CVPR 2020

1. Unified SoftMax, Margin-based, Multi-N-pair
2. self-paced weighting
3. margin

Circle Loss: A Unified Perspective of Pair Similarity Optimization

Yifan Sun[1]*, Changmao Cheng[1]*, Yuhan Zhang[2]*, Chi Zhang[1], Liang Zheng[3], Zhongdao Wang[4], Yichen Wei[1]†
[1]Megvii Inc. [2]Beihang University [3]Australian National University [4]Tsinghua University
{peter, chengchangmao, zhangchi, weiyicheng}@megvii.com

Abstract

This paper provides a pair similarity optimization viewpoint on deep feature learning, aiming to maximize the within-class similarity s_p and minimize the between-class similarity s_n. We find a majority of loss functions, including the triplet loss and the softmax plus cross-entropy loss, embed s_n and s_p into similarity pairs and seek to reduce $(s_n - s_p)$. Such an optimization manner is inflexible, because the penalty strength on every single similarity score is restricted to be equal. Our intuition is that if a similarity score deviates far from the optimum, it should be emphasized. To this end, we simply re-weight each similarity to highlight the less-optimized similarity scores. It results in a Circle loss, which is named due to its circular decision boundary. The Circle loss has a unified formula for two elemental deep feature learning approaches, i.e., *learning with class-level labels and pair-wise labels. Analytically, we show that the Circle loss offers a more flexible optimization approach towards a more definite convergence target, compared with the loss functions optimizing $(s_n - s_p)$. Experimentally, we demonstrate the superiority of the Circle loss on a variety of deep feature learning tasks. On face recognition, person re-identification, as well as several fine-grained image retrieval datasets, the achieved performance is on par with the state of the art.*

1. Introduction

This paper holds a similarity optimization view towards two elemental deep feature learning approaches, *i.e.*, learning from data with class-level labels and from data with pair-wise labels. The former employs a classification loss function (*e.g.*, Softmax plus cross-entropy loss [25, 16, 36]) to optimize the similarity between samples and weight vectors. The latter leverages a metric loss function (*e.g.*, triplet loss [9, 22]) to optimize the similarity between samples. In our interpretation, there is no intrinsic difference between these two learning approaches. They both seek to minimize between-class similarity s_n, as well as to maximize within-class similarity s_p.

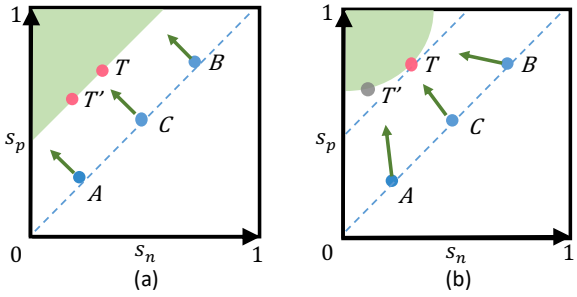


Figure 1: Comparison between the popular optimization manner of reducing $(s_n - s_p)$ and the proposed optimization manner of reducing $(\alpha_n s_n - \alpha_p s_p)$. (a) Reducing $(s_n - s_p)$ is prone to inflexible optimization (A, B and C all have equal gradients with respect to s_n and s_p), as well as ambiguous convergence status (both T and T' on the decision boundary are acceptable). (b) With $(\alpha_n s_n - \alpha_p s_p)$, the Circle loss dynamically adjusts its gradients on s_p and s_n, and thus benefits from flexible optimization process. For A, it emphasizes on increasing s_p; for B, it emphasizes on reducing s_n. Moreover, it favors a specified point T on the circular decision boundary for convergence, setting up a definite convergence target.

From this viewpoint, we find that many popular loss functions (*e.g.*, triplet loss [9, 22], Softmax loss and its variants [25, 16, 36, 29, 32, 2]) share a similar optimization pattern. They all embed s_n and s_p into similarity pairs and seek to reduce $(s_n - s_p)$. In $(s_n - s_p)$, increasing s_p is equivalent to reducing s_n. We argue that this symmetric optimization manner is prone to the following two problems.

• **Lack of flexibility for optimization.** The penalty strength on s_n and s_p is restricted to be equal. Given the specified loss functions, the gradients with respect to s_n and s_p are of same amplitudes (as detailed in Section 2). In some corner cases, *e.g.*, s_p is small and s_n already approaches 0 ("A" in Fig. 1 (a)), it keeps on penalizing s_n with large gradient. It is inefficient and irrational.

• **Ambiguous convergence status.** Optimizing $(s_n - s_p)$

*Equal contribution.
†Corresponding author.

usually leads to a decision boundary of $s_p - s_n = m$ (m is the margin). This decision boundary allows ambiguity (*e.g.*, "T" and "T'" in Fig. 1 (a)) for convergence. For example, T has $\{s_n, s_p\} = \{0.2, 0.5\}$ and T' has $\{s'_n, s'_p\} = \{0.4, 0.7\}$. They both obtain the margin $m = 0.3$. However, comparing them against each other, we find the gap between s'_n and s_p is only 0.1. Consequently, the ambiguous convergence compromises the separability of the feature space.

With these insights, we reach an intuition that different similarity scores should have different penalty strength. If a similarity score deviates far from the optimum, it should receive strong penalty. Otherwise, if a similarity score already approaches the optimum, it should be optimized mildly. To this end, we first generalize $(s_n - s_p)$ into $(\alpha_n s_n - \alpha_p s_p)$, where α_n and α_p are independent weighting factors, allowing s_n and s_p to learn at different paces. We then implement α_n and α_p as linear functions w.r.t. s_n and s_p respectively, to make the learning pace adaptive to the optimization status: The farther a similarity score deviates from the optimum, the larger the weighting factor will be. Such optimization results in the decision boundary $\alpha_n s_n - \alpha_p s_p = m$, yielding a circle shape in the (s_n, s_p) space, so we name the proposed loss function *Circle loss*.

