. In addition, CosFace [41], AM-softmax [39] and ArcFace [6] additionally maximize angular margins when compared above methods. Overall, the existing loss functions seek to maximize the intra-class similarities and reduce the inter-class similarities, where the optimization target in the training process is not compatible with the cosine similarity in the verification process.

Verification Metric for Face recognition. In the beginning, instead of using distance or similarity metric, the SVM [35, 37] and Joint Bayesian [3, 34] model are utilized as classifiers to determine whether a pair of images have the same identity. Recently, deep CNN is utilized to extract the feature embedding for each image. Metric learning loss (e.g., contrastive loss [4], triplet loss [27], the squared L2 distance) between features turned to be the ver-

[1] In this paper, “open-set face recognition” and “face recognition” can be used interchangeably.

ification similarity metric [20, 33, 27] with the widespread application of CNN [30, 36]. Besides, the concept of angle and hyper-sphere manifold are proposed due to the optimal generalization and discriminative representation, where the features are explicitly normalized, and cosine similarity is used as verification metric [41, 6, 39]. In addition, several methods considered each face image as probabilistic distributions [28, 29, 2] in the feature space, where corresponding distribution-based similarity metrics (e.g., uncertainty-aware log-likelihood score [29]) are proposed.

3. Methodology

In this section, we describe our proposed DAM as shown in Fig. 2. Specifically, given a pair of face images, we first use a pre-trained neural network to extract the feature for each face image, and then generate the Local Inter-class Discrepancy (LID) for each face image by our proposed Local Inter-class Discrepancy Estimation (LIDE) method. Finally, the DAM takes the features and the LIDs of a pair of images as input, and generates the similarity score for this pair of face images.

3.1. Discrepancy Alignment Metric

In this section, we first analyze the gap between the existing loss functions and evaluation metrics for face recognition. Then we propose the Discrepancy Alignment Metric (DAM) to evaluate the similarity for a pair of face images. Finally, we theoretically analyze why our DAM is more consistent with existing loss functions by showing the relationship of our DAM with the NormFace [40], which is a common loss function for face recognition.

In general, current face recognition methods [4, 27, 24, 6] tend to extract the feature embedding for each face image by using a backbone network, and adopt effective loss functions to increase the discriminative ability of the feature embedding. During training, the well-defined loss functions [4, 27, 24, 6] aim to minimize the difference between the inter-class similarity and the intra-class similarity, which is a relative-based optimization target. However, for open set evaluation of face recognition, the cosine similarity metric is commonly used to measure whether two face images belong to the same identity, which is an absolute metric for face recognition. Therefore, we observe a natural gap between the loss function for training and the cosine similarity metric for evaluation.

To illustrate this gap more clearly, as many popular cosine-based loss functions [19, 41, 6] can be considered as the variants of NormFace loss [40], we take the NormFace loss [40] as an example. Additionally, we can obtain similar conclusions using triplet loss [27].

In [40], there is totally C classes in the training set. The NormFace loss is defined as follows:

$$\mathcal{L}_{\text{NormFace}} = -\log \frac{e^{s z_{i,y_i}}}{\sum\limits_{j=1}^{C} e^{s z_{i,j}}}, \tag{1}$$

where y_i is the class label for the feature embedding $\boldsymbol{f}_i$, and $z_{i,j} = \cos(\theta_{i,j})$. Here, $\theta_{i,j}$ is the angle between the j-th class weight vector $\boldsymbol{w}_j$ and $\boldsymbol{f}_i$, and s is a positive scale hyper-parameter to adjust the scale of $z_{i,j}$.

We can rewrite the NormFace loss function in the following way:

$$\mathcal{L}_{\text{NormFace}} = -\log \frac{1}{\sum\limits_{j=1, j\neq y_i}^{C} e^{s(z_{i,j} - z_{i,y_i})} + 1}. \tag{2}$$

Theoretically, the above loss function is intrinsically relative-based. Specifically, the loss function aims to minimize the difference (i.e., $z_{i,j} - z_{i,y_i}$) between inter-class $z_{i,j}$ and intra-class similarity z_{i,y_i}, which may cause an ambiguous optimization gap in the verification process in Fig. 1. Therefore, a more accurate metric, which is consistent with the existing loss functions, is needed to compensate such gap between the training and verification for face recognition.

Instantiation of DAM.

