Pairwise Similarity Learning is SimPLE

Yandong Wen[1,*], Weiyang Liu[1,2,*], Yao Feng[1], Bhiksha Raj[3,4], Rita Singh[3]
Adrian Weller[2,5], Michael J. Black[1], Bernhard Schölkopf[1]
[1]Max Planck Institute for Intelligent Systems, [2]University of Cambridge, [3]Carnegie Mellon University,
[4]Mohamed bin Zayed University of Artificial Intelligence, [5]The Alan Turing Institute, [*]Equal contribution

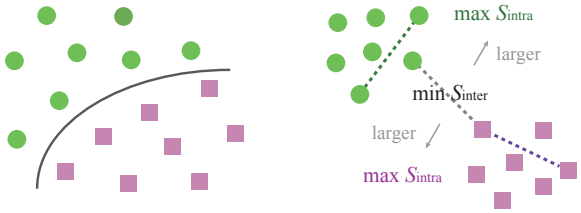
Abstract

In this paper, we focus on a general yet important learning problem, pairwise similarity learning (PSL). PSL subsumes a wide range of important applications, such as open-set face recognition, speaker verification, image retrieval and person re-identification. The goal of PSL is to learn a pairwise similarity function assigning a higher similarity score to positive pairs (i.e., a pair of samples with the same label) than to negative pairs (i.e., a pair of samples with different label). We start by identifying a key desideratum for PSL, and then discuss how existing methods can achieve this desideratum. We then propose a surprisingly simple proxy-free method, called SimPLE, which requires neither feature/proxy normalization nor angular margin and yet is able to generalize well in open-set recognition. We apply the proposed method to three challenging PSL tasks: open-set face recognition, image retrieval and speaker verification. Comprehensive experimental results on large-scale benchmarks show that our method performs significantly better than current state-of-the-art methods. Our project page is available at simple.is.tue.mpg.de.

1. Introduction

How to learn discriminative representations is arguably one of the most fundamental and important problems in computer vision, speech processing and natural language processing. For closed-set classification (*e.g.*, image recognition), it is sufficient to learn class-separable representations as the goal is to infer the label of the input sample. However, for open-set recognition problems such as face recognition [14], speaker verification [1], person re-identification [80] and image retrieval [11], learning class-separable representations is not enough, because the goal becomes learning a similarity function that separates positive and negative pairs well. We study a general problem that is abstracted from these applications – pairwise similarity learning (PSL).

PSL aims to learn a pairwise similarity function such that minimal intra-class similarity is larger than maximal inter-class similarity (or in other words, maximal intra-class distance is smaller than minimal inter-class distance). When this criterion is satisfied, one can easily find a universal threshold that perfectly separates arbitrary positive and negative sample pairs. This property suggests that (i) perfect verification can be achieved and (ii) labels can be fully recovered by simple hierarchical clustering. Compared to classification, PSL presents a more challenging problem of learning large-margin representations, as illustrated by Figure 1.



(a) Standard classification (b) Pairwise similarity learning

Figure 1: Comparison between classification and PSL.

PSL can be viewed as a generalization of deep metric learning (DML). While DML requires the dissimilarity function to be a distance metric that satisfies non-negativity and the triangle inequality, PSL does not necessarily need to follow these criteria. For example, [13, 37, 67, 68] learn a cosine similarity that separates positive and negative pairs.

1.1. Desideratum for Pairwise Similarity Learning

We start by formally describing the desideratum of PSL. PSL seeks to learn a pairwise similarity function $\mathcal{S}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ that is typically symmetric (*i.e.*, $\mathcal{S}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathcal{S}(\boldsymbol{x}_2, \boldsymbol{x}_1)$). The desired pairwise similarity function needs to always satisfy the following inequality: $\mathcal{S}(\boldsymbol{x}_i, \boldsymbol{x}_j) > \mathcal{S}(\boldsymbol{x}_p, \boldsymbol{x}_q)$ where $\boldsymbol{x}_i, \boldsymbol{x}_j$ denote an arbitrary pair of samples with the same label, and $\boldsymbol{x}_p, \boldsymbol{x}_q$ denote an arbitrary pair with different labels. This inequality implies that no negative pair has a larger similarity score than positive pairs. With a labeled dataset, we can further interpret the criterion as

$$\underbrace{\min_{k, i\neq j} \mathcal{S}(\boldsymbol{x}_i^{[k]}, \boldsymbol{x}_j^{[k]})}_{\text{Minimal intra-class similarity score}} > \underbrace{\max_{m\neq n, p, q} \mathcal{S}(\boldsymbol{x}_p^{[m]}, \boldsymbol{x}_q^{[n]})}_{\text{Maximal inter-class similarity score}} \quad (1)$$