Being simple, Circle loss intrinsically reshapes the characteristics of the deep feature learning from the following three aspects:

First, a unified loss function. From the unified similarity pair optimization perspective, we propose a unified loss function for two elemental learning approaches, *learning with class-level labels and with pair-wise labels.*

Second, flexible optimization. During training, the gradient back-propagated to s_n (s_p) will be amplified by α_n (α_p). Those less-optimized similarity scores will have larger weighting factors and consequentially get larger gradient. As shown in Fig. 1 (b), the optimization on A, B and C are different to each other.

Third, definite convergence status. On the circular decision boundary, Circle loss favors a specified convergence status ("T" in Fig. 1 (b)), as to be demonstrated in Section 3.3. Correspondingly, it sets up a definite optimization target and benefits the separability.

The main contributions of this paper are summarized as follows:

- We propose Circle loss, a simple loss function for deep feature learning. By re-weighting each similarity score under supervision, Circle loss benefits the deep feature learning with flexible optimization and definite convergence target.
- We present Circle loss with compatibility to both class-level labels and pair-wise labels. Circle loss degenerates to triplet loss or Softmax loss with slight modifications.
- We conduct extensive experiment on a variety of deep feature learning tasks, *e.g.* face recognition, person re-identification, car image retrieval and so on. On all these tasks, we demonstrate the superiority of Circle loss with performance on par with the state of the art.

2. A Unified Perspective

Deep feature learning aims to maximize the within-class similarity s_p, as well as to minimize the between-class similarity s_n. Under the cosine similarity metric, for example, we expect $s_p \to 1$ and $s_n \to 0$.

To this end, **learning with class-level labels** and **learning with pair-wise labels** are two paradigms of approaches and are usually considered separately. Given class-level labels, the first one basically learns to classify each training sample to its target class with a classification loss, *e.g.* L2-Softmax [21], Large-margin Softmax [15], Angular Softmax [16], NormFace [30], AM-Softmax [29], CosFace [32], ArcFace [2]. In contrast, given pair-wise labels, the second one directly learns pair-wise similarity in the feature space in an explicit manner, *e.g.*, constrastive loss [5, 1], triplet loss [9, 22], Lifted-Structure loss [19], N-pair loss [24], Histogram loss [27], Angular loss [33], Margin based loss [38], Multi-Similarity loss [34] and so on.

This paper views both learning approaches from a unified perspective. Given a single sample x in the feature space, let us assume that there are K within-class similarity scores and L between-class similarity scores associated with x. We denote these similarity scores as $\{s_p^i\}\,(i = 1, 2, \cdots, K)$ and $\{s_n^j\}\,(j = 1, 2, \cdots, L)$, respectively.

To minimize each s_n^j as well as to maximize s_p^i, $(\forall i \in \{1, 2, \cdots, K\}, \forall j \in \{1, 2, \cdots, L\})$, we propose a unified loss function by:

$$\begin{aligned}\mathcal{L}_{uni} &= \log\Big[1 + \sum_{i=1}^{K}\sum_{j=1}^{L}\exp(\gamma(s_n^j - s_p^i + m))\Big] \\ &= \log\Big[1 + \sum_{j=1}^{L}\exp(\gamma(s_n^j + m))\sum_{i=1}^{K}\exp(\gamma(-s_p^i))\Big],\end{aligned} \tag{1}$$

in which γ is a scale factor and m is a margin for better similarity separation.

Eq. 1 is intuitive. It iterates through every similarity pair to reduce $(s_n^j - s_p^i)$. We note that it degenerates to triplet loss or classification loss, through slight modifications.

Given class-level labels, we calculate the similarity scores between x and weight vectors w_i $(i = 1, 2, \cdots, N)$ (N is the number of training classes) in the classification layer. Specifically, we get $(N - 1)$ between-class similarity scores by: $s_n^j = w_j^T x / (\|w_j\|\|x\|)$ (w_j is the j-th non-target weight vector). Additionally, we get a single within-class similarity score (with the superscript omitted) $s_p = w_y^T x / (\|w_y\|\|x\|)$. With these prerequisite, Eq. 1 de-