In our work, we propose a new metric named as *Discrepancy Alignment Metric* (DAM) to measure the similarity score for a pair of face embeddings (i.e., $\boldsymbol{f}_1$, $\boldsymbol{f}_2$), which aims to be consistent with the existing loss functions. The DAM is defined as follows:

$$\text{DAM}(\boldsymbol{f}_1, \boldsymbol{f}_2) = e^{s\langle \boldsymbol{f}_1, \boldsymbol{f}_2 \rangle} \cdot \left(\frac{1}{\mathcal{G}(\boldsymbol{f}_1)} + \frac{1}{\mathcal{G}(\boldsymbol{f}_2)}\right), \tag{3}$$

where s is the scale hyper-parameter, $\langle \cdot, \cdot \rangle$ is the inner product of two face image embeddings, $\mathcal{G}(\boldsymbol{f}_1) = \sum\limits_{i=1}^{k} e^{s\langle \boldsymbol{f}_1, \boldsymbol{f}_1^i \rangle}$ and $\mathcal{G}(\boldsymbol{f}_2) = \sum\limits_{i=1}^{k} e^{s\langle \boldsymbol{f}_2, \boldsymbol{f}_2^i \rangle}$ denote the local inter-class discrepancy for $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$, respectively. Here, $\boldsymbol{f}_1^i$ ($\boldsymbol{f}_2^i$) denote the i-th neighboring embedding for $\boldsymbol{f}_1$ ($\boldsymbol{f}_2$) in the feature space. In our work, we define $\Psi_{\boldsymbol{f}_1} = \{\boldsymbol{f}_1^i\}_{i=1}^k$ and $\Psi_{\boldsymbol{f}_2} = \{\boldsymbol{f}_2^i\}_{i=1}^k$ to denote the neighboring embedding set for $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ in the feature space, respectively, where the size of the neighboring embedding set is k.

Then, we discuss why our proposed DAM is more consistent with the loss function. We also take NormFace as an example in Eq. 2, and minimize the loss function $\mathcal{L}_{\text{NormFace}}$ in the training process. Meanwhile, we can formulate the Eq. 2 as following optimization task:

$$\min \sum_{j=1, j\neq y_i}^{C} e^{s(z_{i,j} - z_{i,y_i})}. \tag{4}$$

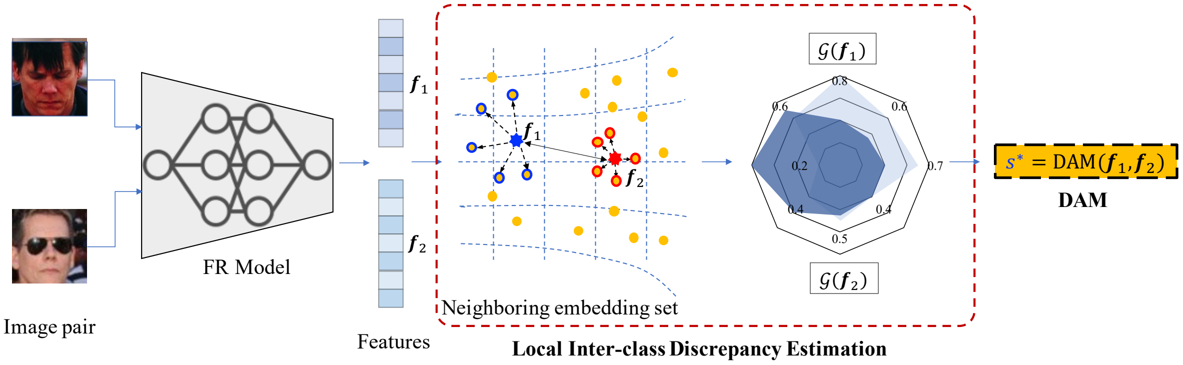


Figure 2: The face verification process of DAM. First, we use a pre-trained face recognition (FR) model to extract the features for a pair of images. Then, we generate the Local Inter-class Discrepancy (LID) for each face image by using our proposed Local Inter-class Discrepancy Estimation (LIDE) method (Here, we take the reference-based LIDE as an example.). Finally, DAM takes both the extracted features and the LIDs from this pair as input, and returns the similarity score for face verification.

The objective in Eq. 4 can be reformulated as follows:

$$\max \quad \frac{e^{sz_{i,y_i}}}{\sum\limits_{j=1,j\neq y_i}^{C} e^{sz_{i,j}}}, \tag{5}$$

where the z_{i,y_i} can be considered as the intra-class similarity, and the $z_{i,j}$ $(j \neq y_i)$ can be considered as the inter-class similarity.

