IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received May 24, 2020, accepted June 10, 2020, date of publication June 15, 2020, date of current version June 24, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3002270*

# LinCos-Softmax: Learning Angle-Discriminative Face Representations With Linearity-Enhanced Cosine Logits

**WEI-FENG OU[1], LAI-MAN PO[1], (Senior Member, IEEE), CHANG ZHOU[1], (Graduate Student Member, IEEE), YU-JIA ZHANG[1], LI-TONG FENG[2], YASAR ABBAS UR REHMAN[3], (Member, IEEE), AND YU-ZHI ZHAO[1], (Graduate Student Member, IEEE)**

[1]Department of Electrical Engineering, City University of Hong Kong, Hong Kong
[2]SenseTime, Hong Kong
[3]TCL Corporate Research Corporation, Ltd., Hong Kong

Corresponding author: Wei-Feng Ou (weifengou2-c@my.cityu.edu.hk)

**ABSTRACT** In recent years, the angle-based softmax losses have significantly improved the performance of face recognition whereas these loss functions are all based on cosine logit. A potential weakness is that the nonlinearity of the cosine function may undesirably saturate the angular optimization between the features and the corresponding weight vectors, thereby preventing the network from fully learning to maximize the angular discriminability of features. As a result, the generalization of learned features may be compromised. To tackle this issue, we propose a Linear-Cosine Softmax Loss (LinCos-Softmax) to more effectively learn angle-discriminative facial features. The main characteristic of the loss function we propose is the use of an approximated linear logit. Compared with the conventional cosine logit, it has a stronger linear relationship with the angle on enhancing angular discrimination through Taylor expansion. We also propose an automatic scale parameter selection scheme, which can conveniently provide an appropriate scale for different logits without the need for exhaustive parameter search to improve performance. In addition, we propose a margin-enhanced Linear-Cosine Softmax Loss (m-LinCos-Softmax) to further enlarge inter-class distances and reduce intra-class variations. Experimental results on several face recognition benchmarks (LFW, AgeDB-30, CFP-FP, MegaFace Challenge 1) demonstrate the effectiveness of the proposed method and its superiority to existing angular softmax loss variants.

**INDEX TERMS** Face recognition, loss function, feature representations, cosine logits, softmax.

## I. INTRODUCTION

In recent year, due to the advances of deep convolutional neural networks (CNNs) [1]–[3], the availability of large-scale face training data [4], [5] and sophisticated loss function designs [13], [20]–[22], [30], face recognition has achieved significant progress. Currently, face recognition mainly involves two types of applications, namely face identification and face verification. Face identification aims to recognize the identity of a target face from a set of registered faces, while face verification aims to verify whether two faces belong to the same identity. Basically, these identification and verification processes are based on the matching of facial features extracted by a CNN network. Therefore, it is critical that the trained CNN network must be able to extract discriminative facial features to achieve outstanding recognition performance. To achieve this, in recent years, a large amount of literature has focused on the design of various loss functions to enforce strong intra-class compactness and large inter-class differences in the feature space to enhance generalization. Existing loss functions for deep representation learning roughly fall into two categories: metric learning-based methods and classification-based methods.

The methods based on metric learning aim to learn a feature representation that maps similar faces as close as possible and maps dissimilar faces as apart as possible. They

The associate editor coordinating the review of this manuscript and approving it for publication was Junhua Li.

usually achieve this by imposing some distance constraints to a pair or a triplet of samples in the feature space to push similar samples together and pull dissimilar samples apart. For example, contrastive loss [11], [12] constructs training sample pairs to minimize the distance of intra-class pairs while maximizing the distance of inter-class pairs. In contrast, the triplet loss [13] employs triplet training samples to train the network. A triplet consists of an anchor sample, a positive sample and a negative sample. The network is trained to ensure that the anchor-negative distance is larger than the anchor-positive distance by a given margin. Recently, Sohn proposed an N-pair loss [43], which generalizes the triplet loss to allow joint comparisons between multiple negative examples. Chen *et al.* further improved the generalization capability of features by proposing a quadruplet network [44] to apply more complex constraints on the features. In addition, Song *et al.* proposed a lifted structure loss [45] for deep metric learning by lifting the vector of pairwise distance within a batch to the matrix of pairwise distances, achieving improved feature representation. Besides, Wu *et al.* [15] proposed a distance-weighted sampling method with a simple margin-based loss by selecting more informative and stable examples, thereby achieving excellent performance. One disadvantage of the metric learning-based approach is that they require to construct tuples of training samples, and the number of tuples grows rapidly with the size of the training data, a large proportion of which is trivial, which leads to slow and unstable convergence. Although using some sample mining techniques [13]–[15] can relieve this problem, they actually introduce additional complexity and make the training process more difficult.

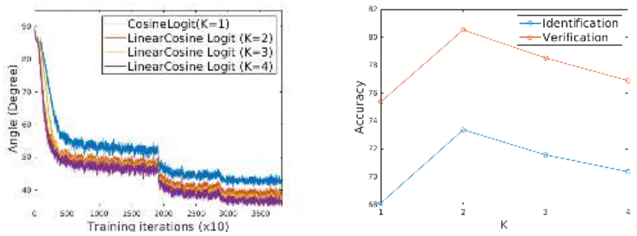
On the other hand, the classification-based methods utilize an additional classifier after the embedding layer to train the network as a classification problem, and only retains the embedding network as a feature extractor during the testing phase. It has been shown that training a classification task with a large number of face identities is helpful to learn robust features [16]. Our work is also based on the classification-based method. The most straightforward way is to use softmax loss for classification training, such as DeepID [16], DeepFace [17] and VggFace [18]. These pioneering works explored the feasibility of using CNN for face feature extraction and achieved promising performance. However, the softmax loss only learns a feature space with overlapped decision boundaries between different classes, so it is not reliable when generalizing it to unseen samples with large variations. In order to solve this problem, various literatures have tried to reformulate the softmax loss by imposing margin constraints to increase inter-class distances and reduce intra-class variations. For example, SphereFace [20] pioneered the concept of angular margin to enhance the angular discrimination of face features by a multiplicative angular margin. This method achieved a significant performance improvement compared to the softmax loss. Recently, CosFace [21] and AM-Softmax [23] proposed the use of additive cosine margin to enhance angular discrimination, thereby further improving the performance of face recognition. Moreover, ArcFace [22] proposed an additive angular margin and incorporated different types of margin into a unified framework to obtain face features with high discriminative ability. Recently, Chen *et al.* proposed a virtual softmax [46] by injecting a dynamic virtual negative class into original softmax loss to enlarge inter-class margin and compress intra-class distribution.

In addition, some literatures have found the benefits of normalizing the features and weights of the classifier in improving feature discriminative power [24]–[27]. The normalization eliminates the influence from the length of the features and weights during training, thus the network can focus on optimizing the cosine similarity between features and class vectors to boost angular discrimination. For instance, Ranjan *et al.* introduced a L2-constrained softmax [26] loss to restrict the features on a hypersphere with a fixed radius, which can significantly improve the performance of face verification. Similarly, NormFace [24] normalized both feature vectors and weight vectors to optimize cosine similarity instead of the inner products in the softmax loss, thereby effectively improving the angular discrimination of the features. Besides, the adaptive selection of the scale and margin hyper-parameters were studied in [28], [29], [47]. Zhang *et al.* studied the settings of scale and angular margin parameter in cosine-based softmax losses and proposed AdaCos [29] to adaptively scale cosine logits to enhance the supervision during training. Liu *et al.* proposed an adaptive margin softmax [28] to adaptively adjust the margins for different classes to tackle the problem of imbalanced training data in face recognition. Recently, Wang *et al.* proposed a mis-classified vector guided softmax loss [47] for face recognition to guide the discriminative feature learning by assigning adaptive feature margins for different classes.

