Unifying Margin-Based Softmax Losses in Face Recognition

Yang Zhang*, Simao Herdade*, Kapil Thadani, Eric Dodds, Jack Culpepper, and Yueh-Ning Ku

Yahoo Research

Abstract

In this work, we develop a theoretical and experimental framework to study the effect of margin penalties on angular softmax losses, which have led to state-of-the-art performance in face recognition. We also introduce a new multiplicative margin which performs comparably to previously proposed additive margins when the model is trained to convergence. A regime of the margin parameters can lead to degenerate minima, but these can be reliably avoided through the use of two regularization techniques that we propose. Our theory predicts the minimal angular distance between sample embeddings and the correct and wrong class prototype vectors learned during training, and it suggests a new method to identify optimal margin parameters without expensive tuning. Finally, we conduct a thorough ablation study of the margin parameters in our proposed framework, and we characterize the sensitivity of generalization to each parameter both theoretically and through experiments on standard face recognition benchmarks.

1. Introduction

Deep learning models trained with margin softmax losses achieve state-of-the-art performance on standard metric learning benchmarks such as face verification and identification [8, 10, 15, 27, 3, 31] and fine-grained classification [20]. However, the existing literature fails to explain why different proposed margin penalties result in better generalization. Claims have been made that reducing intra-class distance helps generalization [27], or that enforcing class separation in angle space is more effective than in cosine space [3], but such claims have not been proven or empirically verified. Furthermore, works reporting competitive performance use different model architectures, training schedules, learning rate schedules, batch sizes, and/or data augmentation paradigms, as well as different test time augmentation strategies and test set preprocessing. These differences obscure the effects of the primary contributions of these works on the quality of their models as measured by benchmark performance. Indeed it has been shown that the improvement of new metric learning loss functions over previous work is often smaller than stated when holding fixed these confounding factors [20].

This work serves as a first attempt to characterize how different margin penalties affect the training optimization as well as the resulting model generalization. Through a comprehensive fair comparison on face recognition benchmarks, we demonstrate that all primary margin parameters on the softmax loss—including CosFace [27], ArcFace [3] and a natural new multiplicative margin which has not been studied before—are fundamentally alike, though they differ in optimization difficulty. We argue that prevailing explanations for margin effectiveness are insufficient and develop a new theoretical framework to better understand the mechanisms underlying the success of margin softmax losses. Specific contributions of our work include:

- A mathematical model of prototypes and samples that enables us to study training dynamics and optimization stopping points as a function of margin values. This allows us to approximate the influence of margin settings on intra-class and inter-class distances.
- A formal characterization of a collapse mode that occurs in some margin parameter regimes. We propose two effective regularization techniques to avoid these degenerate minima without harming performance.
- A natural new margin parameter to complete the family of margin-based softmax losses. Our experiments show that all margin formulations yield similar performance when models are trained to convergence.
- A new observation that optimal values for different margins produce loss functions that appear to coincide at a specific angle which can be analytically derived for any dataset. This leads to a conjecture on how to select optimal margin parameters without computationally expensive tuning experiments.

*These authors contributed equally to this work.

2. Related Work

Face recognition, one of the oldest problems in computer vision, has matured significantly with the advent of large databases and deep feature learning. Facenet [22], an early example of the leverage gained with these tactics, utilizes a metric learning approach based on pairs of similar and dissimilar images. Since the emergence of equivalences between pairwise and proxy-based metric learning [19, 24], advancements in face verification have mainly utilized multi-class classification losses for training [21, 26, 15, 34, 27, 3, 12, 31, 10, 8, 6, 1] which have higher memory requirements but are less reliant upon complex and computationally expensive hard (negative) example mining.

Proxy-based metric learning has converged on the use of a modified softmax loss where each logit is the cosine of the angle between a feature vector and a weight matrix column. A global scaling factor characterizing the "temperature" or sparseness of the softmax then multiplies the cosine as a fixed hyperparameter optimized by cross-validation or as a learnable model parameter [26, 31]. Introducing margin parameters which multiply the angle [15], add to the angle [3], or add to the cosine of the angle [27] have been shown to improve generalization, and to simultaneously reduce intra-class angles and increase inter-class angles. Gains are observed when heavy-tailed class distributions in the training data are addressed by per-class margins with larger margins when classes have fewer training examples [12]. To model variability in sample difficulty, margins can be formulated as a function of the magnitude of the feature vector [17].

While some attention has been paid to their individual effects, we are not aware of a comprehensive analysis of the ways these modifications to the softmax loss affect training. Several studies have characterized the effect of normalization in the modified softmax loss. D-Softmax [6] identifies the entanglement of the intra-class and inter-class optimizations and proposes a new loss to optimize intra-class variation and inter-class distance separately. CircleLoss [24] reveals the sub-optimal magnitude of the gradients throughout the training and proposes a second order margin to correct that. Similarly, we study the intra- and inter-class optimization process both theoretically and empirically in this work.

3. Margins on the Hypersphere

3.1. Unified Angular Margin Softmax Losses

A standard metric learning approach to face recognition is to learn an embedding function such that the embedding $x_i \in \mathbb{R}^d$ of an example input i (i.e., a face image) is closer to other examples of the same class (identity) than to examples from other classes, without knowing about the classes in advance. Following prior work, we learn this function by optimizing a modified classification objective on a training set with C known classes represented by a learnable matrix $W \in \mathbb{R}^{d \times C}$ where we refer to each column W_j as the "prototype" of class j. Taking the dot product as our measure of similarity, the softmax cross-entropy loss for N examples can be written as an average of per-example losses $L_1 \dots L_N$ such that

$$L_i = -\log \frac{e^{W_{y_i}^\top x_i}}{\sum_{j=1}^{C} e^{W_j^\top x_i}} \tag{1}$$

where y_i is the class of the ith example.