Therefore, given a pair of face embeddings $\boldsymbol{f}_1$, $\boldsymbol{f}_2$, we can easily generate the intra-class similarity by computing the cosine similarity score, and an ideal metric should involve the inter-class similarities between the current embedding and other class weights in Eq. 5. Besides, we call the denominator in Eq. 5 as the inter-class discrepancy. In Eq. 5, the inter-class similarities between the embedding of each face image and most other class weights are all around 0, so the effect of most other classes on the inter-class discrepancy is very small [52]. As shown in Fig. 3, the curve of inter-class cosine similarity for each sample becomes flat quickly and decays to 0, which indicates that only several closest classes dominate the inter-class discrepancy information for each sample. Hence, we factorize the inter-class discrepancy (i.e., the denominator in Eq. 5) into two items: $\sum\limits_{j=1,j\neq y_i}^{k} e^{sz_{i,j}}$ and $\sum\limits_{j=k+1,j\neq y_i}^{C} e^{sz_{i,j}}$. The first item consists of similarities between each sample and the corresponding k closest inter-class neighbors in the embedding space. Meanwhile, the similarities in the second term are approaching 0, so the value of the second term can be simplified as $C-k-1$, where C is the number of classes. Therefore, we directly adopt the k-closest neighboring embeddings as the local inter-class discrepancy (LID) $\mathcal{G}(\boldsymbol{f})$ for each face embedding $\boldsymbol{f}$ in Eq.3, which aims to approximate the overall inter-class discrepancy information.

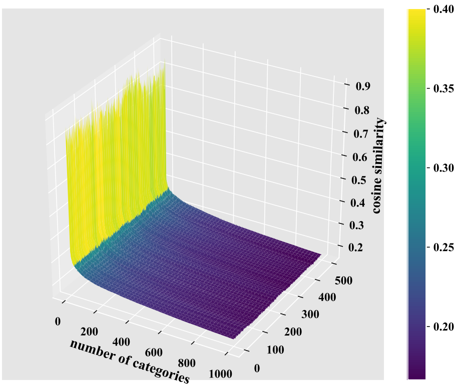


Figure 3: The cosine similarity distribution along category of each sample. We train the ResNet-50 [10] on MS-Celeb-1M [9] using ArcFace [6], and randomly select 500 samples. Cosine similarity distribution of each sample along category is sketched, and one L-shaped curve represents the cosine similarity distribution along category of a sample, i.e., $(\cos\theta_{i,1}, \cos\theta_{i,2}, ..., \cos\theta_{i,C})$. The $\cos\theta_{i,j}$ is the angle between the face embedding of the i-th sample and the weight of the j-th class center. We tile all samples' curves along the y-axis. The similarities with other categories are descending sorted.

3.2. Local Inter-class Discrepancy Estimation

Accordingly, how to acquire the local inter-class discrepancy (LID) $\mathcal{G}(\boldsymbol{f})$ for each face embedding $\boldsymbol{f}$ during the inference is crucial in Eq.3. In this section, we introduce two types of Local Inter-class Discrepancy Estimation (LIDE) methods (i.e., the reference-based LIDE and the learning-based LIDE methods) to estimate the LID for each face embedding.

Reference-based LIDE. As it is difficult to estimate the entire embedding distribution of the face recognition model in the feature space, we propose to sample from the embedding distribution. Specifically, the face recognition model extracts the feature embedding $\boldsymbol{f}$ for each face image I as follows:

$$\boldsymbol{f} = \mathcal{M}_{Face}(I), \quad \text{for } I \in \mathcal{I} \text{ and } \boldsymbol{f} \in \mathcal{E}, \tag{6}$$

where $\mathcal{I}$ and $\mathcal{E}$ are the image space and the embedding space, respectively, and $\mathcal{M}_{Face}$ is the face recognition model. Hence, by uniformly sampling a certain number of face images in the image space $\mathcal{I}$ and then projecting to the embedding space $\mathcal{E}$, we can obtain a sampling from the embedding distribution of the face recognition model. We call the sampled image set as the "anchor image set", which is denoted as $\mathcal{A}_I$. The corresponding embedding set is named as "anchor embedding set", which is denoted as $\mathcal{A}_E$. In the face verification process of each face pair, we extract the feature embedding $\boldsymbol{f}$ for each face image, and search the neighboring embedding set $\Psi_{\boldsymbol{f}}$ from the "anchor embedding set" for each face image. Then, we generate the LID $\mathcal{G}(\boldsymbol{f})$, and compute the similarity score of this face pair by the DAM in Eq.3. The algorithm is shown in Algorithm 1.