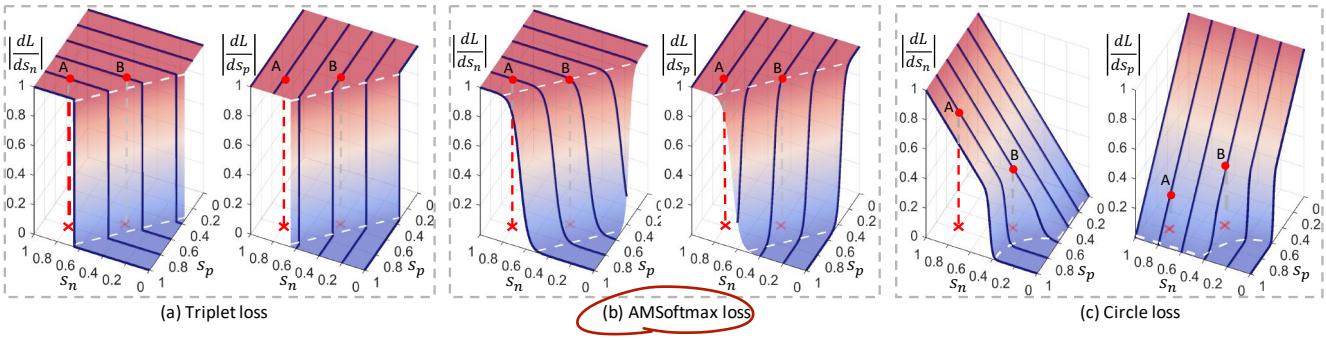


Figure 2: The gradients of the loss functions. (a) Triplet loss. (b) AMSoftmax loss. (c) The proposed Circle loss. Both triplet loss and AMSoftmax loss present lack of flexibility for optimization. The gradients with respect to s_p (left) and s_n (right) are restricted to equal and undergo a sudden decrease upon convergence (the similarity pair B). For example, at A, the within-class similarity score s_p already approaches 1, and still incurs large gradient. Moreover, the decision boundaries are parallel to $s_p = s_n$, which allows ambiguous convergence. In contrast, the proposed Circle loss assigns different gradients to the similarity scores, depending on their distances to the optimum. For A (both s_n and s_p are large), Circle loss lays emphasis on optimizing s_n. For B, since s_n significantly decreases, Circle loss reduces its gradient and thus enforces mild penalty. Circle loss has a circular decision boundary, and promotes accurate convergence status.

generates to AM-Softmax [29, 32], an important variant of Softmax loss:

$$\begin{aligned}\mathcal{L}_{am} &= \log\Big[1 + \sum_{j=1}^{N-1} \exp(\gamma(s_n^j + m))\exp(-\gamma s_p)\Big] \\ &= -\log \frac{\exp(\gamma(s_p - m))}{\exp(\gamma(s_p - m)) + \sum_{j=1}^{N-1}\exp(\gamma s_n^j)}.\end{aligned} \tag{2}$$

Moreover, with $m = 0$, Eq. 2 further degenerates to Normface [30]. By replacing the cosine similarity with inner product and setting $\gamma = 1$, it finally degenerates to Softmax loss (*i.e.*, softmax plus cross-entropy loss).

Given pair-wise labels, we calculate the similarity scores between x and the other features in the mini-batch. Specifically, $s_n^j = x_j^T x/(\|x_j\|\|x\|)$ (x_j is the j-th sample in the negative sample set $\mathcal{N}$) and $s_p^i = x_i^T x/(\|x_i\|\|x\|)$ (x_i is the i-th sample in the positive sample set $\mathcal{P}$). Correspondingly, $K = |\mathcal{P}|$, $L = |\mathcal{N}|$. Eq. 1 degenerates to triplet loss with hard mining [22, 8]:

$$\begin{aligned}\mathcal{L}_{tri} &= \lim_{\gamma\to+\infty}\frac{1}{\gamma}\mathcal{L}_{uni} \\ &= \lim_{\gamma\to+\infty}\frac{1}{\gamma}\log\Big[1 + \sum_{i=1}^{K}\sum_{j=1}^{L}\exp(\gamma(s_n^j - s_p^i + m))\Big] \\ &= \max\big[s_n^j - s_p^i\big]_+.\end{aligned} \tag{3}$$

Specifically, we note that in Eq. 3, the "$\sum\exp(\cdot)$" operation is utilized by Lifted-Structure loss [19], N-pair loss [24], Multi-Similarity loss [34] and *etc.*, to conduct "soft" hard mining among samples. Enlarging γ gradually reinforces the mining intensity and when $\gamma \to +\infty$, it results in the canonical hard mining in [22, 8].

Gradient analysis. Eq. 2 and Eq. 3 show triplet loss, Softmax loss and its several variants can be interpreted as specific cases of Eq. 1. In another word, they all optimize $(s_n - s_p)$. Under the toy scenario where there are only a single s_p and s_n, we visualize the gradients of triplet loss and AMSoftmax loss in Fig. 2 (a) and (b), from which we draw the following observations:

- First, before the loss reaches its decision boundary (upon which the gradients vanish), the gradients with respect to both s_p and s_n are the same to each other. The status A has $\{s_n, s_p\} = \{0.8, 0.8\}$, indicating good within-class compactness. However, A still receives large gradient with respect to s_p. It leads to lack of flexibility during optimization.

- Second, the gradients stay (roughly) constant before convergence and undergo a sudden decrease upon convergence. The status B lies closer to the decision boundary and is better optimized, compared with A. However, the loss functions (both triplet loss and AMSoftmax loss) enforce approximately equal penalty on A and B. It is another evidence of inflexibility.

- Third, the decision boundaries (the white dashed lines) are parallel to $s_n - s_p = m$. Any two points (*e.g.*, T and T' in Fig. 1) on this boundary have an equal similarity gap of m, and are thus of equal difficulties to achieve. In another word, loss functions minimizing $(s_n - s_p + m)$ lay no preference on T or T' for convergence, and are prone to ambiguous convergence. Experimental evidence of this problem is to be accessed in Section 4.6.

These problems originate from the optimization manner of minimizing $(s_n - s_p)$, in which reducing s_n is equivalent

to increasing s_p. In the following Section 3, we will transfer such an optimization manner into a more general one to facilitate higher flexibility.

3. A New Loss Function

3.1. Self-paced Weighting

We consider to enhance the optimization flexibility by allowing each similarity score to learn at its own pace, depending on its current optimization status. We first neglect the margin item m in Eq. 1 and transfer the unified loss function (Eq. 1) into the proposed Circle loss by:

$$\begin{aligned}\mathcal{L}_{circle} &= \log\Big[1+\sum_{i=1}^{K}\sum_{j=1}^{L}\exp\left(\gamma(\alpha_n^j s_n^j - \alpha_p^i s_p^i)\right)\Big] \\ &= \log\Big[1+\sum_{j=1}^{L}\exp(\gamma\alpha_n^j s_n^j)\sum_{i=1}^{K}\exp(-\gamma\alpha_p^i s_p^i),\Big]\end{aligned} \tag{4}$$

in which α_n^j and α_p^i are non-negative weighting factors.