Compared with the tight coupling of angular constraints and softmax loss, some literatures tried to design separable regularization terms to achieve joint supervision, so that multiple regularization effects can be applied to features for improving generalization. For example, Center Loss [30] uses the L2 distance between features and class centers as an auxiliary supervisory signal to reduce the intra-class variation of the features, and the softmax loss is responsible for enlarging the inter-class dispersion. Recently, He *et al.* proposed a D-Softmax [48] by dissecting the softmax loss into independent intra- and inter-class objectives and tuned each part to the best state, which significantly accelerated the training process with only a minor sacrifice in performance. Zhang *et al.* designed a Range Loss [37] joint with softmax loss to overcome the long tail effect of real-world data for face recognition. Similar multi-task losses included Marginal loss [34], Ring loss [38], Center invariant loss [36], Git loss [35], Feature contraction loss [40], RegularFace [32], UniformFace [33] as well as Gaussian mixture loss [39], etc.

Despite the excellent performance achieved by the angle-based softmax loss variants in recent years, one potential weakness is that the angle is nonlinearly mapped by a cosine function. The nonlinearity of the cosine function may lead to insufficient angular optimization between features and corresponding class weights. As a result, the angular discriminability of the features may be compromised, resulting in a reduced generalization ability.

To tackle this issue, we propose a Linear-Cosine Softmax Loss to learn angle-discriminative face features more effectively. The main novelty of the proposed loss function is the use of a linear-cosine logit, which is designed by performing Taylor expansion on a linear logit. The designed linear-cosine logit has a stronger linear relationship with the angle than the conventional cosine logit, so it helps to enhance the angular discriminability of the learned features and improve the generalization ability. To achieve a better comparison and analysis of different logits, we then propose an automatic scale hyper-parameter selection scheme, which can automatically determine the appropriate scaling parameters for different logits. Under this scheme, different logits are scaled properly within the same range, thereby helps to analyze how the different logit curvatures affect the angular discrimination and improve performance. As shown in Fig. 1, the proposed linear-cosine logit achieves a smaller intra-class angle during the training compared to the conventional cosine logit, which promotes stronger angular discriminability of features and leads to a better generalization performance on the testing set. In addition, we designed a margin-enhanced Linear-Cosine Softmax Loss by applying different types of angular margins to the proposed Linear-Cosine Softmax loss to further enhance the intra-class compactness and inter-class separability of features. Experimental results on several well-known face recognition benchmarks showed that the proposed Linear-Cosine Softmax Loss effectively boosts the performance of face recognition and outperformed some well-known angle-based softmax loss variants.



(a) (b)

**FIGURE 1. (a) Changes of intra-class angle between features and corresponding class vectors during training using different logits which are scaled to the same range by the proposed automatic scheme. K is the order of Taylor expansion. (b) Performance (%) comparison of different logits on MegaFace challenge 1.**

Our major contributions can be summarized as follows:

1) We proposed a novel LinCos-Softmax Loss, which utilizes a linear-cosine logit that has stronger linear relationship with the angle by performing a Taylor expansion to the linear logit to enhance the angular discriminability and generalization ability of the learned features. We further impose different margins to the LinCos-Softmax Loss to enhance the intra-class compactness and inter-class separability of the learned features.
2) We propose an automatic scaling hyper-parameter selection scheme to automatically determine the proper scaling parameters for different logits. Under this scheme, different logits are scaled to the same range, which helps to study how different logits affect the angular discrimination and improve the performance.
3) We perform comprehensive experiments on some well-known face recognition benchmarks (LFW [6], CFP-FP [7], AgeDB-30 [8], MegaFace challenge one [9]) by training on small training set (CASIA [4]) and large training set (MS1M [5]) respectively to validate the effectiveness of the proposed methods.

## II. METHODOLOGY

One potential disadvantage of the conventional angle-based softmax losses is that the angle is mapped nonlinearly by the cosine function. The nonlinearity of the cosine function may lead to insufficient angular optimization, which will be analyzed first in this section and then how the proposed Linear-Cosine Softmax loss solves this problem.

### A. COSINE LOGIT SOFTMAX LOSS

Generally, the cosine logit based softmax loss (Cos-Softmax) can be defined as

$$L_i^{cos} = -\log\left(P_{y_i}^{cos}\right) \tag{1}$$

where

$$P_{y_i}^{cos} = \frac{e^{sf_{y_i}^{cos}}}{\sum_{j=1}^{C} e^{sf_j^{cos}}} \tag{2}$$

represents the posterior probability of the $i^{th}$ sample belonging to class label $y_i$. $C$ denotes the total number of classes. $s$ is a scale hyper-parameter. The cosine logit $f_j^{cos}$ is defined as

$$f_j^{cos} = W_j^T x_i = \cos\theta_j \tag{3}$$

where we have omitted the bias term for simplicity following [20], [22]. $x_i \in R^D$ represents the L2-normalized features of the $i^{th}$ sample. $W_j$ denotes the $j^{th}$ column of the L2-normalized weights $W \in R^{D\times C}$. $\theta_j$ denotes the angle between $x_i$ and $W_j$. $D$ denotes the feature dimension.

From an angle perspective, the cosine logit is indirectly modulated by the angle through the cosine function. The nonlinearity of the cosine function may oversaturate the angle, which may limit the network from learning, and thus cannot sufficiently reduce the angle between the feature and the corresponding weight vectors. As a result, the angular discriminability and generalization power of features may be compromised. To demonstrate this phenomenon, we performed a gradient analysis on the partial derivatives of $L_i^{cos}$ with respect to $\cos\theta_{y_i}$ and $\theta_{y_i}$ for determining the cause of the problem. We only analyze the case $j = y_i$ for convenience of explanation, because it plays a major role during training compared to $j \neq y_i$, which is explained in Fig. 2. In the

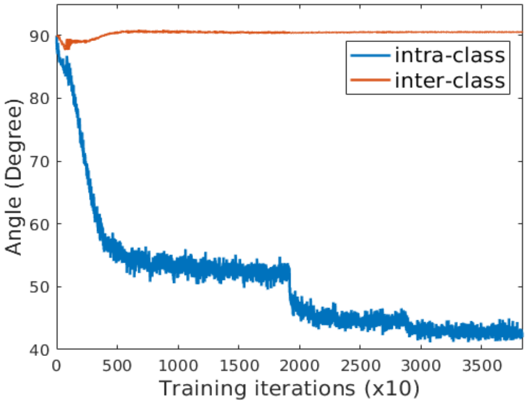


**FIGURE 2.** Changes of intra-class angle $\theta_{y_i}$ and inter-class angle $\theta_j$ ($j \neq y_i$) between feature and weight vectors when training on CASIA. The intra-class angle is gradually decreasing while the inter-class angles almost stay around 90 degree during the whole training process.

rest of the paper, we may use the terms gradient, derivative and partial derivative interchangeably for convenience. The partial derivative of cosine logit $L_i^{cos}$ with respect to $\cos\theta_{y_i}$ is given by

$$\frac{\partial L_i^{cos}}{\partial \cos\theta_{y_i}} = s\left(P_{y_i}^{cos} - 1\right) \tag{4}$$

This indicates that as $P_{y_i}^{cos}$ gradually approaches one, the gradient of $L_i^{cos}$ w.r.t. $\cos\theta_{y_i}$ will vanish accordingly. Basically, this is a good attribute, because if the scaling factor $s$ is properly selected, it means that as $\theta_{y_i}$ gradually decreases to zero, the network will converge spontaneously and stop learning. However, the derivative of cosine logit $L_i^{cos}$ w.r.t. $\theta_{y_i}$ is given by

$$\frac{\partial L_i^{cos}}{\partial \theta_{y_i}} = -s\left(P_{y_i}^{cos} - 1\right)\sin\theta_{y_i} \tag{5}$$

This equation indicates that this gradient w.r.t. $\theta_{y_i}$ is weakened by the term $\sin\theta_{y_i}$, which is resulted from the derivative of the cosine function. Unfortunately, this is undesirable because the factor $P_{y_i}^{cos} - 1$ already well guarantees the vanishing of the gradient as $\theta_{y_i}$ approaches zero. The additional decaying factor may cause excessive gradient reduction and insufficient optimization of the intra-class angle, thereby compromising the angular discriminability of features accordingly.