For any given example i, we also have $W_j^\top x_i = \|W_j\| \|x_i\| \cos\theta_j$, where θ_j is the angle between column W_j and feature vector x_i. By normalizing W_j and x_i to unit vectors and incorporating a global scaling parameter s for all logits [15, 26], the loss in equation (1) becomes

$$L_i = -\log \frac{e^{s\cos\theta_{y_i}}}{\sum_{j=1}^{C} e^{s\cos\theta_j}} \tag{2}$$

where $0 \le \theta_j \le \pi$ for all angles θ_j.

Prior work introduced margin parameters in the correct-class exponent which are multiplied with the angle [15], added to the angle [3] or to its cosine [27]. Following [3], incorporating these margins can be summarized as replacing $\cos\theta_{y_i}$ in the angular softmax loss (2) by $z_{y_i} = \cos(m_1\theta_{y_i} + m_2) - m_3$ so that the loss becomes

$$\begin{aligned} L_i &= -\log \frac{e^{sz_{y_i}}}{e^{sz_{y_i}} + \sum_{j \neq y_i} e^{sz_j}} \\ &= \log\left(1 + \frac{\sum_{j \neq y_i} e^{sz_j}}{e^{sz_{y_i}}}\right) \end{aligned} \tag{3}$$

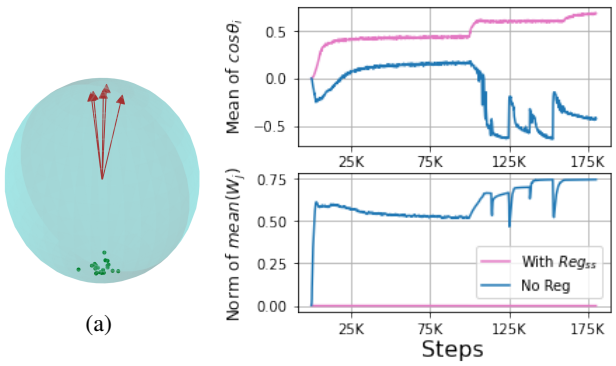
where the wrong-class exponents $z_j = \cos(\theta_j)$ do not have the margins. Each margin can be used during training to encourage smaller correct-class angles than the ordinary softmax loss. The margins are typically fixed hyperparameters but can also be learned [12].

Previously introduced margins m_1, m_2, m_3 modulate respectively the period, phase, and vertical shift of the function $\cos(\theta_{y_i})$. We complete the framework of angular margins with a novel margin parameter m_0 that modulates the amplitude of $\cos(\theta_{y_i})$ by redefining z_{y_i} as

$$z_{y_i} = m_0 \cos(m_1\theta_{y_i} + m_2) - m_3\,, \tag{4}$$

and use the name AmpFace for the family of models defined by equation (4) with $0 < m_0 < 1$. Note that the effect of this margin parameter cannot be subsumed by the scale parameter s in the softmax loss as m_0 affects *only* the positive logit while s appears in all terms, scaling positive and negative logits alike.

These four margin hyperparameters, together with the scale parameter s, play important roles in generalization for metric learning models. We study each margin penalty independently by analyzing its impact on the training dynamics.



(a)

(b)

Figure 1: (a): Illustration of polar collapse: prototypes (red arrows) cluster together and feature vectors (green circles) cluster in the opposite direction. (b): Training dynamics for $m_0 = 0.35$ demonstrating polar collapse when training with (purple) and without (blue) regularization. Top: minibatch mean of $\cos(\theta_i)$. Bottom: L2 norm of mean prototype.

3.2. Polar Collapse

Using the margin parameters described above can create an undesirable minimum of the loss function, which we call the *polar collapse* mode, since the model may collapse towards this degenerate configuration rather than learning a useful representation. Fig. 1a illustrates the polar collapse configuration in $d = 3$. Blue curves in Fig. 1b log typical training dynamics for $m_0 = 0.35$. After around 100k optimization steps (upper subplot, blue trace), the mean correct-class angle θ_i increases rapidly towards π. Polar collapse drives the norm of the average prototype $\frac{1}{C}\sum_j W_j$ towards 1, indicating that the prototypes are converging towards a single point (lower subplot). In the limit of total collapse, all angles equal π *i.e.* $\theta_{y_i} = \theta_j = \pi$, and the per-example loss (3) reduces to:

$$L_i^{\text{collapse}} = \log\left(1 + \frac{C-1}{e^{s(z'+1)}}\right) \tag{5}$$

where $z' = m_0\cos(m_1\pi + m_2) - m_3$. With no margins applied ($m_0 = m_1 = 1$ and $m_2 = m_3 = 0$), we have $z' = -1$ and so $L_i = \log C$, which is an appropriately large loss for an undesirable model. However, using aggressive margins can result in $e^{s(z'+1)} \gg C - 1$, which would decrease L_i to zero. In geometric terms, the optimizer could simply minimize the training loss by collapsing all feature vectors x_i to a single point and all prototypes W_j to the opposite pole on the d-dimensional hypersphere. Figure 2 shows how the loss L_i varies with a constant distance to all prototypes and with each margin parameter while keeping the other margins fixed; angles and margin settings that bring this loss to near-zero are in danger of inducing the collapse mode. In practice, polar collapse is regularly observed for multiplicative margins, e.g., SphereFace with $m_1 \geq 1.8$ or AmpFace with $m_0 \leq 0.65$.