Eq. 4 is derived from Eq. 1 by generalizing $(s_n^j - s_p^i)$ into $(\alpha_n^j s_n^j - \alpha_p^i s_p^i)$ (with margin m neglected). During training, the gradient with respect to $(\alpha_n^j s_n^j - \alpha_p^i s_p^i)$ is to be multiplied with α_n^j (α_p^i) when back-propagated to s_n^j (s_p^i). Let us assume that the optimum for s_p^i is O_p, and the optimum for s_n^j is O_n ($O_n < O_p$). When a similarity score deviates far from its optimum (*i.e.*, O_n for s_n^j and O_p for s_p^i), it should get a large weighting factor so as to get effective update with large gradient. To this end, we define α_p^i and α_n^j in a self-paced manner:

$$\begin{cases}\alpha_p^i = [O_p - s_p^i]_+, \\ \alpha_n^j = [s_n^j - O_n]_+,\end{cases} \tag{5}$$

in which $[\cdot]_+$ is the "cut-off at zero" operation to ensure α_p^i and α_n^j are non-negative.

Re-scaling the cosine similarity under supervision is a common practice in modern classification losses [21, 30, 29, 32, 39, 40]. Conventionally, all the similarity score share an equal scale factor γ. The non-normalized weighting operation in Circle loss can be also interpreted as a specific scaling operation. Different from the other loss functions, Circle loss re-weights (re-scales) each similarity score independently and thus allows different learning paces. We empirically show that Circle loss is robust to various γ settings in Section 4.5.

Discussions. We notice another difference beyond the scaling strategy. The output of softmax function in a classification loss is conventionally interpreted as the probability of a sample belonging to a certain class. Since the probabilities are based on comparing each similarity score against all the similarity scores, equal re-scaling is prerequisite for fair comparison. Circle loss abandons such an probability-related interpretation and holds a similarity pair optimization perspective, instead. Correspondingly, it gets rid of the constraint of equal re-scaling and allows more flexible optimization.

3.2. Within-class and Between-class Margins

In loss functions optimizing $(s_n - s_p)$, adding a margin m reinforces the optimization [15, 16, 29, 32]. Since s_n and $-s_p$ are in symmetric positions, a positive margin on s_n is equivalent to a negative margin on s_p. It thus only requires a single margin m. In Circle loss, s_n and s_p are in asymmetric position. Naturally, it requires respective margins for s_n and s_p, which is formulated by:

$$\mathcal{L}_{circle} = \log\Big[1+\sum_{j=1}^{L}\exp(\gamma\alpha_n^j(s_n^j - \Delta_n))\sum_{i=1}^{K}\exp(-\gamma\alpha_p^i(s_p^i - \Delta_p))\Big] \tag{6}$$

in which Δ_n and Δ_p are the between-class and within-class margins, respectively.

Basically, Circle loss in Eq. 6 expects $s_p^i > \Delta_p$ and $s_n^j < \Delta_n$. We further analyze the settings of Δ_n and Δ_p by deriving the decision boundary. For simplicity, we consider the case of binary classification, in which the decision boundary is achieved at $\alpha_n(s_n - \Delta_n) - \alpha_p(s_p - \Delta_p) = 0$. With Eq. 5 and Eq. 6, the decision boundary is achieved as:

$$(s_n - \frac{O_n + \Delta_n}{2})^2 + (s_p - \frac{O_p + \Delta_p}{2})^2 = C \tag{7}$$

in which $C = \left((O_n - \Delta_n)^2 + (O_p - \Delta_p)^2\right)/4$.

Eq. 7 shows that the decision boundary is the arc of a circle, as shown in Fig. 1 (b). The center of the circle is at $s_n = (O_n + \Delta_n)/2, s_p = (O_p + \Delta_p)/2$, and its radius equals $\sqrt{C}$.

There are five hyper-parameters for Circle loss, *i.e.*, O_p, O_n in Eq. 5 and γ, Δ_p, Δ_n in Eq. 6. We reduce the hyper-parameters by setting $O_p = 1+m$, $O_n = -m$, $\Delta_p = 1-m$, and $\Delta_n = m$. Consequently, the decision boundary in Eq. 7 is reduced to:

$$(s_n - 0)^2 + (s_p - 1)^2 = 2m^2. \tag{8}$$

With the decision boundary defined in Eq. 8, we have another intuitive interpretation of Circle loss. It aims to optimize $s_p \to 1$ and $s_n \to 0$. The parameter m controls the radius of the decision boundary and can be viewed as a relaxation factor. In another word, Circle loss expects $s_p^i > 1 - m$ and $s_n^j < m$.

Hence there are only two hyper-parameters, *i.e.*, the scale factor γ and the relaxation margin m. We will experimentally analyze the impacts of m and γ in Section 4.5.

3.3. The Advantages of Circle Loss

The gradients of Circle loss with respect to s_n^j and s_p^i are derived as follows:

$$\frac{\partial \mathcal{L}_{circle}}{\partial s_n^j} = Z\frac{\exp\left(\gamma((s_n^j)^2 - m^2)\right)}{\sum_{l=1}^{L}\exp\left(\gamma((s_n^l)^2 - m^2)\right)}\gamma(s_n^j + m), \tag{9}$$

and

$$\frac{\partial \mathcal{L}_{circle}}{\partial s_p^i} = Z \frac{\exp\left(\gamma(m^2 - (s_p^i - 1)^2)\right)}{\sum_{k=1}^{K} \exp\left(\gamma(m^2 - (s_p^k - 1)^2)\right)} \gamma(s_p^i - 1 - m), \quad (10)$$

in both of which $Z = 1 - \exp(-\mathcal{L}_{circle})$.