### B. LINEAR LOGIT SOFTMAX LOSS

The cosine logit is indirectly modulated by the angle through the cosine function, which can cause a harmful factor from the perspective of angle $\theta_{y_i}$. Based on this observation, it seems reasonable to use a linear logit which is directly modulated by the angle so that the network can learn to reduce the intra-class angle more effectively. Specifically, the linear logit is defined as

$$f_j^{linear} = -\theta_j + \frac{\pi}{2} = -\arccos\left(\cos\theta_j\right) + \frac{\pi}{2} \tag{6}$$

where the angle is obtained by taking the arccosine because it is not directly available. Thus, the Linear-logit Softmax Loss (Lin-Softmax) is defined as

$$L_i^{linear} = -\log\left(P_{y_i}^{linear}\right) \tag{7}$$

where

$$P_{y_i}^{linear} = \frac{e^{sf_{y_i}^{linear}}}{\sum_{j=1}^{C} e^{sf_j^{linear}}} \tag{8}$$

The gradient of $L_i^{linear}$ with respect to $\theta_{y_i}$ is given by

$$\frac{\partial L_i^{linear}}{\partial \theta_{y_i}} = -s\left(P_{y_i}^{linear} - 1\right) \tag{9}$$

where the undesirable factor $\sin\theta_{y_i}$ does not exist. However, the gradient with respect to $\cos\theta_{y_i}$ is

$$\frac{\partial L_i^{linear}}{\partial \cos\theta_{y_i}} = s\left(P_{y_i}^{linear} - 1\right)\frac{1}{\sqrt{1-(\cos\theta_{y_i})^2}} \tag{10}$$

This equation shows that as $\theta_{y_i}$ gradually decreases, the gradient is inversely amplified by the factor $1/\sqrt{1-(\cos\theta_{y_i})^2}$. In extreme cases, when $\theta_{y_i}$ equals zero, the gradient will tend to infinity. This is harmful because it makes the training more difficult to converge and increases the risks of gradient explosion.

### C. LINEAR-COSINE LOGIT SOFTMAX LOSS

In order to alleviate the gradient decaying problem of cosine logit and the gradient amplification problem of linear logit, we proposed a new logit called linear-cosine logit, which is a tradeoff between cosine logit and linear logit. Specifically, we represent $\theta_j$ as the arccosine of $\cos\theta_j$, then perform a Taylor expansion over the arccosine function, and approximate the angle using the first $K$ terms:

$$\theta_j = \arccos\left(\cos\theta_j\right) \approx \hat{\theta}_j \tag{11}$$

where

$$\hat{\theta}_j = \frac{\pi}{2} - \sum_{n=0}^{K-1} c_n(\cos\theta_j)^{2n+1} \tag{12}$$

and

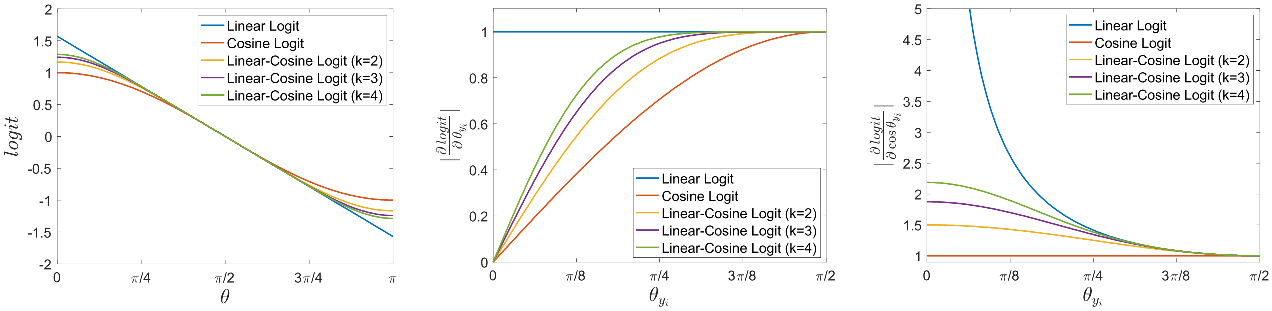
$$c_n = \frac{(2n)!}{2^{2n}(n!)^2(2n+1)} \tag{13}$$

$\hat{\theta}_j$ is the approximated angle using $K$ Taylor series terms and $c_n$ are the coefficients of the Taylor series. By substituting the original $\theta_j$ with the approximated angle $\hat{\theta}_j$ into (6), the linear-cosine logit is defined as

$$f_j^{LinCos} = -\hat{\theta}_j + \frac{\pi}{2} = \sum_{n=0}^{K-1} c_n(\cos\theta_j)^{2n+1} \tag{14}$$

Thus, the Linear-Cosine Softmax Loss (LinCos-Softmax) for the $i^{th}$ sample is defined as

$$L_i^{LinCos} = -\log\left(P_{y_i}^{LinCos}\right) \tag{15}$$



(a) Different logits

(b) Derivative magnitude of different logits w.r.t to $\theta_{y_i}$.

(c) Derivative magnitude of different logits w.r.t to $\cos\theta_{y_i}$.

**FIGURE 3. Comparison of linear logit, cosine logit and linear-cosine logit. In (b) and (c), we only plot the range between [0, $\pi/2$] for better viewing because the angle usually stays within this range during training (as shown in Fig. 2).**

where

$$P_{y_i}^{LinCos} = \frac{e^{sf_{y_i}^{LinCos}}}{\sum_{j=1}^{C} e^{sf_j^{LinCos}}} \tag{16}$$

Its gradients with respect to $\theta_{y_i}$ and $\cos\theta_{y_i}$ are given by

$$\frac{\partial L_i^{LinCos}}{\partial \theta_{y_i}} = -s\left(P_{y_i}^{LinCos} - 1\right)\sum_{n=0}^{K-1}\frac{(2n)!}{2^{2n}(n!)^2}(\cos\theta_{y_i})^{2n}\sin\theta_{y_i} \tag{17}$$

$$\frac{\partial L_i^{LinCos}}{\partial \cos\theta_{y_i}} = s\left(P_{y_i}^{LinCos} - 1\right)\sum_{n=0}^{K-1}\frac{(2n)!}{2^{2n}(n!)^2}(\cos\theta_{y_i})^{2n} \tag{18}$$

Now, we have formulated three loss functions, Cos-Softmax, Lin-Softmax, and LinCos-Softmax, as well as their partial derivatives w.r.t $\cos\theta_{y_i}$ and $\theta_{y_i}$. To explain why the linear-cosine logit can achieve a good trade-off between linear logit and cosine logit, we illustrated the three logits in Fig. 3(a) and their magnitudes of derivatives w.r.t $\cos\theta_{y_i}$ and $\theta_{y_i}$ in Fig. 3(b) and Fig. 3(c), respectively.

From Fig. 3(a), we can observe that linear-cosine logit using different $K$ terms all live between the linear logit and cosine logit as a trade-off between them. In Fig. 3(b), the derivative magnitude of cosine logit w.r.t $\theta_{y_i}$ decays more rapidly than the other logits as the angle decreases from $\pi/2$ to 0. This could prevent the intra-class angle $\theta_{y_i}$ from being optimized sufficiently and reduce generalization power. Comparatively, the linear-cosine logit decays slower than the cosine logit as $\theta_{y_i}$ decreases, thus can optimize $\theta_{y_i}$ more sufficiently to achieve more compact intra-class distribution. Moreover, Fig. 3(c) shows that the derivative magnitude of linear logit w.r.t $\cos\theta_{y_i}$ increases rapidly as $\theta_{y_i}$ decreases from $\pi/2$ to 0. In extreme cases, it will tend to infinity when $\theta_{y_i}$ becomes zero. This could increase the risks of gradient explosion and make the training unstable. Comparatively, the linear-cosine logit grows much slower than the linear logit and is always bounded as $\theta_{y_i}$ decreases, thus can avoid the gradient explosion problem. Hence, we can see that the linear-cosine logit can reach a good trade-off between linear logit and cosine logit by overcoming both of their disadvantages. In fact, the cosine logit and linear logit can be considered as special cases of the linear-cosine logit, corresponding to $K = 1$ and $K = \infty$ respectively.