Note that this collapse mode is different from the triplet loss collapse observed in [22], in that the triplet loss always has a bad *local* minimum where all feature vectors collapse, and this problem can be mitigated by optimization and sampling strategies. In contrast, the margin softmax polar collapse described here may be a local *or global* minimum of the loss, so reliably avoiding it requires modifications to the loss function.

We propose two strategies to avoid this polar collapse during model training:

1. **Spherical Symmetry Regularization (Reg_{ss})**

   Add a regularization term to the total loss with weight λ that encourages prototypes W_j to be symmetrically distributed on the hypersphere manifold by minimizing the L2-norm of the mean normalized prototype.

   $$\text{Reg}_{\text{ss}} = \left\| \frac{1}{C}\sum_j \frac{W_j}{\|W_j\|_2} \right\|_2 \tag{6}$$

2. **Wrong-class Logit Rectification (WC-ReLU)**

   Rectify the values of z_j in equation (3) using a ReLU activation, denoted by $[\cdot]_+$.

   $$L_i = \log\left(1 + \frac{\sum_{j\neq y_i} e^{s[z_j]_+}}{e^{sz_{y_i}}}\right) \tag{7}$$

   When $\theta_j > \frac{\pi}{2}$, the gradient $\partial L_i/\partial\theta_j$ becomes 0 to avoid driving x_i and W_j to opposite poles. Figure 2 shows that no margin setting results in zero loss for examples which are $\frac{\pi}{2}$ away from prototypes.

To our knowledge, this is the first consideration in the literature of this common failure mode for softmax losses with margin penalties. The ad-hoc easy-margin trick from the original implementation of ArcFace [3] does not extend to multiplicative margins because removing a fractional multiplicative factor on the correct-class logit (when the corresponding cosine is negative) makes its value *smaller* and the correct classification harder. Reg_{ss} has been studied previously in the context of generalization through hyperspherical uniformity [13]. UniformFace [4] was not originally proposed as a regularization method but shares the same high-level intuition as Reg_{ss}. Both encourage the symmetric distribution of prototypes W_j, although the calculation of Reg_{ss} is simpler and more efficient. We empirically compare our two proposed strategies with UniformFace in Section 4.4 and show that either Reg_{ss} or WC-ReLU allows a significantly larger hyperparameter search space where regular training would normally fail due to polar collapse.

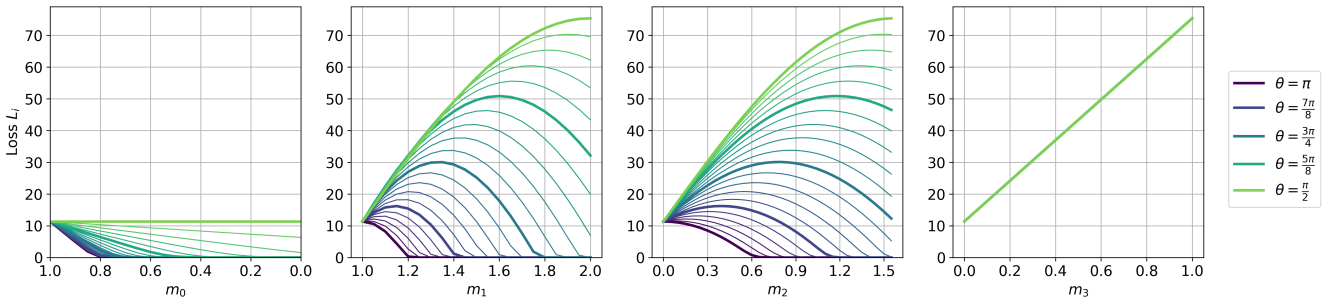


Figure 2: Loss for an example i which has an angular distance θ to all prototypes as a function of each margin parameter. As loss approaches zero, further training on such an example cannot draw x_i closer to its prototype W_{y_i}. This scenario can occur with even small multiplicative margins (m_0 and m_1) but cannot be caused by increasing m_3. Best viewed in color.

3.3. The Hypersphere Manifold

In this section, we make theoretical predictions for the average smallest wrong-class angle $M_{\text{wrong}} = \mathbb{E}_i[\min_j \theta_j]$ and the average correct-class angle $M_{\text{correct}} = \mathbb{E}_i \theta_{y_i}$ for an example i using properties of high-dimensional spaces. In Appendix A, we provide numerical simulations for our experimental setting that verify these expressions obtained in the limit of large C and d.

The class prototypes W_j represent points on the hypersphere of dimension $d-1$. Since we initialize the entries of W to i.i.d. samples from a normal distribution (before normalizing), the initial prototypes are approximately uniformly distributed on the surface of the hypersphere after normalization. We theoretically model optimization and generalization as a function of the different margin parameters, under the assumption of a spherical uniform distribution of the prototypes *throughout training*.

Our assumption of prototypes distributed uniformly on the hypersphere predicts an approximate value for the angle from a prototype to the nearest prototype for large d [2, 3].

Proposition 1 *Let $W_i \in \mathbb{R}^d$ for $i = 1, \ldots, C$ be i.i.d. from the uniform distribution on the unit sphere. Then the expected angle from a vector W_i to its nearest neighbor, $\theta_{min}(W_i) = \min_{j \neq y_i} \arccos(W_i^\top W_j)$, converges to*

$$\mathbb{E}\left[\theta_{min}(W_i)\right] \xrightarrow[C\to\infty]{} C^{\frac{-2}{d-1}} \Gamma\left(\frac{d}{d-1}\right) \left(\frac{\Gamma\left(\frac{d}{2}\right)}{2\sqrt{\pi}(d-1)\Gamma\left(\frac{d-1}{2}\right)}\right)^{\frac{-1}{d-1}} \tag{8}$$

For large d and C, this expression also approximates the angle between a random vector (which need not be a prototype) and the nearest prototype. If we assume each feature vector has either a random direction (as is likely at initialization) or is only much closer to the correct prototype, equation (8) gives an estimate of M_{wrong}.