Under the toy scenario of binary classification (or only a single s_n and s_p), we visualize the gradients under different settings of m in Fig. 2 (c), from which we draw the following three observations:

• *Balanced optimization on s_n and s_p.* We recall that the loss functions minimizing $(s_n - s_p)$ always have equal gradients on s_p and s_n and is inflexible. In contrast, Circle loss presents dynamic penalty strength. Among a specified similarity pair $\{s_n, s_p\}$, if s_p is better optimized in comparison to s_n (*e.g.*, $A = \{0.8, 0.8\}$ in Fig. 2 (c)), Circle loss assigns larger gradient to s_n (and vice versa), so as to decrease s_n with higher superiority. The experimental evidence of balanced optimization is to be accessed in Section 4.6.

• *Gradually-attenuated gradients.* At the start of training, the similarity scores deviate far from the optimum and gains large gradient (*e.g.*, "A" in Fig. 2 (c)). As the training gradually approaches the convergence, the gradients on the similarity scores correspondingly decays (*e.g.*, "B" in Fig. 2 (c)), elaborating mild optimization. Experimental result in Section 4.5 shows that the learning effect is robust to various settings of γ (in Eq. 6), which we attribute to the automatically-attenuated gradients.

• *A (more) definite convergence target.* Circle loss has a circular decision boundary and favors T rather than T' (Fig. 1) for convergence. It is because T has the smallest gap between s_p and s_n, compared with all the other points on the decision boundary. In another word, T' has a larger gap between s_p and s_n and is inherently more difficult to maintain. In contrast, losses that minimize $(s_n - s_p)$ have a homogeneous decision boundary, that is, every point on the decision boundary is of the same difficulty to reach. Experimentally, we observe that Circle loss leads to a more concentrate similarity distribution after convergence, as to be detailed in Section 4.6 and Fig. 5.

4. Experiment

We comprehensively evaluate the effectiveness of Circle loss under two elemental learning approaches, *i.e.*, learning with class-level labels and learning with pair-wise labels. For the former approach, we evaluate our method on face recognition (Section 4.2) and person re-identification (Section 4.3) tasks. For the latter approach, we use the fine-grained image retrieval datasets (Section 4.4), which are relatively small and encourage learning with pair-wise labels. We show that Circle loss is competent under both settings. Section 4.5 analyzes the impact of the two hyper-parameters, *i.e.*, the scale factor γ in Eq. 6 and the relaxation factor m in Eq. 8. We show that Circle loss is robust under reasonable settings. Finally, Section 4.6 experimentally confirms the characteristics of Circle loss.

4.1. Settings

Face recognition. We use the popular dataset MS-Celeb-1M [4] for training. The native MS-Celeb-1M data is noisy and has a long-tailed data distribution. We clean the dirty samples and exclude the tail identities ($\leq$ 3 images per identity). It results in $3.6M$ images and $79.9K$ identities. For evaluation, we adopt MegaFace Challenge 1 (MF1) [12], IJB-C [17], LFW [10], YTF [37] and CFP-FP [23] datasets and the official evaluation protocols. We also polish the probe set and 1M distractors on MF1 for more reliable evaluation, following [2]. For data preprocessing, we resize the aligned face images to 112×112 and linearly normalize the pixel values of RGB images to $[-1, 1]$ [36, 15, 32]. We only augment the training samples by random horizontal flip. We choose the popular residual networks [6] as our backbones. All the models are trained with 182k iterations. The learning rate is started with 0.1 and reduced by $10\times$ at 50%, 70% and 90% of total iterations respectively. The default hyper-parameters of our method are $\gamma = 256$ and $m = 0.25$ if not specified. For all the model inference, we extract the 512-D feature embeddings and use cosine distance as metric.

Person re-identification. Person re-identification (re-ID) aims to spot the appearance of a same person in different observations. We evaluate our method on two popular datasets, *i.e.*, Market-1501 [41] and MSMT17 [35]. Market-1501 contains 1,501 identities, 12,936 training images and 19,732 gallery images captured with 6 cameras. MSMT17 contains 4,101 identities, 126,411 images captured with 15 cameras and presents long-tailed sample distribution. We adopt two network structures, *i.e.* a global feature learning model backboned on ResNet50 and a part-feature model named MGN [31]. We use MGN with consideration of its competitive performance and relatively concise structure. The original MGN uses a Sofmax loss on each part feature branch for training. Our implementation concatenates all the part features into a single feature vector for simplicity. For Circle loss, we set $\gamma = 256$ and $m = 0.25$.

Fine-grained image retrieval. We use three datasets for evaluation on fine-grained image retrieval, *i.e.* CUB-200-2011 [28], Cars196 [14] and Stanford Online Products [19]. CARS-196 contains $16,183$ images which belongs to 196 class of cars. The first 98 classes are used for training and the last 98 classes are used for testing. CUB-200-2010 has 200 different class of birds. We use the first 100 class with $5,864$ images for training and the last 100 class with $5,924$ images for testing. SOP is a large dataset consists of $120,053$ images belonging to $22,634$ classes of online products. The training set contains $11,318$ class in-

Table 1: Identification rank-1 accuracy (%) on MFC1 dataset with different backbones and loss functions.

Loss function	MFC1 [12] rank-1		
	ResNet34	ResNet50	ResNet100
Softmax	92.36	93.91	95.04
NormFace [30]	92.62	94.12	95.27
AM-Softmax [29, 32]	97.54	97.86	98.31
ArcFace [2]	97.68	98.03	98.36
CircleLoss (ours)	**97.81**	**98.17**	**98.50**

Table 2: Face verification accuracy (%) on LFW, YTF and CFP-FP with ResNet34 backbone.