### D. AUTOMATIC SCALE PARAMETER SELECTION

In previous sections, we discussed different softmax loss variants from a gradient perspective, but did not discuss how to choose the scaling hyper-parameter $s$ involved in these loss functions. The scaling hyper-parameter has a significant impact on the gradient of the loss with respect to network parameters, thereby greatly affecting the network optimization process and the final recognition performance. Therefore, it is important to choose $s$ properly. Furthermore, as shown in Fig. 3(a), different logit functions have different output ranges, and it is more reasonable to compare them in the same range by considering the scaling of $s$ together. This helps to better analyze the influence of curvature of different logits. The choice of the scaling hyper-parameter $s$ usually relies on heuristic trials, which are both time consuming and inconvenient to use. The automatic selection of $s$ has been discussed in [29]. Inspired by these efforts, we designed a simple scheme to automatically determine this scale parameter for different logits, so that their scale ranges are the same. In general, we have

$$P_{y_i} = \frac{e^{sf_{y_i}}}{\sum_{j=1}^{C} e^{sf_j}} \tag{19}$$

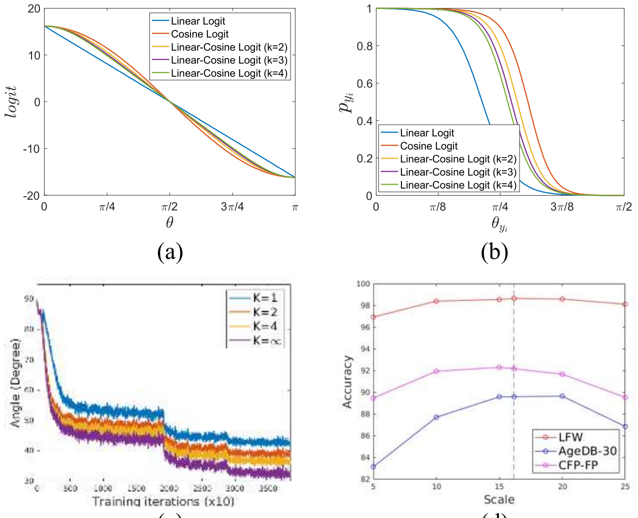
In this probability expression, we have omitted the name of the logit in the super-script of $P_{y_i}$ and $f_{y_i}$ because the following discussion applies to all the logits introduced in previous sections. Based on the findings that $\theta_j \approx \pi/2$ for $j \neq y_i$ during the whole training (see Fig. 2) and $f_j\big|_{\theta_j=\frac{\pi}{2}} = 0$, we can simplify $P_{y_i}$ as

$$P_{y_i} = \frac{e^{sf_{y_i}}}{e^{sf_{y_i}} + C - 1} \tag{20}$$

We want to find a suitable $s$ such that $P_{y_i}$ properly spans over the whole range of [0, 1] as $\theta_{y_i}$ decreases from $\pi/2$

**TABLE 1. Scale parameter for different logits determined by the proposed scheme.**

| Logit | Cos<br>$(K = 1)$ | LinCos<br>$(K = 2)$ | LinCos<br>$(K = 3)$ | LinCos<br>$(K = 4)$ | Lin<br>$(K = \infty)$ |
|---|---|---|---|---|---|
| $s$ | 16.17 | 13.86 | 13.03 | 12.57 | 10.29 |



(a) (b)

(c) (d)

**FIGURE 4. (a) Scaled logits by our automatic scheme; (b) The corresponding $P_{y_i}$ for the scaled logits; (c) Changes of intra-class angle $\theta_{y_i}$ during training for different $K$; (d) Performance comparison of using different scales for cosine logit ($K = 1$), the dash line corresponds to the scale determined by our proposed scheme, and the other scales are empirically traversed with $s = 5, 10, 15, 20, 25$. Experiments are conducted on CASIA with $C = 10575$ classes.**

to zero during training. Since $P_{y_i}|_{\theta_{y_i}=\frac{\pi}{2}} \approx \frac{1}{C} \approx 0$ (the number of training classes $C$ is usually very large in deep face recognition), we can simply require $P_{y_i}$ to be close to one, when $\theta_{y_i} = 0$. Specifically, we can set up the following equation:

$$P_{y_i}|_{\theta_{y_i}=0} = \eta \tag{21}$$

where $\eta$ is another hyper-parameter close to one and we use $\eta = 0.999$ in all our experiments. By solving (21), $s$ can be automatically determined as

$$s = \left( \log \frac{\eta}{1-\eta}(C-1) \right) / f_{y_i}|_{\theta_{y_i}=0} \tag{22}$$

By using this simple equation, we can automatically determine an appropriate scale $s$ for different logits. The scales for different logits are shown in Table 1, when trained on CASIA with $C = 10575$ classes, and the scaled logits and corresponding probabilities $P_{y_i}$ are illustrated in Fig. 4(a) and Fig. 4(b), respectively. We can see from Fig. 4(a) that different logits are scaled in the same range. As $K$ increases, Fig. 4(b) shows that the proposed linear-cosine logit will apply more penalty for the same angle $\theta_{y_i}$ to boost angular discrimination. This can be further validated from Fig. 4(c), which shows that a larger $K$ achieves a smaller intra-class angle during training. The choice of $K$ involves balancing a suitable strength of penalty between enhancing angular discrimination on train data and maximizing feature generalization. In our experiments, we found $K = 2$ worked the best. In addition, we also compared the performance of using different scales for cosine logit ($K = 1$) in Fig. 4(d). We can find that the scale determined by our automatic scheme (dashed line) has higher performance than the other empirically traversed scales in all the validation sets, demonstrating the effectiveness of our proposed scheme. This is mainly because our automatic selection method can select the appropriate scaling hyper-parameter during the network optimization process to maintain the appropriate gradient strength, thereby improving the generalization performance.

### *E. MARGIN ENHANCEMENT*

Recently, it has been shown that incorporating margins with softmax loss can lead to significantly improved features [20]–[23]. These methods can be applied to the proposed LinCos-Softmax loss for further improvement. We apply three types of margin to the LinCos-Softmax loss, and we denote the margin-enhanced loss as m-LinCos-Softmax.

Specifically, to impose the multiplicative angular margin $m_1$ and additive angular margin $m_2$, we simply substitute $\hat{\theta}_{y_i}$ with $m_1\hat{\theta}_{y_i}+m_2$ into Eq. (14) to enforce extra angular penalty. Then, we further impose the additive cosine margin by substituting $\cos\theta_{y_i}$ with $\cos\theta_{y_i} - m_3$ into Eq. (14) to enforce extra cosine penalty. Note that the margins are only applied to the target class logit, i.e., $j = y_i$. Based on (12) and (14), the margin-enhanced linear-cosine logit for the target class is defined as

$$\begin{aligned} f_{y_i}^{m-LinCos} &= f_{y_i}^{LinCos}\Big|_{\hat{\theta}_{y_i} \leftarrow m_1\hat{\theta}_{y_i}+m_2, \cos\theta_{y_i} \leftarrow \cos\theta_{y_i}-m_3} \\ &= m_1 \sum_{n=0}^{K-1} c_n(\cos\theta_{y_i}-m_3)^{2n+1} - \frac{\pi}{2}(m_1-1)-m_2 \end{aligned} \tag{23}$$

while for $j \neq y_i$, $f_j^{m-LinCos} = f_j^{LinCos}$. Finally, the margin-enhanced linear-cosine softmax loss is defined as

$$L_i^{m-LinLos} = -\log \left( \frac{e^{sf_{y_i}^{m-LinCos}}}{\sum_{j=1}^{C} e^{sf_j^{m-LinCos}}} \right) \tag{24}$$

Fig. 5 illustrates the margin-enhanced logits under different margin settings using $K = 2$ as an example, where the blue curve is the baseline logit without margin enhancement. We can see that the margins are essentially pulling down the baseline logit curve in certain ways to impose extra penalty, thereby enforcing angular discrimination of features. When $m_1 = 1$, $m_2 = 0$, $m_3 = 0$, the m-LinCos-Softmax loss reduces to LinCos-Softmax loss.