To predict the average correct-class distance M_{correct}, we note that the gradient of the margin softmax loss for an example with respect to the correct prototype nearly vanishes at a certain angle value. This value is dependent on the margins and the expected wrong-class angles, as discussed in [6] and examined in Section 3.4. While correct-class distances could be smaller than this angle value after optimization, they will with high probability lie very close to the value due to the concentration-of-measure phenomenon in high-dimensional hyperspheres.

Proposition 2 *Fix $0 < \epsilon \ll 1$ and define $\tau_\epsilon = \max\{0 < \theta < \pi/2 : \frac{\partial L_i}{\partial \theta_{y_i}} < \epsilon\}$ as the angle for which optimization gradients w.r.t. the positive prototype "vanish." Then*

$$M_{correct} \approx \mathbb{E}_{x \in cap(W_i, \tau_\epsilon)}[\theta_{y_i}] \xrightarrow[d\to\infty]{} \tau_\epsilon \tag{9}$$

where $cap(W_i, \tau_\epsilon)$ denotes the spherical cap centered on prototype W_i with radius subtended at the origin by τ_ϵ.

In the following section, we analyse the effect of different margins on the loss curves and attend to the angle values for which the corresponding gradients vanish. To do so we make use of the following estimate of the sum of wrong-class logits in the denominator of the softmax loss.

Proposition 3 *Assume the prototypes W_j are uniformly distributed on the hypersphere, and denote by $z_j = W_j^\top x_i = \cos(\theta_j)$, $\forall j \neq y_i$, the wrong-class logits for the ith example. Then for sufficiently large C and d ($e^{s^2/d}/C \ll 1$)*

$$\sum_{j \neq y_i} e^{s z_j} \approx (C-1) * e^{s^2/(2d)} \tag{10}$$

For large dimension d, the inner product between a fixed unit vector x_i and a random unit vector is approximately a sample from $\mathcal{N}(0, 1/\sqrt{d})$. Thus $e^{s z_j}$ is approximately log-normally distributed with mean $e^{s^2/(2d)}$. By the law of large numbers, assuming C sufficiently large, the sum converges to the expectation and we have

$$\sum_{j \neq y_i} e^{s z_j} \to (C-1) * \mathbb{E}[e^{s z_j}] \approx (C-1) e^{s^2/(2d)} \tag{11}$$

See Appendix A for a proof of the approximation.

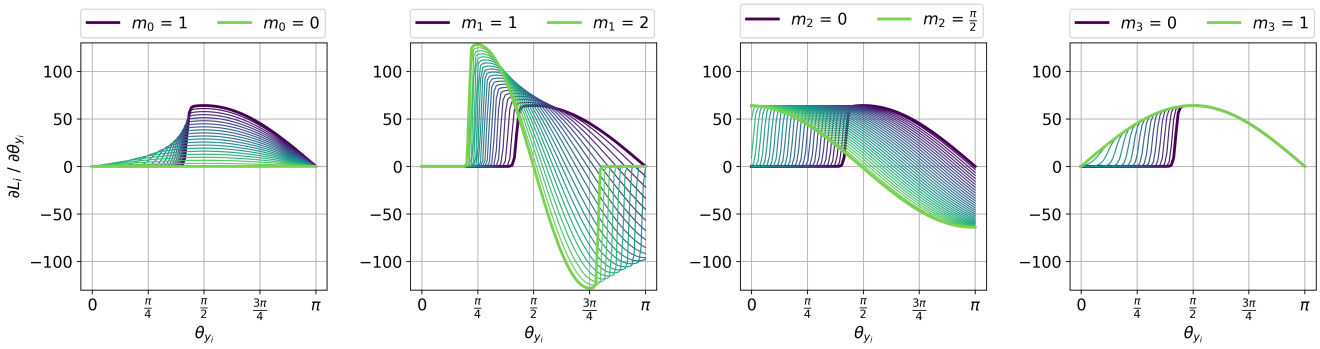


Figure 3: Gradient of the loss L_i with respect to the correct-class angle θ_{y_i} when varying each margin parameter. Margins that modify the angle (m_1 and m_2) introduce *negative* gradients for large θ_{y_i} which further separates features from their prototypes and can lead to the collapse mode. Best viewed in color. See Appendix C for additional gradient and loss plots.

3.4. Gradients and Training Dynamics

The gradient of the per-example loss from equation (3) $L_i = -\log \mathsf{P}(y_i|x_i)$ with respect to the distance θ_{y_i} to the correct class is

$$\begin{aligned}\frac{\partial L_i}{\partial \theta_{y_i}} &= -(1 - \mathsf{P}(y_i|x_i)) \cdot s \cdot \frac{\partial z_{y_i}}{\partial \theta_{y_i}} \\ &= (1 - \mathsf{P}(y_i|x_i)) \cdot s \cdot m_0 \cdot m_1 \cdot \sin(m_1\theta_{y_i} + m_2)\end{aligned} \tag{12}$$

Using the approximation of wrong-class logits from equation (10), Figure 3 illustrates the gradient $\partial L_i/\partial\theta_{y_i}$ as a function of θ_{y_i} and shows how it varies as each margin parameter is adjusted individually. The magnitude of $\partial L_i/\partial\theta_{y_i}$ for some θ_{y_i} depends on $1 - \mathsf{P}(y_i|x_i)$ regardless of the margin, but also has a factor that grows linearly with the introduced margin m_0, approximately quadratically with m_1 as $m_1\theta_{y_i}/2 \leq \sin(m_1\theta_{y_i}) \leq m_1\theta_{y_i}$ when $m_1\theta_{y_i} \in [0, \frac{\pi}{2}]$, and non-linearly with m_2. The dependence of the gradient magnitude on the margin implies that models with different margins will converge at different rates when training hyperparameters are held fixed. Therefore, there are two fair training schedules for comparing models. The first is long enough to make sure models are close to convergence and will likely produce the best performance but may be expensive or time consuming. The second is a shorter schedule with a limited computation budget.