Algorithm 1 Reference-based LIDE.

```
Require:
    two face images $I_1$ and $I_2$;
    the trained FR model $\mathcal{M}_\theta$;
    the anchor embedding set $\mathcal{A}_E$;
Ensure:
    normalized similarity score $s^*$ of the two face images;
Extract features of two images, i.e., $\boldsymbol{f}_1 = \mathcal{M}_\theta(I_1)$, $\boldsymbol{f}_2 = \mathcal{M}_\theta(I_2)$;
Acquire the LID $\mathcal{G}(\boldsymbol{f}_1)$ by searching for the top-k maximum similarity scores of $\boldsymbol{f}_1$ in the $\mathcal{A}_E$;
Acquire the LID $\mathcal{G}(\boldsymbol{f}_2)$ by searching for the top-k maximum similarity scores of $\boldsymbol{f}_2$ in the $\mathcal{A}_E$;
return $s^* = \text{DAM}(\boldsymbol{f}_1, \boldsymbol{f}_2)$;
```

However, it is difficult to achieve uniform sampling, and the purpose of the anchor image set is to generate the feature embedding distribution of the face recognition model. In practice, a straight-forward way is to randomly sample from the training set to construct the anchor image set $\mathcal{A}_I$, and extract the features of the anchor image set by face recognition model to build the anchor embedding set $\mathcal{A}_E$. We call the anchor feature set as the "real-db". In addition, we also propose to use fake face images generated by GAN [8, 15] to construct the anchor embedding set, which is called as "fake-db". Compared with the "real-db", generated fake images do not contain private information, which is more conducive to practical use and dissemination. Besides, the identities of the "fake-db" will not conflict with the samples in the testing dataset. In the following experiment section, the "fake-db" is shown to have comparable performance with the "real-db". Furthermore, in our experiments, the effectiveness of "fake-db" also shows that the specific identity is not important to estimate the local inter-class discrepancy.

Meanwhile, compared to the original verification process, the external computational overhead is the process of searching for the neighboring embedding set. However, in practice, the size of the anchor embedding set is usually small (no more than 100000), and many off-the-shelf nearest neighbor search libraries can be used to reduce the time consumption [21, 14], so the external computational cost is accessible for face recognition. The complete face verification process is also shown in Fig. 2.

Learning-based LIDE. In addition to obtaining the LID by querying the anchor embedding set, we also propose a learning-based LIDE method, where the LID $\mathcal{G}(\boldsymbol{f})$ is directly predicted by the neural network for each face embedding $\boldsymbol{f}$. Specifically, a learnable local inter-class discrepancy regression (LIDR) module denoted as $\mathcal{M}_{LIDR}$ is adopted to learn the $\mathcal{G}(\boldsymbol{f})$. Formally, the loss function is defined as follows:

$$\mathcal{L}_{LIDR} = \|\mathcal{M}_{LIDR}(\boldsymbol{f}) - \mathcal{G}(\boldsymbol{f})\|^2 \tag{7}$$

The LIDR module consists of two fully connected layers with ReLU activation function, so the additional computational overhead is also negligible. LIDR takes the face embedding $\boldsymbol{f}$ as input and predicts the local inter-class discrepancy $\mathcal{G}(\boldsymbol{f})$ for $\boldsymbol{f}$. Inspired by [29], a stage-wise training strategy is adopted. Specifically, we first pre-train the face recognition model. Then, we fix the parameters of the pre-trained face recognition model and only optimize the LIDR module to learn the $\mathcal{G}(\boldsymbol{f})$. Besides, the LIDR module is trained on the same dataset of the face recognition model, so the stage-wise training strategy provides a fair comparison between the proposed method and the original method.

Plug-and-play. Overall, it is convenient to incorporate our proposed DAM into the existing methods for face recognition. First, the proposed DAM does not change the way of feature extraction. Thus, DAM can be combined with modern network architectures, such as VGGNet [30], GoogleNet [36] and ResNet [10]. Second, most FR loss functions [19, 41, 6, 45, 54] aims to learn discriminative and deterministic representations, and DAM does not change the feature distribution on the hypersphere, which means that DAM is easy to improve the performance of different loss functions with an extra local inter-class discrepancy estimation process in the verification stage.

4. Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of our proposed DAM. Then, we conduct a detailed ablation study to further analyse the contributions of different components in our DAM.