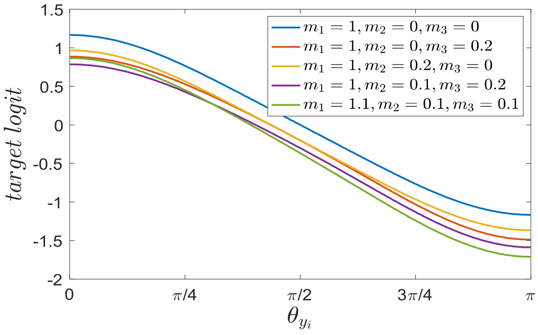


**FIGURE 5.** Comparison of target logit curves under different margin settings using $K = 2$.

**TABLE 2.** Verification accuracies (%) of different methods on LFW, CFP-FP and AgeDB-30 when training CASIA.

| Method | LFW | AgeDB-30 | CFP-FP |
|---|---|---|---|
| Lin-Softmax | 98.60 | 89.33 | 92.66 |
| Cos-Softmax | 98.67 | 89.63 | 92.19 |
| LinCos-Softmax ($K = 2$) | **98.82** | **91.05** | **93.11** |
| LinCos-Softmax ($K = 3$) | 98.53 | 90.38 | 93.17 |
| LinCos-Softmax ($K = 4$) | 98.67 | 90.07 | 93.17 |
| m-LinCos-Softmax ($m_1$=1,$m_2$=0,$m_3$=0.2, $K = 1$) | 99.17 | 91.98 | 93.70 |
| m-LinCos-Softmax ($m_1$=1,$m_2$=0,$m_3$=0.2, $K = 2$) | **99.20** | **92.30** | **93.81** |

## III. EXPERIMENTS

### A. EXPERIMENTAL SETTINGS

#### 1) TRAINING DATA

We used two public training datasets, CASIA [4] and MS1M [5], for training our models with different loss functions. CASIA contains around 500000 images from 10575 identities. We used a cleaned subset of MS1M by removing the label noises, leaving 3.8M images from 85K identities. All the face images were preprocessed by MTCNN [10] for face detection and aligned by a similarity transformation to adjust to a size of 112 × 112 before feeding them to the network. Each pixel was then subtracted by 127.5 and divided by 128 for normalization. Random horizontal flipping was used for data augmentation.

#### 2) NETWORK SETTINGS

Following the settings of SphereFace [20], we adopted a 20-layer ResNet-like model as our network architecture with output feature dimension 512. This network achieves a good trade-off between performance and model complexity. We used SGD for network training with momentum 0.9 and weight decay factor 0.0005. The initial learning rate was 0.1, and divided by 0.1 at 10th, 15th and 18th epoch. The training was finished in 20 epochs. We used PyTorch for implementation.

#### 3) EVALUATION

We evaluate the model performance using three validation sets, LFW [6], CFP-FP [7], AgeDB-30 [8] and a well-known testing benchmark, MegaFace challenge 1 [9]. LFW is a widely used face verification benchmark, containing 13222 images of 5749 identities with large variations in illumination, expression, pose, etc. Since many of today's deep learning-based face recognition models can easily achieve beyond 98% verification accuracy on LFW, we used more challenging verification benchmarks, CFP-FP and AgeDB-30, for a better performance evaluation. The CFP dataset is for pose-invariant face verification with 7000 images of 500 identities, while the AgeDB dataset is for age-invariant face recognition with 16488 images of 568 identities. Following the *unrestricted with labeled outside data protocol*, we evaluate verification accuracy with 10-fold cross validation on 6000 face pairs of LFW, 7000 frontal-profile face pairs of CFP, and 6000 face pairs of AgeDB under the age difference 30. Half of the face pairs are positive pairs from the same identity, while the other half are negative pairs from different identities. MegaFace is a challenging testing benchmark that can be used to evaluate face recognition models at million-scale distractors. MegaFace challenge 1 (MF1) includes a probe set from FaceScrub with 100K images of 530 identities and a gallery set containing more than 1 million images of 690K identities. We evaluate Rank-1 identification accuracy and verification True Positive Rate (TPR) at $10^{-6}$ False Positive Rate (FPR) with one million distractors. The cosine similarity is used for measuring the similarity between feature vectors extracted from face images.

### B. LOGIT FUNCTION EVALUATION

In this section, we conducted an ablation study on different logits. Specifically, we studied the effectiveness of the proposed linear cosine logit with and without margin enhancement respectively. Under non-margin setting, the scale parameter was determined by our proposed automatic scheme for different logits. Under margin setting, we empirically used a fixed scale parameter $s = 20$ for different logits for simplicity instead of using the automatic scaling scheme. This is because using various margin settings will significantly change the logit curves, making the automatic and joint determination of both scale and margin parameters more complicated, which is out of the scope of this work. The results for training on CAISA are summarized Table 2 and Table 3, the results for training on MS1M are summarized in Table 4, and the CMC/ROC curves of different methods are illustrated in Fig. 6.

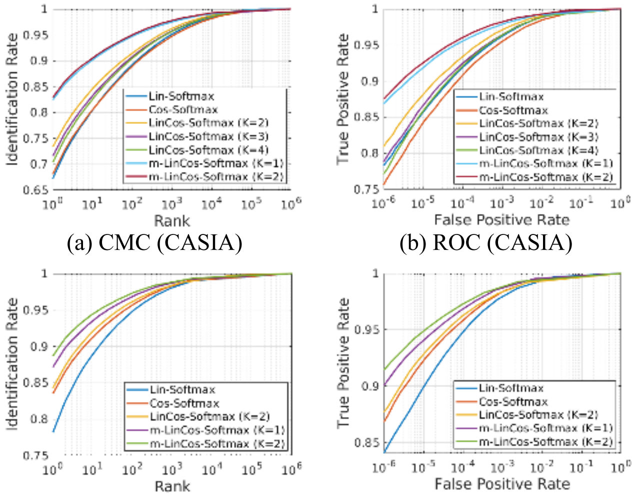
From Table 2 and Table 3, we can clearly see that under non-margin setting (row 1 ∼ row 5), the proposed LinCos-Softmax outperformed both Lin-Softmax and Cos-Softmax significantly in all the evaluation sets. The performance of LinCos-Softmax decreased gradually as $K$ increased from 2 to 4. The best performance is obtained for $K = 2$, with an improvement of around 5% compared to Cos-Softmax and an improvement of around 6% compared to Lin-Softmax in

**TABLE 3. Identification and verification performance (%) on MegaFace challenge 1 when training on CASIA.**

| Method | Ident. | Veri. |
|---|---|---|
| Lin-Softmax | 67.27 | 78.71 |
| Cos-Softmax | 68.12 | 75.38 |
| LinCos-Softmax ($K = 2$) | **73.38** | **80.53** |
| LinCos-Softmax ($K = 3$) | 71.58 | 78.52 |
| LinCos-Softmax ($K = 4$) | 70.37 | 76.90 |
| m-LinCos-Softmax<br>(s=20,$m_1$=1,$m_2$=0,$m_3$=0.2, $K = 1$) | 82.44 | 87.09 |
| m-LinCos-Softmax<br>(s=20,$m_1$=1,$m_2$=0,$m_3$=0.2, $K = 2$) | **82.93** | **87.68** |