Inspired by [26, 6], we are interested in when $\partial L_i/\partial\theta_{y_i}$ becomes zero as θ_{y_i} decreases during training as this is the termination point for intra-class optimization. This angle is also our theoretical estimate of the average correct-class distance M_{correct}, defining the edge of the spherical cap around prototypes W_{y_i} where examples x_i accumulate as shown by Proposition 2. Figure 3 shows that this angle decreases linearly with m_2 and non-linearly with the other margins. In Section 4.3, we compare these estimates M_{correct} with empirical correct-class angles under varying margins.

4. Experiments

4.1. Implementation Details

The architecture of our feature embedding network is identical to the ArcFace [3] network: it is their ResNet-100 [5] backbone acting on input images of resolution 112x112, and outputting a feature vector of dimension 512.

Training The feature embedding network is trained on a cleaned version of the MS-Celeb-1M (C-MS1M) as provided by the authors of [3]. This database contains 5,822,653 images of 85,742 identities, and each image has already been aligned and cropped to 112×112 following standard procedures [3, 27, 15]. We train our model from random initialization with 8 NVIDIA V100 GPUs using synchronous stochastic gradient descent (SGD) and momentum 0.9. We use batch size 512 and weight decay $5e{-}4$. Our training schedule starts the learning rate at 0.1 and runs 100K steps, then runs 60K steps at 0.01, and then finally runs 20K steps at 0.001 – 180K steps in total.

Testing We evaluate on the following benchmarks:[1]

- LFW [7] which contains 13,233 images of 5,749 individuals. We follow the 'unrestricted, with labelled outside data' protocol, and evaluate our model with the dataset as is, without mislabel correction.
- CFP-FP [23] which contains 10 frontal images and 4 profiles images of 500 identities.
- AgeDB-30 [18] which is composed of 16,488 images of 568 unique subjects.
- CALFW [33] and CPLFW [32], reconstructed from LFW with additional age and pose variations.
- YTF [29] which contains 3,425 videos from 1,595 identities. We calculate the mean of embedding features from all frames of videos then evaluate pairs by their feature centre.

[1] All datasets are publicly available and are only used to benchmark our models for a fair comparison with prior work in the literature. Our work is conducted for non-commercial research purposes only.

Method	Verification Results						IJB		MegaFace	
	LFW	CALFW	CPLFW	AgeDB	CFP-FP	YTF	IJB-B	IJB-C	Id	Ver
GroupFace [10]	99.85	96.20	93.17	98.28	98.63	97.8	94.93	96.26	98.74	98.79
CurricularFace [8]	99.80	96.20	93.13	98.32	98.37	-	94.8	96.1	98.71	98.64
AmpFace ($m_0 = 0.375$)	99.76$\pm$0.02	95.53$\pm$0.09	90.93$\pm$0.20	97.82$\pm$0.17	97.72$\pm$0.20	97.71$\pm$0.20	93.02$\pm$0.49	94.54$\pm$0.47	97.93$\pm$0.05	98.06$\pm$0.07
SphereFace ($m_1 = 1.35$)	99.74$\pm$0.05	95.49$\pm$0.03	90.72$\pm$0.14	97.68$\pm$0.10	98.19$\pm$0.10	97.83$\pm$0.19	92.23$\pm$0.12	94.14$\pm$0.10	96.51$\pm$0.09	96.89$\pm$0.13
ArcFace ($m_2 = 0.5$)	99.80$\pm$0.02	95.75$\pm$0.04	91.42$\pm$0.27	98.09$\pm$0.10	98.45$\pm$0.16	97.88$\pm$0.02	94.10$\pm$0.07	95.63$\pm$0.14	98.38$\pm$0.13	98.53$\pm$0.15
CosFace ($m_3 = 0.35$)	99.79$\pm$0.01	95.75$\pm$0.05	91.60$\pm$0.09	98.06$\pm$0.13	98.31$\pm$0.13	97.85$\pm$0.09	94.22$\pm$0.28	95.75$\pm$0.12	98.16$\pm$0.10	98.39$\pm$0.06
AmpFace ($m_0 = 0.375$) $*$	99.77$\pm$0.03	95.70$\pm$0.07	92.00$\pm$0.28	98.08$\pm$0.14	98.40$\pm$0.06	97.86$\pm$0.17	94.44$\pm$0.23	95.91$\pm$0.18	98.83$\pm$0.06	98.93$\pm$0.05
ArcFace ($m_2 = 0.5$) $*$	99.82$\pm$0.01	95.82$\pm$0.04	92.05$\pm$0.11	98.21$\pm$0.14	98.67$\pm$0.06	97.88$\pm$0.13	94.91$\pm$0.06	96.18$\pm$0.02	98.80$\pm$0.07	98.98$\pm$0.04
CosFace ($m_3 = 0.5$) $*$	99.76$\pm$0.05	95.72$\pm$0.08	92.07$\pm$0.11	98.19$\pm$0.05	98.52$\pm$0.16	97.85$\pm$0.17	94.51$\pm$0.33	95.93$\pm$0.19	98.91$\pm$0.06	99.02$\pm$0.04

Table 1: Common face identification and verification benchmark results in %. For IJB, TAR@FAR=1e-4 is reported. For MegaFace, "Id" refers to the rank-1 identification accuracy against 1M distractors and "Ver" refers to the face verification task where TAR@FPR=1e-6 is reported. For a fair comparison, the first group contains only recent models trained on the cleaned version of MS-Celeb-1M [3]. The second and third groups contain averaged results over three identical runs. All runs follow the protocol described in Section 4.1 with 180K steps and a batch size of 512, except runs indicated with $*$ which are trained to convergence using 300K steps and a batch size of 1536. AmpFace models are trained with Reg_{ss}.