where $\boldsymbol{x}_i^{[k]}$ denotes the i-th sample in the k-th class. Normally, there are two ways to parameterize the similarity

function. A naive way is to directly parameterize the similarity function as a neural network $\boldsymbol{\theta}$, resulting in $\mathcal{S}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. However, this parameterization is not scalable for inference, since obtaining all the pairwise similarity scores for N samples requires network inference with complexity $\mathcal{O}(N^2)$. A more sensible way is to, instead, parameterize the similarity score as $\mathcal{S}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), f_{\boldsymbol{\theta}}(\boldsymbol{x}_j))$, where $\mathcal{S}$ is usually a simple and efficient similarity measure (*e.g.*, cosine similarity) and $f_{\boldsymbol{\theta}}$ is a neural feature encoder parameterized by $\boldsymbol{\theta}$. Such a parameterization only requires $\mathcal{O}(N)$ time for network inference and $\mathcal{O}(N^2)$ time for simple pairwise similarity function evaluation. Current PSL methods really boil down to learning a feature encoder that can achieve Eq. 1.

The criterion of Eq. 1 essentially suggests that simple clustering of features leads to perfect classification. Unlike classification problems seeking separable feature representations, PSL aims at large-margin features such that the labels can be recovered by hierarchical clustering. Bearing the desideratum in mind, we first examine how existing PSL methods approach this goal, and then propose a PSL framework that is surprisingly simple yet effective to achieve this.

1.2. Taxonomy of Pairwise Similarity Learning

Towards such a desideratum, there are currently two main types of method: proxy-based PSL (*e.g.*, [13, 37, 67, 69]) and proxy-free PSL (*e.g.*, [17, 57, 63]). Proxy-based PSL utilizes an intermediate parametric sample to serve as a proxy for a group of samples (typically one proxy for one class), which has been shown to benefit convergence and training stability. However, these advantages also come at a price in the sense that it is more difficult for proxy-based PSL to achieve the desideratum. How to achieve Eq. 1 with the presence of proxies is highly nontrivial and usually requires additional design to the loss function [37, 38]. Typical examples of proxy-free PSL include contrastive loss [8, 17] and triplet loss [73], where no proxies are used during training. Although proxy-free PSL can easily use Eq. 1 as the training target, how to construct pairs or triplets becomes especially crucial for convergence and generalization. Hard sample mining matters significantly for performance [77]. Because Eq. 1 is generally intractable to achieve for large training sets, the key difference between proxy-based PSL and proxy-free PSL originates from how they approximate this criterion. Proxy-based PSL achieves Eq. 1 by crafting a relationship between samples and proxies. Proxy-free PSL implements Eq. 1 by sampling a few representative intra-class and inter-class sample pairs, rather than enumerating all the possible pairs. Therefore, how these representative pairs are selected plays a crucial role in determining whether Eq. 1 can be effectively achieved.

Categorization of PSL can also be made from the perspective of how the similarity scores between different pairs interact with each other during optimization [74]. Specifically, if the training involves comparing the similarity scores between different pairs, then we call it triplet-based learning. Typical examples include triplet loss [57, 73] and almost all the margin-based softmax cross-entropy losses [13, 26, 37, 43, 67, 69]. In contrast, if the training directly compares the pair similarity scores to a universal value, then we call it pair-based learning. Examples include contrastive loss [8] and binary cross-entropy [74]. For downstream tasks that focus on comparing pairs of samples, pair-based learning can be preferable since its training objective is more aligned with the testing scenario.

	Proxy-based		Proxy-free	
	Angular	Non-angular	Angular	Non-angular
Triplet	VGGFace [52] Triplet [45] SphereFace [37] NormFace [68] CosFace [67, 69] ArcFace [13] SoftTriple [55] Circle Loss [61] HUG [39]	DeepID [62] DeepFace [63] DeepID2* [60] L-Softmax [38] Center Loss* [75] Proxy NCA [47] Proxy-Anchor [27]	FaceNet [57] Angular Loss [70] Tuplet [81] SupCon [25] Smooth-AP [2] HUG [39]	Triplet Loss [73] N-pair [58] LiftedStruct [59] InfoNCE [51] Log-ratio [28] Ranked List [72] SNR [82]
Pair	SphereFace2 [74]	BCE [24, 30] Center Loss* [75]	AMC-Loss [7] RBM [49]	Siamese [8, 17] DeepID2* [60] Multi-sim [71] SNR [82] **SimPLE**

Table 1: Taxonomy of some representative PSL methods. * indicates that the method has hybrid components.

There are also several similarity functions that are widely adopted in PSL: angular similarity [13, 37, 57, 67, 69, 70], inner product [60, 62] and Euclidean distance [8, 17]. Angular similarity has become a *de facto* choice in open-set recognition, since it can effectively avoid degenerate solutions in triplet-based learning [36, 57] and also help to incorporate angular margin for softmax cross-entropy losses [13, 36, 37, 67, 69, 74]. We summarize a taxonomy for some representative PSL methods in Table 1.