**TABLE 4. Identification and verification performance (%) on MegaFace challenge 1 when training on MS1M.**

| Method | Ident. | Veri. |
|---|---|---|
| Lin-Softmax | 78.18 | 83.63 |
| Cos-Softmax | 83.51 | 86.37 |
| LinCos-Softmax ($K = 2$) | **84.30** | **88.11** |
| m-LinCos-Softmax<br>(s=20,$m_1$=1,$m_2$=0,$m_3$=0.2, $K = 1$) | 87.15 | 90.51 |
| m-LinCos-Softmax<br>(s=20,$m_1$=1,$m_2$=0,$m_3$=0.2, $K = 2$) | **88.69** | **91.22** |



(a) CMC (CASIA) (b) ROC (CASIA)

(c) CMC (MS1M) (d) ROC (MS1M)

**FIGURE 6. CMC and ROC curves of different *K* settings on MegaFace challenge 1 for training on CASIA and MS1M respectively.**

MegaFace identification. This showed that the linear cosine logit can achieve a good trade-off between cosine logit and linear logit to improve feature representation. Under margin enhancement with identical margin settings (the last two rows of Table 2 and Table 3), the m-LinCos-Softmax still obtained a higher performance for $K = 2$ than $K = 1$, with an around 0.5% improvement of MegaFace identification and a 0.6% improvement of MegaFace verification for $K = 2$ compared to $K = 1$. These results showed that our proposed linear-cosine logit can effectively improve performance under both margin and non-margin settings.

The results for training on MS1M in Table 4 further demonstrated the superiority of our proposed linear cosine logit, which obtained a 0.79% improvement of MegaFace identification over the cosine logit under non-margin setting and a 1.54% improvement under identical margin settings. We also observed that the Lin-Softmax obtained a significantly lower performance than Cos-Softmax and LinCos-Softmax. This is due to the negative impact of the gradient amplifying problem of the linear logit. Hence, selecting a proper $K$ to achieve an appropriate trade-off between cosine logit and linear logit to enhance the angular discriminability and generalization capability of features is important. We found that $K = 2$ worked best in most cases.

In addition, the CMC and ROC curves in Fig. 6 also demonstrated a consistently larger envelop of LinCos-Softmax using $K = 2$ than those of Cos-Softmax and Lin-Softmax under non-margin settings. Similarly, the m-LinCos-Softmax using $K = 2$ also obtained a larger CMC and ROC envelop than using $K = 1$ under identical margin enhancement settings. These results again validated the effectiveness of the proposed linear-cosine logit.

### C. MARGIN ENHANCEMENT EVALUATION

In this section, we evaluate the effectiveness of margin enhancement under the proposed Linear Cosine Softmax Loss framework. We compared the performance of m-LinCos-Softmax loss using different margin settings with the baseline LinCos-Softmax without margin enhancement for $K = 2$, and the results of training on CASIA and MS1M are summarized in Table 5.

We can clearly see that the m-LinCos-Softmax loss obtained significant performance improvement compared to the baseline LinCos-Softmax in all the margin settings. Although the performance on LFW is close to saturation, obvious performance gains can be observed in other evaluation sets. An improvement of over 8% in MegaFace identification and an improvement of over 6% in MegaFace verification is achieved for all the margin settings when training on CASIA, while a performance gain of more than 4% in MegaFace identification is obtained when training on MS1M.

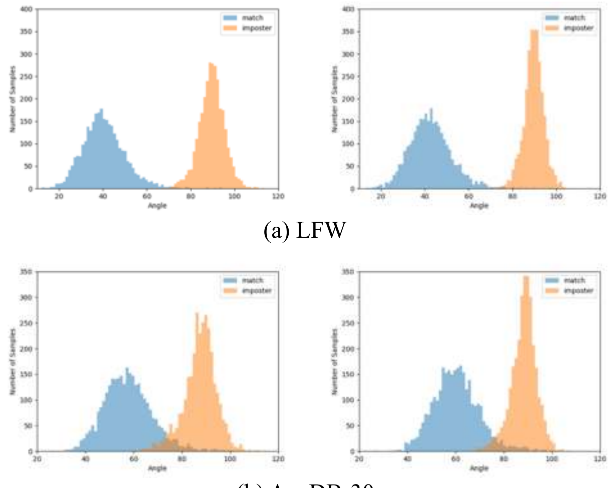
We also compared the angle distribution of face pairs on LFW and AgeDB-30 between with margin enhancement and without margin enhancement when trained on MS1M in Fig. 7. We can observe that the m-LinCos-Softmax loss achieved a more compact angle distribution with smaller overlapping confusion regions compared to its non-margin counterpart LinCos-Softmax, further verifying the effectiveness of margin enhancement under our proposed loss framework. In addition, we also illustrated the training process in Fig. 8 by showing the performance convergence on different validation sets during training. It showed that the proposed m-LinCos-Softmax loss using various margin combinations can converge without difficulty when trained on both CASIA and MS1M. Besides, it also demonstrated an obvious performance gain by the margin enhancement compared to the baseline blue curve without using margins.

**TABLE 5. Performance (%) comparison of LinCos-Softmax and m-LinCos-Softmax using different margin settings for $K = 2$.**

| | CASIA | | | | | MS1M | | | |
|---|---|---|---|---|---|---|---|---|---|
| | LFW | AgeDB-30 | CFP-FP | MF1 | | LFW | AgeDB-30 | MF1 | |
| | | | | Ident. | Veri. | | | Ident. | Veri. |
| LinCos-Softmax | 98.82 | 91.05 | 93.11 | 73.38 | 80.53 | 99.47 | 95.10 | 84.30 | 88.11 |
| $m_1$=1, $m_2$=0, $m_3$=0.2 | 99.20 | 92.30 | 93.81 | 82.93 | 87.68 | 99.53 | 96.12 | 88.69 | 91.22 |
| $m_1$=1, $m_2$=0.1, $m_3$=0.2 | 99.02 | 92.38 | 93.61 | 82.95 | 87.03 | 99.60 | 96.10 | 88.54 | 90.85 |
| $m_1$=1.1, $m_2$=0.1, $m_3$=0.1 | 98.93 | 92.62 | 93.50 | 82.21 | 86.60 | 99.68 | 96.05 | 88.90 | 90.84 |

**TABLE 6. Performance (%) comparison of the proposed methods and state-of-the-art loss functions.**

| Method | CASIA | | | | | MS1M | | | |
|---|---|---|---|---|---|---|---|---|---|
| | LFW | AgeDB-30 | CFP-FP | MF1 | | LFW | AgeDB-30 | MF1 | |
| | | | | Ident. | Veri. | | | Ident. | Veri. |
| Softmax | 97.13 | 84.23 | 86.14 | 54.8 | 56.39 | 98.25 | 91.28 | 69.56 | 71.7 |
| Center Loss [30] | 98.65 | 89.18 | 92.33 | 70.96 | 75.12 | 99.53 | 94.02 | 78.79 | 82.31 |
| Cos-Softmax [24] | 98.67 | 89.63 | 92.19 | 68.12 | 75.38 | 99.55 | 95.00 | 83.51 | 86.37 |
| SphereFace [20] | 99.15 | 92.20 | 92.33 | 79.41 | 83.94 | 99.48 | 95.40 | 87.15 | 89.85 |
| CosFace [21] | 99.17 | 91.98 | 93.70 | 82.44 | 87.09 | 99.58 | 95.95 | 87.15 | 90.51 |
| ArcFace [22] | 99.05 | 92.23 | 93.89 | 80.30 | 85.93 | 99.52 | 95.97 | 87.89 | 90.27 |
| MV-AM-Softmax [47] | 99.07 | 92.35 | 93.40 | 81.60 | 86.62 | 99.52 | 95.78 | 88.08 | 91.12 |
| m-LinCos-Softmax | **99.20** | **92.30** | **93.81** | **82.93** | **87.68** | **99.53** | **96.12** | **88.69** | **91.22** |