- MegaFace [9] which includes 1,027,060 images of 690,572 unique identities.
- IJB-B [28] which contains 1,845 subjects with 21,798 images, and 55,025 frames from 7,011 videos.
- IJB-C [16] which includes 31,334 images and 11,779 videos of 3,531 subjects, with more occlusion and diversity of subjects from IJB-B.

4.2. Model Baselines

Even though recent literature aims for a fair comparison with other work, we were unable to find consistently-reported baselines. For example, on the Refined MegaFace Identification task (R-MegaFace-Id) [3]: ArcFace [3] reports 98.35% with a model trained on CASIA-WebFace [30]. CosFace [27] makes use of a private dataset and reported 82.72% without removing the noise. GroupFace [10] reports 98.74%, despite the network architecture being significantly different; they use pre-training, a batch size of 1024 and a different learning rate schedule. CurricularFace [8] employs a training schedule of 24 epochs, which is equivalent to 270K steps when the batch size is 512, and reports 98.71%. CircleLoss [24] reports 98.50% where tail identities from C-MS1M are excluded. With the goal of establishing more consistent baselines, we carefully evaluate SphereFace, CosFace and ArcFace in Table 1 under the same training configuration as described in Section 4.1 (bs=512, 180K steps). We include models trained with the new multiplicative margin $m_0 < 1$ proposed in Section 3.1, which we name AmpFace.

As pointed out in Section 3.4, the margin parameters affect the magnitude of $\partial L_i / \partial \theta_{y_i}$ and hence some adjustment to the training hyperparameters may be needed for the model to fully converge. For comparison purposes, we also train models with an increased batch size (1536) and number of steps (300K) in each leg of the training schedule, in order to get closer to convergence; these models achieve better performance. In group three of Table 1, AmpFace has identical performance to previous state-of-the-art margins ArcFace and CosFace. And all three have comparable performance to recent state-of-the-art models in group one.

4.3. Margin Comparison

We strive to clearly understand how different margins affect the corresponding trained model optimization and generalization. In Fig. 4, we display final trained model performance on R-MegaFace-Id (bottom row) for AmpFace, SphereFace, ArcFace, and CosFace under a range of margin values. Following a few recent works [31, 6, 24] we also measure (top row) the corresponding empirical intra- and inter-class angular distances averaging at batch level by:

- $M_{\text{intra}} = \frac{1}{N} \sum_{i=1}^{N} \theta_{y_i}$
- $M_{\text{inter}} = \frac{1}{N} \sum_{i=1}^{N} \min_{j \neq y_i} \theta_j$

Lastly we compare these with their theoretical counterparts derived in Section 3.3 (middle row) under the assumption of spherically equidistributed prototypes. Comparing these predicted angle distances with the empirical values of M_{intra} and M_{inter}, we can observe that:

1. Inter-class distances do not seem to depend strongly on the value of the margin hyperparameters.
2. The test set inter-class distances are lower than corresponding training set inter-class distances, as predicted by the greater size of the test set and Proposition 1.
3. The slopes of our theoretical intra-class angle curves reflect approximately the empirical results where the