Loss function	LFW [10]	YTF [37]	CFP-FP [23]
Softmax	99.18	96.19	95.01
NormFace [30]	99.25	96.03	95.34
AM-Softmax [29, 32]	99.63	96.31	95.78
ArcFace [2]	99.68	96.34	95.84
CircleLoss(ours)	**99.73**	**96.38**	**96.02**

Table 3: Comparison of true accept rates (%) on the IJB-C 1:1 verification task.

Loss function	IJB-C [17] (TAR@FAR)		
	1e-3	1e-4	1e-5
ResNet34, AM-Softmax [29, 32]	95.87	92.14	81.86
ResNet34, CircleLoss(ours)	**96.04**	**93.44**	**86.78**
ResNet100, AM-Softmax [29, 32]	95.93	93.19	88.87
ResNet100, CircleLoss(ours)	**96.29**	**93.95**	**89.60**

cludes $59,551$ images and the rest $11,316$ class includes $60,499$ images are for testing. The experimental setup follows [19]. We use BN-Inception [11] as the backbone to learn 512-D embeddings. We adopt P-K sampling trategy [8] to construct mini-batch with $P = 16$ and $K = 5$. For Circle loss, we set $\gamma = 80$ and $m = 0.4$.

4.2. Face Recognition

For face recognition task, we compare Circle loss against several popular classification loss functions, *i.e.*, vanilla Softmax, NormFace [30], AM-Softmax [29] (or CosFace [32]), ArcFace [2]. Following the original papers [29, 2], we set $\gamma = 64, m = 0.35$ for AM-Softmax and $\gamma = 64, m = 0.5$ for ArcFace.

We report the rank-1 accuracy on MegaFace Challenge 1 dataset (MFC1) in Table 1. On all the three backbones, Circle loss marginally outperforms the counterparts. For example, with ResNet34 as the backbone, Circle loss surpasses the most competitive one (ArcFace) by +0.13%. With ResNet100 as the backbone, while ArcFace achieves a high rank-1 accuracy of 98.36%, Circle loss still outperforms it by +0.14%.

Table 2 summarizes face verification results on LFW [10], YTF [37] and CFP-FP [23]. We note that performance on these datasets is already near saturation. Specifically, ArcFace is higher than AM-Softmax by +0.05%, +0.03%, +0.07% on three datasets, respectively. Circle loss remains the best one, surpassing ArcFace by +0.05%, +0.06% and +0.18%, respectively.

We further compare Circle loss with AM-Softmax on IJB-C 1:1 verification task in Table 3. Our implementation of Arcface is unstable on this dataset and achieves abnormally low performance, so we did not compare Circle loss against Arcface. With ResNet34 as the backbone, Circle loss significantly surpasses AM-Softmax by +1.30% and +4.92% on "TAR@FAR=1e-4" and "TAR@FAR=1e-5", respectively. With ResNet100 as the backbone, Circle loss still maintains considerable superiority.

Table 4: Evaluation of Circle loss on re-ID task. We report R-1 accuracy (%) and mAP (%).

Method	Market-1501		MSMT17	
	R-1	mAP	R-1	mAP
PCB [26] (Softmax)	93.8	81.6	68.2	40.4
MGN [31] (Softmax+Triplet)	95.7	86.9	-	-
JDGL [42]	94.8	86.0	**77.2**	**52.3**
ResNet50 + AMSoftmax	92.4	83.8	75.6	49.3
ResNet50 + CircleLoss(ours)	94.2	84.9	76.3	50.2
MGN + AMSoftmax	95.3	86.6	76.5	51.8
MGN + CircleLoss(ours)	**96.1**	**87.4**	76.9	52.1

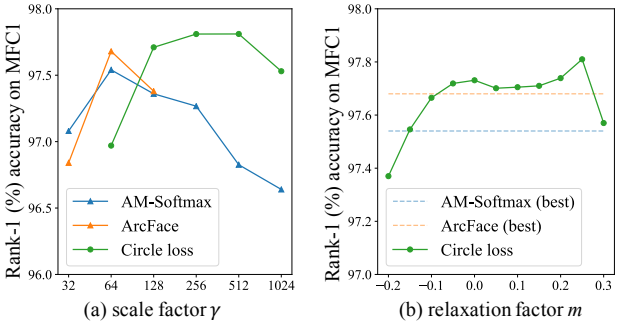
4.3. Person Re-identification

We evaluate Circle loss on re-ID task in Table 4. MGN [31] is one of the state-of-the-art method and is featured for learning multi-granularity part-level features. Originally, it uses both Softmax loss and triplet loss to facilitate a joint optimization. Our implementation of "MGN (ResNet50) + AMSoftmax" and "MGN (ResNet50)+ Circle loss" only use a single loss function for simplicity.

We make three observations from Table 4. First, comparing Circle loss against state of the art, we find that Circle loss achieves competitive re-ID accuracy, with a concise setup (no more auxiliary loss functions). We note that "JDGL" is slightly higher than "MGN + Circle loss" on MSMT17 [35]. JDGL [42] uses generative model to augment the training data, and significantly improves re-ID over long-tailed dataset. Second, comparing "Circle loss" with "AMSoftmax", we observe the superiority of Circle loss, which is consistent with the experimental results on face recognition task. Third, comparing "ResNet50 + Circle loss" against "MGN + Circle loss", we find that part-level

Table 5: Comparison with state of the art on CUB-200-2011, Cars196 and Stanford Online Products. R@K(%) is reported.