4.1. Implementation details

Datasets. For the training datasets, we employ the CASIA-WebFace [50] and the refined version of MS-Celeb-1M [9] provided by [6]. For the testing datasets, we use the following benchmark datasets: LFW [12], CALFW [55], YTF [48], IJB-B [47], IJB-C [22], and RFW datasets [44].

Experiments setting For pre-prepossessing, we follow the recent works [6, 19, 41, 17, 5] to generate the normalised face crops (112 × 112). For the backbone network, we utilize the widely used neural networks(e.g., ResNet-18, ResNet-50, ResNet-100 [10]), in which we follow [6] to leverage BN-Dropout-FC-BN network structure to produce 256-dim embedding feature representation. By default, the size of the anchor image set is 50000, the size of the neighboring embedding set is 10, and the s is 1. We utilize the SGD algorithm with a momentum of 0.9 and weight decay of 5×10^{-4}. For all experiments, we first pre-train the backbone network using the existing loss functions (*e.g.*, ArcFace, CosFace). For the pre-training on the CASIA-WebFace, the initial learning rate is 0.1, and is divided by 10 at the 20k, 30k, 35k iterations. The total iteration is 40k. For the pre-training on MS-Celeb-1M, the initial learning rate is 0.1 and divided by 10 at the 100k, 140k, 160k iterations. The total iteration is 200k. As for the learning-based method in LIDE, we utilize two fully-connected layers with ReLU activation function [23] as the regression network. The initial learning rate for the regression network is 0.001, and is divided by 0.1 at the 15k, 20k, 25k iterations. The total iteration is 30k. The batch size of all experiments is set as 512. Besides, we use StyleGAN [15] to generate the anchor image set. In addition, we use our proposed reference-based LIDE method and learning-based LIDE method based on the pre-trained network, which are called as **DAM-R** and **DAM-L** in the following experiments, respectively.

4.2. Results on IJB-B and IJB-C datasets

We provide the results of DAM on challenging IJB-B [47] and IJB-C [47] datasets. Since our method can be readily integrated into existing loss functions, we provide detailed experiments based on CosFace [41], ArcFace [6] and CurricularFace [26]. The backbone network is ResNet-100 trained on MS-Celeb-1M [9]. As shown in Table 1, both the DAM-R and the DAM-L methods achieve significant performance improvements on IJB-B and IJB-C datasets in all cases when compared with original baselines, which indicates our proposed methods are robust for different loss functions. Besides, the performance of our proposed DAM-L is comparable with the DAM-R, which means that our learning-based approach DAM-L can learn to estimate the local inter-class discrepancy well.

Table 1: The TAR results on IJB-B and IJB-C datasets with different loss functions.

Method	IJB-B (@FAR=1e-4)	IJB-C (@FAR=1e-4)
CosFace [41]	94.80	96.37
+DAM-R	**94.97**	**96.45**
+DAM-L	94.87	96.43
ArcFace [6]	94.25	95.63
+DAM-R	**94.63**	**95.78**
+DAM-L	94.54	95.73
CurricularFace [26]	94.81	96.11
+DAM-R	**95.12**	**96.20**
+DAM-L	95.01	96.18

4.3. Results on LFW, CALFW and YTF datasets

To further demonstrate the effectiveness of our method, we provide the results on LFW [12], CALFW [55] and YTF [48] in Table 2. Specifically, we leverage our proposed DAM-R and DAM-L methods based on pre-trained ResNet-18 on CASIA-WebFace dataset using ArcFace loss function [6]. As shown in Table 2, when compared with the original method based on ArcFace, the proposed DAM-R improves the accuracy by +0.30% on LFW, +0.38% on CALFW, and +0.26% on YTF, respectively.

Table 2: The verification accuracy (%) on the LFW, CALFW and YTF datasets.

Methods	LFW(%)	CALFW	YTF(%)
ArcFace [6]	98.73	91.67	94.97
+DAM-R	**99.03**	**92.05**	**95.23**
+DAM-L	98.98	92.03	95.17

4.4. Results on RFW dataset

To show the effect of DAM on non-uniform distribution (e.g., different races), we follow the setting of [43] to report the results of DAM on RFW [44] with the ResNet-34 model based on ArcFace loss function and using the BUPT-Balancedface [43] as the training set, where RFW contains faces from four race groups (African, Asian, Caucasian, and Indian). In Table 3, DAM also achieves superior results on all races in the RFW dataset, which demonstrates the effectiveness of our proposed DAM.

Table 3: The verification accuracy (%) on RFW.