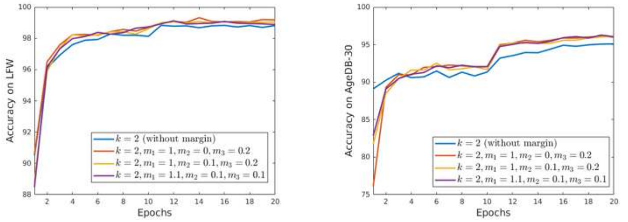
(a) LFW

(b) AgeDB-30

**FIGURE 7. Angle distributions of face pairs on LFW and AgeDB-30 when trained on MS1M. Left column: LinCos-Softmax ($K = 2$); Right column: m-LinCos-Softmax ($s = 20$, $m_1$ =1, $m_2$ =0, $m_3$ =0.2, $K = 2$).**

### D. COMPARING WITH STATE-OF-THE-ART METHODS

In this section, we compared our proposed method with several well-known face recognition benchmarks. The performance comparison results for training on CASIA and MS1M are summarized in Table 6, and the corresponding CMC and ROC curves of different methods are illustrated in Fig. 9. For m-LinCos-Softmax, we use $s = 20$, $K = 2$, $m_1 = 1$, $m_2 = 0$, $m_3 = 0.2$. For Center loss [30], we use a weight factor of 0.002 for the center loss term. For SphereFace [20], we use a margin of 3. For CosFace [21] and ArcFace [22], we use a margin of 0.2 and a scale of 20.

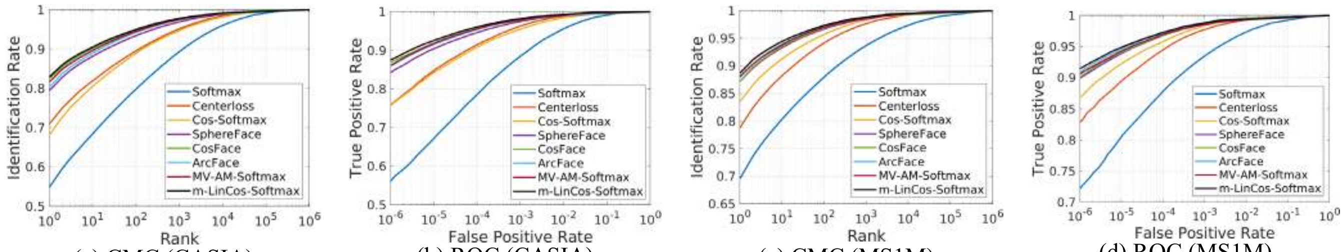


(a) LFW (CASIA)

(b) AgeDB-30 (MS1M)

**FIGURE 8. Performance convergence of m-LinCos-Softmax on LFW and AgeDB-30 during the training process on CASIA and MS1M using various margin settings.**

For MV-AM-Softmax [47], we use a margin of 0.2, a scale of 20 and t = 0.1 with fixed weights. We can find that the proposed m-LinCos-Softmax obtained a superior or competitive performance compared to the state-of-the-art angle-based softmax loss variants, SphereFace, CosFace, ArcFace, and MV-AM-Softmax. Comparing m-LinCos-Softmax with CosFace, 0.5% improvement was obtained for MegaFace identification when trained on CASIA while 1.5% improvement can be obtained when training on MS1M. Actually, the CosFace can be considered as a special case of the m-LinCos-Softmax loss for $K = 1$. Comparing m-LinCos-Softmax with ArcFace, 2.6% improvement is obtained in MegaFace identification for CASIA and a 0.8% improvement is obtained for MS1M. Comparing with MV-AM-Softmax, 1.33% and 0.61% improvements in MegaFace identification are obtained for CASIA and MS1M respectively. The CMC and ROC curves in Fig. 9 also showed that our proposed m-LinCos-Softmax had a larger envelop than the other losses, showing that the proposed method can effectively improve the performance of face recognition.



(a) CMC (CASIA) (b) ROC (CASIA) (c) CMC (MS1M) (d) ROC (MS1M)

**FIGURE 9.** CMC and ROC curves of the proposed methods and state-of-the-art loss functions on MegaFace challenge 1.

**TABLE 7.** A comparison of the proposed loss with relevant face recognition benchmarks on LFW.

| Method | Data | Models | Layers | LFW |
|---|---|---|---|---|
| Deep Face [17] | 4.4M | 3 | / | 97.35 |
| DeepID2+ [43] | 300K | 25 | / | 99.47 |
| VGG Face [18] | 2.6M | 1 | 16 | 98.95 |
| FaceNet [13] | 200M | 1 | 22 | 99.63 |
| Baidu [42] | 1.3M | 1 | / | 99.13 |
| Center Loss [30] | 0.7M | 1 | / | 99.28 |
| L2-Softmax [26] | 3.7M | 1 | 101 | 99.78 |
| AM-Softmax [23] | 0.5M | 1 | 20 | 99.12 |
| L-Softmax [19] | 0.5M | 1 | / | 98.71 |
| SphereFace [20] | 0.5M | 1 | 64 | 99.47 |
| CosFace [21] | 5M | 1 | 64 | 99.73 |
| ArcFace [22] | 5.8M | 1 | 101 | 99.83 |
| MV-AM-Softmax [47] | 3.28M | 1 | / | 99.79 |
| **Ours** | 3.8M | 1 | 20 | **99.68** |

In Table 7, we also listed a performance comparison on LFW with more relevant methods based on their reported results. We can find that our method obtained a competitive performance compared with various face recognition benchmarks. Although our method did not obtain the best performance on LFW, it did achieve a good trade-off in terms of performance, training size as well as model complexities.

## IV. CONCLUSIONS

In this paper, we proposed a Linear-Cosine Softmax Loss to effectively learn angle-discriminative face features. By using a linearity-enhanced cosine logit derived by Taylor expansion, our proposed loss function can more sufficiently optimize the angle between the feature and the corresponding weight vectors to enhance angular discriminability of features and achieve better generalization. We also designed an automatic scale parameter selection scheme, which can automatically determine an appropriate scale without exhaustive parameter tuning to save time and improve performance. In addition, we further improved our method by applying margin enhancement to the proposed loss framework. Experimental results on well-known face recognition dataset (LFW, AgeDB-30, CFP-FP, MegaFace) showed that the proposed linear cosine logit can effectively improve the performance of face recognition models under both margin and non-margin settings, and the margin enhancement can bring significant performance improvements to obtain a superior performance than the well-known angular softmax loss variants. Finally, mining angular discriminative information in the feature space to improve feature representation is a very promising approach, we will continue this direction in our future study.

## REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. CVPR*, Jun. 2015, pp. 1–9.

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

[5] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-celeb-1M: A dataset and benchmark for large-scale face recognition," in *Proc. ECCV*, 2016, pp. 87–102.

[6] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces 'Real-Life' Images, Detection, Alignment, Recognit.*, 2008.

[7] 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 *Proc.WACV*, Mar. 2016, pp. 1–9.

[8] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "AgeDB: The first manually collected, in-the-wild age database," in *Proc. CVPRW*, Jul. 2017, pp. 51–59.

[9] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The MegaFace benchmark: 1 million faces for recognition at scale," in *Proc. CVPR*, Jun. 2016, pp. 4873–4882.

[10] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[11] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification verification," in *Proc. NeurIPS*, 2014, pp. 1988–1996.

[12] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. CVPR*, 2006, pp. 1735–1742.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, Jun. 2015, pp. 815–823.

[14] W. Ge, "Deep metric learning with hierarchical triplet loss," in *Proc. ECCV*, 2018, pp. 269–285.

[15] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proc. ICCV*, Oct. 2017, pp. 2840–2848.

[16] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[17] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, ''DeepFace: Closing the gap to human-level performance in face verification,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[18] O. M. Parkhi, A. Vedaldi, and A. Zisserman, ''Deep face recognition,'' in *Proc. BMVC*, 2015.