Loss function	CUB-200-2011 [28]				Cars196 [14]				Stanford Online Products [19]			
	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8	R@1	R@10	R@10^2	R@10^3
LiftedStruct [19]	43.6	56.6	68.6	79.6	53.0	65.7	76.0	84.3	62.5	80.8	91.9	97.4
HDC [18]	53.6	65.7	77.0	85.6	73.7	83.2	89.5	93.8	69.5	84.4	92.8	97.7
HTL [3]	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7	74.8	88.3	94.8	98.4
ABIER [20]	57.5	71.5	79.8	87.4	82.0	89.0	93.2	96.1	74.2	86.9	94.0	97.8
ABE [13]	60.6	71.5	79.8	87.4	**85.2**	**90.5**	94.0	96.1	76.3	88.4	94.8	98.2
Multi-Simi [34]	65.7	77.0	**86.3**	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0	**98.7**
CircleLoss(ours)	**66.7**	**77.4**	86.2	**91.2**	83.4	89.8	**94.1**	**96.5**	**78.3**	**90.5**	**96.1**	98.6



(a) scale factor γ (b) relaxation factor m

Figure 3: Impact of two hyper-parameters. In (a), Circle loss presents high robustness on various settings of scale factor γ. In (b), Circle loss surpasses the best performance of both AMSoftmax and ArcFace within a large range of relaxation factor m.

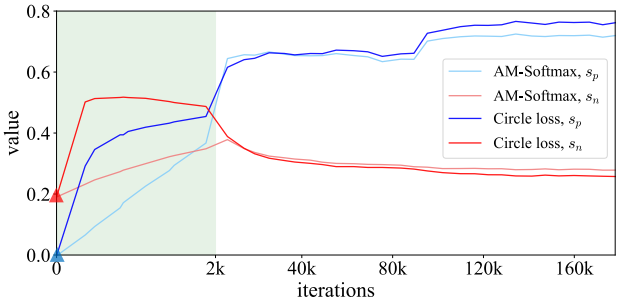


Figure 4: The change of s_p and s_n values during training. We linearly lengthen the curves within the first $2k$ iterations to highlight the initial training process (in the green zone). During the early training stage, Circle loss rapidly increases s_p, because s_p deviates far from the optimum at the initialization and thus attracts higher optimization priority.

features bring incremental improvement to Circle loss. It implies that Circle loss is compatible to the part-model specially designed for re-ID.

4.4. Fine-grained Image Retrieval

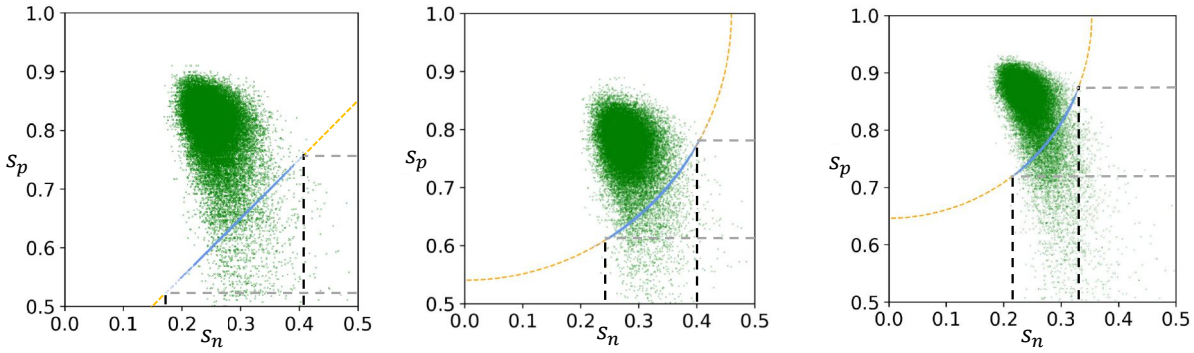
We evaluate the compatibility of Circle loss to pair-wise labeled data on three fine-grained image retrieval datasets, *i.e.*, CUB-200-2011, Cars196, and Standford Online Products. On these datasets, majority methods [19, 18, 3, 20, 13, 34] adopt the encouraged setting of learning with pair-wise labels. We compare Circle loss against these state-of-the-art methods in Table 5. We observe that Circle loss achieves competitive performance, on all of the three datasets. Among the competing methods, LiftedStruct [19] and Multi-Simi [34] are specially designed with elaborate hard mining strategies for learning with pair-wise labels. HDC [18], ABIER [20] and ABE [13] benefit from model ensemble. In contrast, the proposed Circle loss achieves performance on par with the state of the art, without any bells and whistles.

4.5. Impact of the Hyper-parameters

We analyze the impact of two hyper-parameters, *i.e.*, the scale factor γ in Eq. 6 and the relaxation factor m in Eq. 8 on face recognition tasks.

The scale factor γ determines the largest scale of each similarity score. The concept of scale factor is critical in a lot of variants of Softmax loss. We experimentally evaluate its impact on Circle loss and make a comparison with several other loss functions involving scale factors. We vary γ from 32 to 1024 for both AMSoftmax and Circle loss. For ArcFace, we only set γ to 32, 64 and 128, as it becomes unstable with larger γ in our implementation. The results are visualized in Fig. 3. Compared with AM-Softmax and ArcFace, Circle loss exhibits high robustness on γ. The main reason for the robustness of Circle loss on γ is the automatic attenuation of gradients. As the training progresses, the similarity scores approach toward the optimum. Consequentially, the weighting scales along with the gradients automatically decay, maintaining a mild optimization.

The relaxation factor m determines the radius of the circular decision boundary. We vary m from -0.2 to 0.3



(a) AMSoftmax (m=0.35) (b) Circle loss (m=0.325) (c) Circle loss (m=0.25)

Figure 5: Visualization of the similarity distribution after convergence. The blue dots mark the similarity pairs crossing the decision boundary during the whole training process. The green dots mark the similarity pairs after convergence. (a) AMSoftmax seeks to minimize $(s_n - s_p)$. During training, the similarity pairs cross the decision boundary through a wide passage. After convergence, the similarity pairs scatter in a relatively large region in the (s_n, s_p) space. In (b) and (c), Circle loss has a circular decision boundary. The similarity pairs cross the decision boundary through a narrow passage and gather into a relatively concentrated region.