1.3. Motivation and Contribution

Looking into Eq. 1 for PSL, we can observe a few characteristics: (1) similarity is only computed between samples and no proxies are involved; (2) there exists a universal threshold that separates intra-class similarity score and inter-class similarity score. The two observations suggest that pair-based proxy-free learning is best aligned with the desideratum. Despite the perfect alignment between the training target of pair-based proxy-free learning and the desideratum, this category remains largely unexplored and existing methods from it are not particularly competitive. Some natural questions arise: *Why don't pair-based proxy-free PSL methods work as well as expected? Can we realize the full potential for this type of method?* Driven by these questions, our paper studies pair-based proxy-free learning and develops a working algorithm for this approach.

To this end, we first challenge the necessity of a few *de facto* components in state-of-the-art PSL methods, such as angular similarity [37, 40, 57, 68] and angular margin [37], and then propose a surprisingly simple yet effective pair-based proxy-free PSL framework, dubbed *SimPLE*, where neither angular similarity nor margin is needed. Our major contributions can be summarized as follows:

- We rethink the desideratum of pairwise similarity learning, which effectively subsumes many important applications. We identify that pair-based proxy-free learning is most aligned with such a desideratum.
- We challenge a few dominant components in current PSL methods (*e.g.*, angular similarity and margin), and find them unnecessary in the pair-based proxy-free regime.
- We propose SimPLE, a surprisingly simple yet effective pair-based proxy-free learning framework that is designed directly based on the desideratum of PSL.
- Most importantly, we show that SimPLE can easily achieve state-of-the-art performance on open-set face recognition, image retrieval, and speaker verification. We note that this is the first time that a PSL method achieves state-of-the-art performance without the help of angular similarity and margin in open-set face recognition.

2. Rethinking Pairwise Similarity Learning

We start by examining how different types of PSL methods achieve Eq. 1. Since proxy-based PSL models the relationship between samples and proxies, it approximates Eq. 1 through the constraint embedded in the similarity function. Specifically, we consider a two-class scenario. We have samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the first class constitute the minimal intra-class similarity. $\boldsymbol{x}_k$ (class 1) and $\boldsymbol{z}_k$ (class 2) yield the maximal inter-class similarity. Then PSL's desideratum requires us to have $\mathcal{S}(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j) > \mathcal{S}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{z}}_k)$ where we define $\tilde{\boldsymbol{x}} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ for notation convenience. For proxy-based PSL to achieve this inequality, we first consider a triangular inequality for the similarity score function:

$$\mathcal{S}(\boldsymbol{v}_1, \boldsymbol{v}_3) - \mathcal{S}(\boldsymbol{v}_2, \boldsymbol{v}_3) \geq \mathcal{S}(\boldsymbol{v}_1, \boldsymbol{v}_2) \geq \mathcal{S}(\boldsymbol{v}_1, \boldsymbol{v}_3) + \mathcal{S}(\boldsymbol{v}_2, \boldsymbol{v}_3)$$

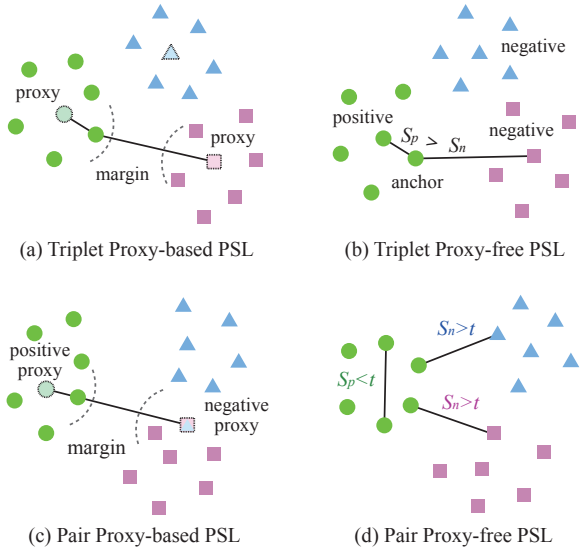
which is also satisfied by the prominent angular similarity, *i.e.*, $\mathcal{S}(\boldsymbol{v}_1, \boldsymbol{v}_2) = 1 - \frac{1}{\pi}\arccos(\frac{\boldsymbol{v}_1^\top \boldsymbol{v}_2}{\|\boldsymbol{v}_1\| \cdot \|\boldsymbol{v}_2\|})$. Then we have