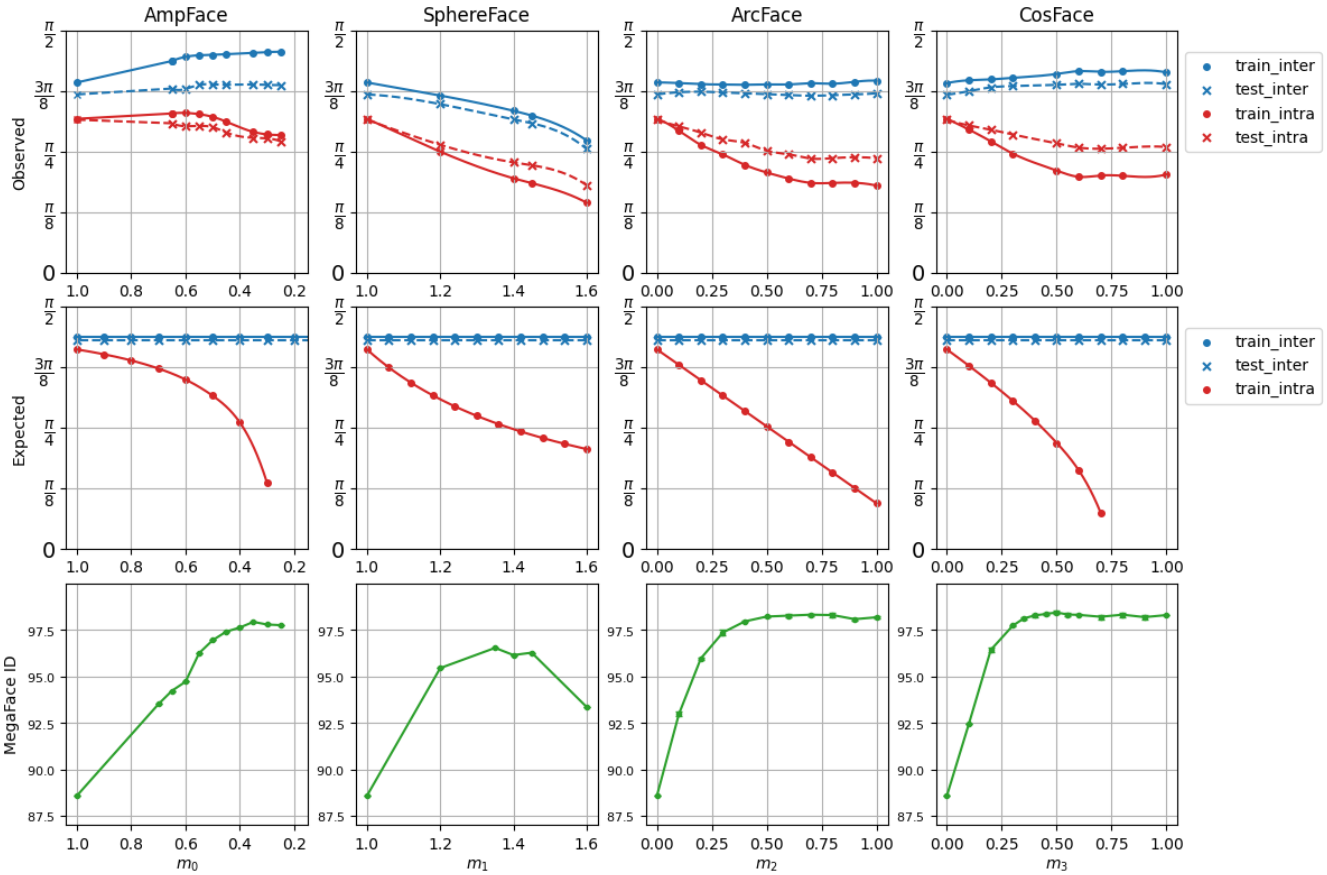


Figure 4: Dependence on margin value of observed inter- and intra-class angles on train and test sets (top row), our predictions of these quantities from Section 3.3 (middle row), and performance on the MegaFace Id benchmark (bottom row).

predictions are possible. For ArcFace, the predicted linear relationship between intra-class distance and the margin appears to hold until the point where the model achieves peak performance.

The prevailing explanation in the literature for the value of introducing a margin on the correct class logit has been that doing so *encourages smaller intra-class distances for the corresponding trained model on the training set*. This explanation seems to imply that increasing the margin further should always improve model performance, and that improved training set intra-class distance implies improved test set intra-class distance. However, we observe that the SphereFace and AmpFace plots in Fig. 4 seem to refute that observation. When we increase SphereFace's margin past the optimal model performance value of $m_1 = 1.35$, the intra-class angle keeps decreasing as before but model performance drops significantly. More dramatically, decreasing AmpFace to the optimal value $m_0 = 0.35$ greatly improves model performance without a significant change in final intra-class distance. This indicates that though these multiplicative margins correlate more or less with intra-class distance, they contribute to generalization at least in part for different reasons.

In Fig. 5, we display the curves theoretically derived in Section 3.4 for the loss, as a function of the correct class angle, for each of the empirically verified best margin values for m_0, m_2, and m_3 (see Appendix B for details). We also plot as a dashed blue line the inter-class expected angle given by Proposition 1 ($M_{\text{inter}} = 78.64°$), and as a dashed red line the corresponding half-angle $M_{\text{inter}}/2 = 39.32°$. Notice that this half-angle coincides very closely with the optimization termination point that defines our M_{intra}. The same half-angle relationship holds approximately between the observed train intra-class and inter-class distance for the model checkpoints trained with optimal margin settings $m_0 = 0.35$, $m_2 = 0.6$, $m_3 = 0.5$ identified through a careful linear margin sweep with step size 0.05. Our hypothesis also indicates an optimal value of $m_1 = 1.85$, which lies out of the parameter range where our SphereFace training converges. Recent work [14] however enables model convergence for larger values of m_1 by flattening the loss function when $m_1\theta > \pi$. This shows that SphereFace can achieve maximum performance matching that of ArcFace

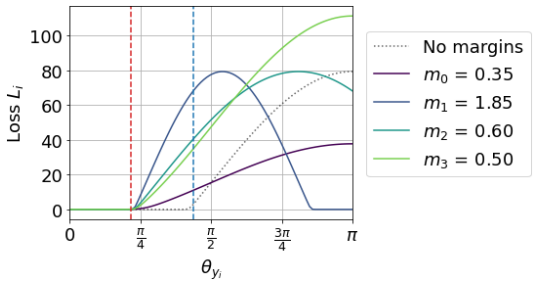


Figure 5: Predicted loss, as a function of the correct class angle, for the margins that achieve best model performance. The dashed red vertical line marks *half* the predicted inter-class distance ($M_{\text{inter}}/2 = 78.64°/2 = 39.32°$) for $C = 85,742$ in the $(512-1)$-unit sphere, which is very close to the optimization termination point and predicted intra-class distance for AmpFace, ArcFace, CosFace, as well as a recent regularized formulation of SphereFace [14].

and CosFace for $m_1 = 1.9$ (Table 8 of [14]), which is nearly identical to the value predicted here.

We therefore conjecture that the optimal margin value reliably produces a loss function with optimization termination point at half the mean inter-class angle. This angle can be understood as the decision boundary between two prototypes after which a higher margin would result in train classification error. However, two spherical caps in high dimensions have a small intersection area even for larger angles [11]. A clear explanation thus defies our current understanding, but if true in general our conjecture is valuable for two reasons. First, it predicts the optimal value for each of the margin parameters, obviating the need for a laborious and expensive grid search across multiple benchmarks. Second, it may shed light on the reason that margins are helpful for optimization and generalization.