(with 0.05 as the interval) and visualize the results in Fig. 3 (b). It is observed that under all the settings from -0.1 to 0.25, Circle loss surpasses the best performance of Arcface, as well as AMSoftmax, presenting considerable degree of robustness.

4.6. Investigation of the Characteristics

Analysis of the optimization process. To intuitively understand the learning process, we show the change of s_n and s_p during the whole training process in Fig. 4, from which we draw two observations:

First, at the initialization, all the s_n and s_p scores are small. It is because in the high dimensional feature space, randomized features are prone to be far away from each other [40, 7]. Correspondingly, s_p get significantly larger weights (compared with s_n), and the optimization on s_p dominates the training, incurring a fast increase in similarity values in Fig. 4. This phenomenon evidences that Circle loss maintains a flexible and balanced optimization.

Second, at the end of training, Circle loss achieves both better within-class compactness and between-class discrepancy (on the training set), compared with AMSoftmax. Considering the fact that Circle loss achieves higher performance on the testing set, we believe that it indicates better optimization.

Analysis of the convergence. We analyze the convergence status of Circle loss in Fig. 5. We investigate two issues: how do the similarity pairs consisted of s_n and s_p cross the decision boundary during training and how do the similarity pairs distribute in the (s_n, s_p) space after convergence. The results are shown in Fig. 5. In Fig. 5 (a), AMSoftmax loss adopts the optimal setting of $m = 0.35$. In Fig. 5 (b), Circle loss adopts a compromised setting of $m = 0.325$. The decision boundaries of (a) and (b) are tangent to each other, allowing an intuitive comparison. In Fig. 5 (c), Circle loss adopts its optimal setting of $m = 0.25$. Comparing Fig. 5 (b) and (c) against Fig. 5 (a), we find that Circle loss presents a relatively narrower passage on the decision boundary, as well as a more concentrated distribution for convergence (especially when $m = 0.25$). It indicates that Circle loss facilitates more consistent convergence for all the similarity pairs, compared with AMSoftmax loss. This phenomenon confirms that Circle loss has a more definite convergence target, which promotes better separability in the feature space.

5. Conclusion

This paper provides two insights into the optimization process for deep feature learning. First, a majority of loss functions, including the triplet loss and popular classification losses, conduct optimization by embedding the between-class and within-class similarity into similarity pairs. Second, within a similarity pair under supervision, each similarity score favors different penalty strength, depending on its distance to the optimum. These insights result in Circle loss, which allows the similarity scores to learn at different paces. The Circle loss benefits deep feature learning with high flexibility in optimization and a more definite convergence target. It has a unified formula for two elemental learning approaches, *i.e.*, learning with class-level labels and learning with pair-wise labels. On a variety of deep feature learning tasks, *e.g.*, face recognition, person re-identification, and fine-grained image retrieval, the Circle loss achieves performance on par with the state of the art.

References

[1] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:539–546 vol. 1, 2005. 2

[2] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 5, 6

[3] W. Ge. Deep metric learning with hierarchical triplet loss. In *The European Conference on Computer Vision (ECCV)*, September 2018. 7

[4] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, 2016. 5

[5] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[7] L. He, Z. Wang, Y. Li, and S. Wang. Softmax dissection: Towards understanding intra- and inter-clas objective for embedding learning. *CoRR*, abs/1908.01281, 2019. 8

[8] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 3, 6

[9] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015. 1, 2

[10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 5, 6

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6

[12] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. 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. 5, 6

[13] W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon. Attention-based ensemble for deep metric learning. In *The European Conference on Computer Vision (ECCV)*, September 2018. 7

[14] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 5, 7

[15] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 2, 4, 5

[16] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. 1, 2, 4

[17] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165. IEEE, 2018. 5, 6

[18] H. Oh Song, S. Jegelka, V. Rathod, and K. Murphy. Deep metric learning via facility location. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 7

[19] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016. 2, 3, 5, 6, 7

[20] M. Opitz, G. Waltner, H. Possegger, and H. Bischof. Deep metric learning with bier: Boosting independent embeddings robustly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 7

[21] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. 2, 4

[22] F. Schroff, D. Kalenichenko, and J. 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. 1, 2, 3

[23] 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 *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 5, 6

[24] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016. 2, 3

[25] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1891–1898, 2014. 1

[26] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *The European Conference on Computer Vision (ECCV)*, September 2018. 6

[27] E. Ustinova and V. S. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, 2016. 2

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5, 7

[29] F. Wang, J. Cheng, W. Liu, and H. Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018. 1, 2, 3, 4, 6

[30] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049. ACM, 2017. 2, 3, 4, 6

[31] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou. Learning discriminative features with multiple granularities for person re-identification. *2018 ACM Multimedia Conference on Multimedia Conference - MM 18*, 2018. 5, 6

[32] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 4, 5, 6

[33] J. J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. Deep metric learning with angular loss. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2612–2620, 2017. 2

[34] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 2, 3, 7

[35] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person transfer gan to bridge domain gap for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5, 6

[36] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 1, 5

[37] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011. 5, 6

[38] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017. 2

[39] X. Zhang, F. X. Yu, S. Karaman, W. Zhang, and S.-F. Chang. Heated-up softmax embedding. *ArXiv*, abs/1809.04157, 2018. 4

[40] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *CVPR*, 2019. 4, 8

[41] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 5

[42] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz. Joint discriminative and generative learning for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6