Methods	Caucasian	Indian	Asian	African	Avg
ArcFace	96.13	94.70	93.75	93.95	94.63
+DAM-R	**96.30**	**95.20**	**94.31**	**94.51**	**95.08**
+DAM-L	96.20	95.11	94.15	94.32	94.95

4.5. Ablation study

The size of the neighboring embedding set and anchor embedding set. We evaluate our DAM-R method using different sizes of the neighboring embedding sets and the anchor embedding sets, and results on the IJB-B dataset are shown in Fig 4. Specifically, we leverage the ResNet-50 trained on MS-Celeb-1M dataset based on the ArcFace loss function. In Fig. 4a, we set the size of the anchor embedding set as 50000, and use different sizes of the neighboring embedding sets. When the size of the neighboring embedding set increases from 1 to 10, our method achieves better performance, which indicates that the neighboring embedding set could represent the LID appropriately. However, when the size of the neighboring embedding set continues to increase from 10, the performance on the IJB-B dataset begins to drop. It is reasonable that as the size increases, the LID of each face image tends to be similar, and the discriminative ability is diminished. Meanwhile, in Fig. 4b, we set the size of the neighboring set as 10, and use different sizes of the anchor embedding set. As the size of the anchor embedding set increases, the performance first gradually improves, and then tends to be flat. The anchor embedding set is to sample from the feature distribution for the trained model. When we increase the size of the anchor embedding set, we can obtain more accurate sampling, which helps to generate a better estimation of LID. Whereas, when the size is large enough, the improvement of performance becomes relatively stable.

The effect of the hyper-parameter s. To demonstrate the effect of the hyper-parameter s, we conduct more experiments by setting different values of s on the IJB-B dataset, and the results at FAR=0.001% are shown in Fig. 4c. Specifically, we leverage the ResNet-50 trained on MS-Celeb-1M dataset based on the ArcFace loss function. As shown in Fig. 4c, when the s increases from 0.5 to 1, the performance on IJB-B becomes better. However, when we continue to increase the s from 1, the performance begins to drop. In the training process, as analyzed in AdaCos [53], the scale factor of the loss function aims to balance the difficulty of the optimization, where the larger scale factor leads to an easier optimization goal. In contrast, in our DAM, if we use a large scale factor, the value of the local inter-class discrepancy is dominated by very few restricted neighbor samples (e.g., only the nearest one), which cannot reflect inter-class discrepancy well. Meanwhile, with $s \to 0$, the local inter-class discrepancy becomes indiscriminative.

Different types of anchor embedding sets. To analyze the effect of the anchor embedding set, we leverage different types of anchor embedding sets for our DAM-R method, and the results on the IJB-B dataset are reported in Table 4. The ResNet-50 model is adopted and trained on MS-Celeb-1M dataset. Specifically, "ArcFace" denotes the original result based on Arcface loss. "Weights of FC layers" denotes that we use the converged weights of the last fully-connected layer of ArcFace loss function. The weights approximate the centers of all classes of the training dataset. "Real-db" means that we randomly sample one image per identity from MS-Celeb-1M dataset, and extract features by the ResNet-50 model. "Fake-db" means that the images are generated by StyleGAN [15] trained on the MS-Celeb-1M dataset. As shown in Table 4, similar results are achieved when using different types of anchor embedding sets, which shows that our method is not sensitive to the types of anchor embedding sets.

Table 4: The results on IJB-B dataset when using different types of anchor embedding sets.

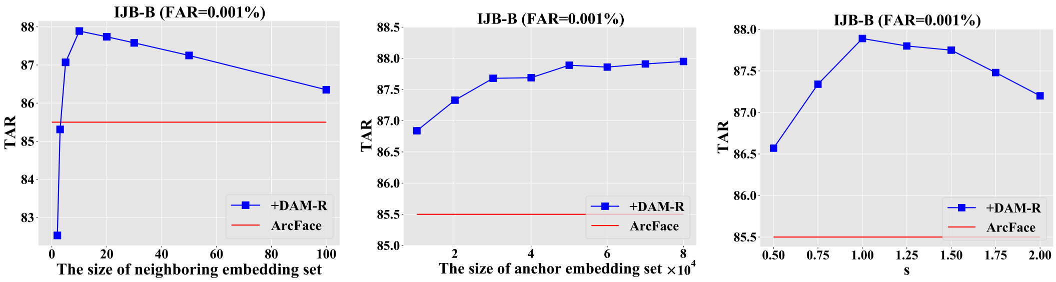
Types of anchor image set	IJB-B (TAR@FAR)	
	0.001%	0.01%
ArcFace [6]	85.50	93.09
Weights of FC layers	87.50	93.48
Real-db	87.86	**93.64**
Fake-db	**87.89**	93.63