[19] W. Liu, Y. Wen, Z. Yu, and M. Yang, ''Large-margin softmax loss for convolutional neural networks,'' in *Proc. ICML*, 2016, p. 7.

[20] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, ''SphereFace: Deep hypersphere embedding for face recognition,'' in *Proc. CVPR*, Jul. 2017, pp. 212–220.

[21] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, ''CosFace: Large margin cosine loss for deep face recognition,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[22] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, ''ArcFace: Additive angular margin loss for deep face recognition,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[23] F. Wang, J. Cheng, W. Liu, and H. Liu, ''Additive margin softmax for face verification,'' *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.

[24] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, ''NormFace: L2 hypersphere embedding for face verification,'' 2017, *arXiv:1704.06369*. [Online]. Available: http://arxiv.org/abs/1704.06369

[25] Y. Liu, H. Li, and X. Wang, ''Rethinking feature discrimination and polymerization for large-scale recognition,'' 2017, *arXiv:1710.00870*. [Online]. Available: http://arxiv.org/abs/1710.00870

[26] R. Ranjan, C. D. Castillo, and R. Chellappa, ''L2-constrained softmax loss for discriminative face verification,'' 2017, *arXiv:1703.09507*. [Online]. Available: http://arxiv.org/abs/1703.09507

[27] C. Luo, J. Zhan, X. Xue, L. Wang, R. Ren, and Q. Yang, ''Cosine normalization: Using cosine similarity instead of dot product in neural networks,'' in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2018, pp. 382–391.

[28] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, ''AdaptiveFace: Adaptive margin and sampling for face recognition,'' in *Proc. CVPR*, Jun. 2019, pp. 11947–11956.

[29] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, ''AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations,'' in *Proc. CVPR*, Jun. 2019, pp. 10823–10832.

[30] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, ''A discriminative feature learning approach for deep face recognition,'' in *Proc. ECCV*, 2016, pp. 499–515.

[31] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, ''A comprehensive study on center loss for deep face recognition,'' *Int. J. Comput. Vis.*, vol. 127, pp. 668–683, Jun. 2019.

[32] K. Zhao, J. Xu, and M.-M. Cheng, ''RegularFace: Deep face recognition via exclusive regularization,'' in *Proc. CVPR*, Jun. 2019, pp. 1136–1144.

[33] Y. Duan, J. Lu, and J. Zhou, ''UniformFace: Learning deep equidistributed representation for face recognition,'' in *Proc. CVPR*, Jun. 2019, pp. 3415–3424.

[34] J. Deng, Y. Zhou, and S. Zafeiriou, ''Marginal loss for deep face recognition,'' in *Proc. CVPRW*, Jul. 2017, pp. 60–68.

[35] A. Calefati, M. K. Janjua, S. Nawaz, and I. Gallo, ''Git loss for deep face recognition,'' in *Proc. BMVC*, 2018.

[36] Y. Wu, H. Liu, J. Li, and Y. Fu, ''Deep face recognition with center invariant loss,'' in *Proc. Thematic Workshops ACM Multimedia-Thematic Workshops*, 2017, pp. 408–414.

[37] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, ''Range loss for deep face recognition with long-tailed training data,'' in *Proc. ICCV*, Oct. 2017, pp. 5409–5418.

[38] Y. Zheng, D. K. Pal, and M. Savvides, ''Ring loss: Convex feature normalization for face recognition,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5089–5097.

[39] W. Wan, Y. Zhong, T. Li, and J. Chen, ''Rethinking feature distribution for loss functions in image classification,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9117–9126.

[40] V. Li and A. Maki, ''Feature contraction: New convnet regularization in image classification,'' in *Proc. BMVC*, 2018, p. 213.

[41] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, ''Targeting ultimate accuracy: Face recognition via deep embedding,'' 2015, *arXiv:1506.07310*. [Online]. Available: http://arxiv.org/abs/1506.07310

[42] Y. Sun, X. Wang, and X. Tang, 'Deeply learned face representations are sparse, selective, and robust,'' in *Proc. CVPR*, 2015, pp. 2892–2900.

[43] K. Sohn, ''Improved deep metric learning with multi-class n-pair loss objective,'' in *Proc. NIPS*, 2016, pp. 1857–1865.

[44] W. Chen, X. Chen, J. Zhang, and K. Huang, ''Beyond triplet loss: A deep quadruplet network for person re-identification,'' in *Proc. CVPR*, Jul. 2017, pp. 403–412.

[45] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, ''Deep metric learning via lifted structured feature embedding,'' in *Proc. CVPR*, Jun. 2016, pp. 4004–4012.

[46] B. Chen, W. Deng, and H. Shen, ''Virtual class enhanced discriminative embedding learning,'' in *Proc. NIPS*, 2018, pp. 1942–1952.

[47] X. Wang, S. Zhang, S. Wang, T. Fu, H. Shi, and T. Mei, ''Mis-classified vector guided softmax loss for face recognition,'' in *Proc. AAAI*, 2020.

[48] L. He, Z. Wang, Y. Li, and S. Wang, ''Softmax dissection: Towards understanding intra-and inter-clas objective for embedding learning,'' in *Proc. AAAI*, 2020.

**WEI-FENG OU** received the B.Eng. degree from the Guangdong University of Technology, in 2013, and the M.Eng. degree from the South China University of Technology, in 2016. He is currently pursuing the Ph.D. degree with the City University of Hong Kong. He was an Engineer with Huawei, from 2016 to 2018. His research interests include deep learning and computer vision.

**LAI-MAN PO** (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electronic engineering from the City University of Hong Kong, Hong Kong, in 1988 and 1991, respectively. He has been with the Department of Electronic Engineering, City University of Hong Kong, since 1991, where he is currently an Associate Professor with the Department of Electrical Engineering. He has authored over 150 technical journals and conference papers. His research interests include image and video coding with an emphasis deep learning based computer vision algorithms.

Dr. Po is a member of the Technical Committee on Multimedia Systems and Applications and the IEEE Circuits and Systems Society. He was the Chairman of the IEEE Signal Processing Hong Kong Chapter, in 2012 and 2013. He also served on the Organizing Committee of the IEEE International Conference on Acoustics, Speech and Signal Processing, in 2003, and the IEEE International Conference on Image Processing, in 2010. He was an Associate Editor of *HKIE Transactions*, in 2011 to 2013.

**CHANG ZHOU** (Graduate Student Member, IEEE) received the B.Sc. degree from Donghua University, China, Shanghai, in 2016, and the master's degree from the City University of Hong Kong, Hong Kong, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His research interests are in computer vision and deep learning.

**YU-JIA ZHANG** received the B.E. degree in electrical engineering and automation from the Huazhong University of Science and Technology, in 2015, and the M.S. degree in electrical engineering from the South China University of Technology, China, in 2018. He is currently pursuing the Ph.D. degree with the City University of Hong Kong. His current research interests include human activity recognition and computer vision.

**LI-TONG FENG** received the B.Eng. degree from the Harbin Institute of Technology, in 2008, and the Ph.D. degree from the City University of Hong Kong, in 2016. He is currently a Senior Researcher with SenseTime. His research interests include image classification and object detection.

**YASAR ABBAS UR REHMAN** (Member, IEEE) received the B.Sc. degree in electrical engineering (telecommunication) from the City University of Science and Information Technology, Peshawar, Pakistan, in 2012, the M.Sc. degree in electrical engineering from the National University of Computer and Emerging Sciences, Pakistan, in 2015, and the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2019. He is currently working with TCL Corporate Research (HK) Company Ltd., as a Postdoctoral Researcher. His research interests include computer vision, machine learning, deep learning, and its applications in facial recognition, biometric anti-spoofing, and video understanding.

**YU-ZHI ZHAO** (Graduate Student Member, IEEE) received the B.Eng. degree in electronic information from the Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, City University of Hong Kong. His research interests include low-level vision and deep learning.