4.4. Reg_{ss} vs. WC-ReLU

Here, we empirically compare our two proposed regularization strategies with UniformFace [4] on the MegaFace Id benchmark with one million distractors. Table 2 shows that models that perform well without these extra regularizers still perform equally well after the addition of either strategy. When m_0 is used as the margin, Fig. 6 shows that both strategies are effective at avoiding collapse while UniformFace is much less effective. For extreme m_0 values, WC-ReLU still performs well whereas Reg_{ss} fails. Reg_{ss} may fail if prototypes collapse in low dimensions in a more complicated symmetrical way rather than towards a single point, though studying a more general degenerate solution space is out of the scope of this work. We argue that in practice, either strategy—and especially WC-ReLU—allows a larger search space for the optimal margin hyperparameter and is beneficial for systematic studies.

	Margin	Plain	WC-ReLU	Reg_{ss}
Softmax	-	88.58	88.95	88.54
SphereFace	$m_1 = 1.35$	96.53	96.89	96.45
ArcFace	$m_2 = 0.50$	98.24	98.30	98.30
CosFace	$m_3 = 0.35$	98.11	98.10	98.06

Table 2: Comparison of MegaFace Id for two proposed regularization strategies. When the original training works well, adding either does not harm the performance.

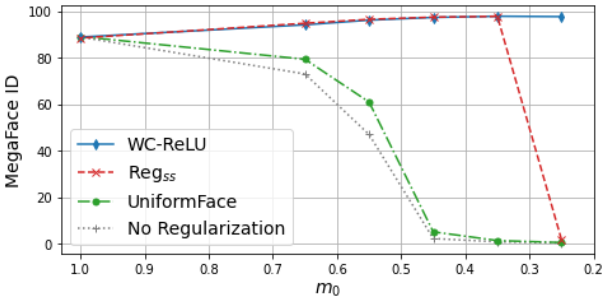


Figure 6: Comparison of the two proposed regularization strategies from Section 3.2 to avoid collapse during training. WC-ReLU outperforms Reg_{ss} for extreme margin values, while UniformFace [4] is less effective than both.

5. Conclusion

Our analysis reveals several new important insights: a training failure mode for certain margin ranges and two regularization strategies to address it; a new multiplicative margin with competitive results when regularized properly; predictions of intra-class and inter-class distances; and clarification of the generally held assumption of smaller intra-class distance leading to better generalization.

Our work further suggests that there is a fixed inter-class angle at which optimization should stop, and that it informs the optimal value of the margins for AmpFace, ArcFace, and CosFace. Namely, that angle seems to be given by the point at which intra-class distance equals half the inter-class distance. This seems to be true both in practice and in our theoretical predictions, and represents a potential direction for future research.

Although further work is needed to fully understand generalization in metric learning applications, the theoretical framework presented here is a first step to a clearer picture of margins in softmax losses and their effects on the spherical manifold during optimization. We also hope the evaluation settings presented here will help researchers benchmark quickly and fairly against prior work.

Acknowledgements: The authors thank Arun Isaac for useful conversations on his algorithm for uniform sampling in spherical cones and for making the code publicly available.

References

[1] Xiang An, Xuhan Zhu, Yang Xiao, Lan Wu, Ming Zhang, Yuan Gao, Bin Qin, Debing Zhang, and Ying Fu. Partial fc: Training 10 million identities on a single machine. *arXiv preprint arXiv:2010.05222*, 2020.

[2] Johann S Brauchart, Alexander B Reznikov, Edward B Saff, Ian H Sloan, Yu Guang Wang, and Robert S Womersley. Random point sets on the sphere—hole radii, covering, and separation. *Experimental Mathematics*, 27(1):62–81, 2018.

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

[4] Yueqi Duan, Jiwen Lu, and Jie Zhou. UniformFace: Learning deep equidistributed representation for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3415–3424, 2019.

[5] 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.

[6] Lanqing He, Zhongdao Wang, Yali Li, and Shengjin Wang. Softmax dissection: Towards understanding intra-and inter-class objective for embedding learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10957–10964, 2020.

[7] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*, 2008.

[8] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. CurricularFace: Adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5901–5910, 2020.

[9] 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.

[10] Yonghyun Kim, Wonpyo Park, Myung-Cheol Roh, and Jongju Shin. GroupFace: Learning latent groups and constructing group-based representations for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5621–5630, 2020.

[11] Yongjae Lee and Woo Chang Kim. Concise formulas for the surface area of the intersection of two hyperspherical caps. *KAIST Technical Report*, 2014.

[12] Hao Liu, Xiangyu Zhu, Zhen Lei, and Stan Z Li. AdaptiveFace: Adaptive margin and sampling for face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11947–11956, 2019.

[13] Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. Learning with hyperspherical uniformity. In *AISTATS*, 2021.

[14] Weiyang Liu, Yandong Wen, Bhiksha Raj, Rita Singh, and Adrian Weller. SphereFace revived: Unifying hyperspherical face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[15] 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.

[16] 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.

[17] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. MagFace: A universal representation for face recognition and quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14220–14229, 2021.

[18] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017.

[19] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[20] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020.

[21] Rajeev Ranjan, Carlos D. Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification, 2017.

[22] 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.

[23] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[24] 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.

[25] Murugesan Venkatapathi et al. An O(n) algorithm for generating uniform random vectors in n-dimensional cones. *arXiv preprint arXiv:2101.00936*, 2021.

[26] 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.

[27] 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.

[28] 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.

[29] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534. IEEE, 2011.

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

[31] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019.

[32] Tianyue Zheng and Weihong Deng. Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5, 2018.

[33] 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.

[34] Yutong Zheng, Dipan K. Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.