4.6. Further analysis

Statistical analysis of cosine similarity and probability. We train the ResNet-50 [10] with MS-Celeb-1M [9]. As shown in the first two columns of Fig. 5, we visualize the cosine similarity score distribution corresponding to the positive category center and probability (output of softmax) distribution at different optimization steps. Then we select two disjoint segments in the probability distribution (the second column), and show their corresponding cosine distributions in the last column of Fig. 5. The two cosine distributions overlap each other. We have two observations from Fig. 5: 1) As the training proceeds, the curves of both score distribution and probability become sharp. 2) Meanwhile, the overlap of the selected distribution does not disappear with the convergence of the model, which shows the gap between the cosine similarity and probability.

Effectiveness of DAM. To analyze the effect of the DAM, in Fig. 6, we visualize the similarity score computed by DAM of the same samples in the two disjoint segments in Fig. 5. Compared with the original cosine similarity, the overlap of normalized score decreases, and the curve becomes sharper, as shown in Fig. 6. It demonstrates that the misalignment between probability and similarity is reduced with the incorporation of local inter-class discrepancy.

Differences with cohort score normalization. Cohort score normalization (CSN) [38] has been used for face recognition by post-processing the raw matching score using the cohort samples. The differences of our proposed DAM and CSN are as follows. First, for a pair of face images, CSN utilizes the similarity scores between each face image with respective neighboring face images as an addi-



(a) (b) (c)

Figure 4: (a) The effect of the size of neighboring embedding set. (b) The effect of the size of anchor image set. (c) The effect of the hyper-parameter s.

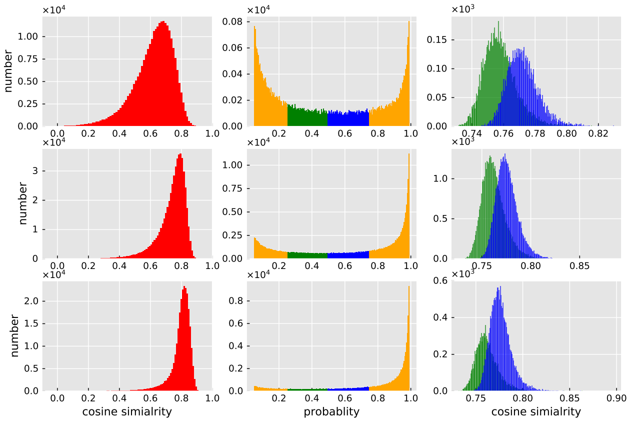


Figure 5: The distributions of softmax probability and cosine similarity between the training sample and its positive category at different optimization stages. The total iteration is 20,0000 steps, and the first row, the second row and the third row show the distributions of the 80,000th, 140,000th, and 200,000th steps, respectively. Two disjoint segments of the probability distribution are selected, and their corresponding cosine distributions are demonstrated in the last column.

tional discriminative feature to assist recognition. The motivation of CSN is to generate more convincing features for face verification. In contrast, we propose a new metric for inference, which is more consistent with optimization target in the training process. The similarities with neighbors in DAM are used to estimate the local inter-class discrepancy instead of additional representation in CSN. Second, CSN tries to exploit the patterns from sorted similarities and needs regression strategies to produce discriminative information. Whereas, our DAM is plug and play following Eq. 3 without external regression process. Third, CSN is proposed based on traditional facial descriptors, but our DAM is based on the SOTA framework using deep neural network architectures and effective loss functions, where the extracted features are distributed on the hyper-sphere. Moreover, we also propose a learning based method to estimate the LID without searching neighboring samples.

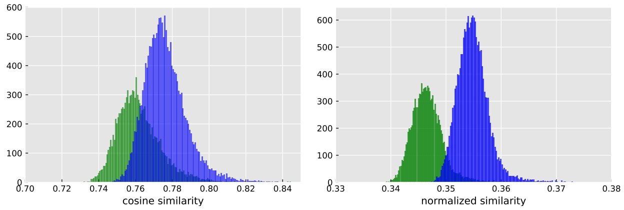


Figure 6: Comparison between the cosine similarity and the normalized similarity score of DAM. The green and blue curves represent similarity distributions correspond to the two disjoint segments of probability, as described in Fig. 5. The right illustrates the original cosine similarity, and the left represents the similarity score of DAM.

Discussion on the combination format of DAM. We replace the summation operation with the multiplication operation in Eq. 3. We adopt the same setting of Table 1 to compare the summation and multiplication operations under the reference-based DAM, which are called as DAM-R-S and DAM-R-M, respectively. The TAR results of DAM-R-M are 94.56% and 90.80% on the IJB-B dataset under the FAR of 1e-4 and 1e-5, respectively, which are comparable with the results (94.63%, 90.83%) of DAM-R-S. It indicates that the choices of summation and multiplication operations do not bring explicit differences of our proposed DAM.

5. Conclusion

In this paper, we have investigated the gap between the training and verification process and the effectiveness of local inter-class discrepancy information for face recognition. Then, we have proposed a novel verification metric called DAM for face recognition. Extensive experiments among different face recognition benchmarks demonstrate the effectiveness of our proposed DAM.

References

[1] Dong Cao, Xiangyu Zhu, Xingyu Huang, Jianzhu Guo, and Zhen Lei. Domain balancing: Face recognition on long-tailed domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5710–5719, 2020.

[3] Dong Chen, Xudong Cao, Liwei Wang, Fang Wen, and Jian Sun. Bayesian face revisited: A joint formulation. In *European conference on computer vision*, pages 566–579. Springer, 2012.

[4] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[5] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *Proceedings of the IEEE Conference on European Conference on Computer Vision*, 2020.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[7] Jiankang Deng, Jia Guo, Jing Yang, Alexandros Lattas, and Stefanos Zafeiriou. Variational prototype learning for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11906–11915, June 2021.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[12] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[13] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[14] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[16] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[17] Yonghyun Kim, Wonpyo Park, and Jongju Shin. Broadface: Looking at tens of thousands of people at once for face recognition. *arXiv preprint arXiv:2008.06674*, 2020.

[18] Jiaheng Liu, Shunfeng Zhou, Yichao Wu, Ken Chen, Wanli Ouyang, and Dong Xu. Block proposal neural architecture search. *IEEE Transactions on Image Processing*, 30:15–25, 2020.

[19] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[20] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.

[21] Yury A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[22] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165. IEEE, 2018.

[23] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[24] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016.

[25] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[26] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[28] Gregory Shakhnarovich, John W Fisher, and Trevor Darrell. Face recognition from long-term observations. In *European Conference on Computer Vision*, pages 851–865. Springer, 2002.

[29] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6902–6911, 2019.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[32] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020.

[33] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[34] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[35] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900, 2015.

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[37] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[38] Massimo Tistarelli et al. On the use of discriminative cohort score normalization for unconstrained face recognition. *IEEE Transactions on information forensics and security*, 9(12):2063–2075, 2014.

[39] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[40] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017.

[41] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[42] M Wang and W Deng. Deep face recognition: A survey. arxiv 2018. *arXiv preprint arXiv:1804.06655*.

[43] Mei Wang and Weihong Deng. Mitigating bias in face recognition using skewness-aware reinforcement learning. In *CVPR*, pages 9322–9331, 2020.

[44] Mei Wang, Weihong Deng, Jiani Hu, Xunqiang Tao, and Yaohai Huang. Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *ICCV*, pages 692–702, 2019.

[45] Xiaobo Wang, Shuo Wang, Shifeng Zhang, Tianyu Fu, Hailin Shi, and Tao Mei. Support vector guided softmax loss for face recognition. *arXiv preprint arXiv:1812.11317*, 2018.

[46] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[47] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–98, 2017.

[48] Lior Wolf, Tal Hassner, and Itay Maoz. *Face recognition in unconstrained videos with matched background similarity*. IEEE, 2011.

[49] Yudong Wu, Yichao Wu, Ruihao Gong, Yuanhao Lv, Ken Chen, Ding Liang, Xiaolin Hu, Xianglong Liu, and Junjie Yan. Rotation consistent margin loss for efficient low-bit face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[50] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[51] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017.

[52] Xingcheng Zhang, Lei Yang, Junjie Yan, and Dahua Lin. Accelerated training for massive classification via dynamic class selection. *arXiv preprint arXiv:1801.01687*, 2018.

[53] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019.

[54] Kai Zhao, Jingyi Xu, and Ming-Ming Cheng. Regularface: Deep face recognition via exclusive regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1136–1144, 2019.

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

[56] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *ICLR*, 2017.