$$\begin{aligned} \mathcal{S}(\tilde{\boldsymbol{x}}_i, \boldsymbol{w}_1) - \mathcal{S}(\tilde{\boldsymbol{x}}_j, \boldsymbol{w}_1) &\geq \mathcal{S}(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j) \\ \mathcal{S}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{z}}_k) &\geq \mathcal{S}(\tilde{\boldsymbol{x}}_k, \boldsymbol{w}_2) + \mathcal{S}(\tilde{\boldsymbol{z}}_k, \boldsymbol{w}_2) \end{aligned} \tag{2}$$

which leads to the following sufficient condition for $\mathcal{S}(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j) > \mathcal{S}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{z}}_k)$ to hold:

$$\underbrace{\mathcal{S}(\tilde{\boldsymbol{x}}_i, \boldsymbol{w}_1)}_{\text{Intra-class similarity}} - \underbrace{\big(\mathcal{S}(\tilde{\boldsymbol{x}}_j, \boldsymbol{w}_1) + \mathcal{S}(\tilde{\boldsymbol{x}}_k, \boldsymbol{w}_2)\big)}_{\text{Margin between similarity scores}} > \underbrace{\mathcal{S}(\tilde{\boldsymbol{x}}_k, \boldsymbol{w}_2)}_{\text{Inter-class similarity}}$$

where $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ denote the proxy for class 1 and 2, respectively. For proxy-based PSL to achieve the desideratum,



(a) Triplet Proxy-based PSL (b) Triplet Proxy-free PSL

(c) Pair Proxy-based PSL (d) Pair Proxy-free PSL

Figure 2: Comparison of different types of PSL.

we have to introduce a margin between intra-class and inter-class similarity score. Without the margin, $\mathcal{S}(\tilde{\boldsymbol{x}}_i, \boldsymbol{w}_1) > \mathcal{S}(\tilde{\boldsymbol{x}}_k, \boldsymbol{w}_2)$ is the criterion for standard classification and only implies separable features. The triangular inequality indicates that with a proper distance metric being the dissimilarity function, it will be easier for proxy-based PSL to achieve the desideratum in Eq. 1. Therefore, margin is actually indispensable for proxy-based PSL.

Then we discuss why angular similarity is widely adopted in PSL. We consider the softmax cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = \log\big(1 + \sum_{i \neq y} \exp(\boldsymbol{w}_i^\top \tilde{\boldsymbol{x}} - \boldsymbol{w}_y^\top \tilde{\boldsymbol{x}})\big) \tag{3}$$

where $\boldsymbol{w}_i$ denotes the i-th class proxy (*i.e.*, last-layer classifier) and $\tilde{\boldsymbol{x}}$ is the feature (y is the label). We have that

$$\lim_{\|\boldsymbol{x}\| \to \infty} \mathcal{L}_{\text{CE}} = \begin{cases} 0 & \text{if } \forall i \neq y,\ \boldsymbol{w}_y^\top \tilde{\boldsymbol{x}} > \boldsymbol{w}_i^\top \tilde{\boldsymbol{x}} \\ +\infty & \text{if } \exists i \neq y,\ \boldsymbol{w}_y^\top \tilde{\boldsymbol{x}} < \boldsymbol{w}_i^\top \tilde{\boldsymbol{x}} \end{cases} \tag{4}$$

which implies that as long as the feature can be classified to the correct class (*i.e.*, features are separable), then the softmax cross-entropy loss can be trivially minimized by increasing the feature norm. In order to eliminate these degenerate solutions, common practice [13, 34, 37, 67–69] resorts to normalizing both proxy weights and features to a fixed length, leading to the popular angular similarity. One may notice an obvious caveat here – both angular margin and angular similarity are especially designed for proxy-based learning. Neither is necessary for proxy-free learning.

Now we discuss how proxy-free learning approximates the desideratum in Eq. 1. Contrastive loss [8, 17] and triplet loss [57, 73] are arguably the most representative proxy-free PSL methods. Since it is computationally intractable to enumerate all the possible sample pairs or triplets, both

contrastive and triplet losses heavily rely on hard sample mining which essentially seeks representative samples to approximate the minimal intra-class and maximal inter-class similarity. Moreover, triplet loss also has similar degenerate solutions as the softmax cross-entropy loss, so it is usually used together with feature normalization [57]. One significant difference between triplet-based learning and pair-based learning is the use of a universal threshold. For example, a triplet loss enforces the similarity between an anchor and a positive sample to be larger than the similarity between the anchor and a negative sample. In contrast, pair-based learning (*e.g.*, contrastive loss and SphereFace2 [74]) compares both positive and negative pairs to a universal threshold, which inherently draws a consistent decision boundary between positive pairs and negative pairs and is more aligned with PSL's desideratum.

We give an intuitive comparison of different types of PSL in Figure 2. The target of pair-based proxy-free PSL is perfectly aligned with the desideratum that minimal intra-class similarity score is larger than maximal inter-class similarity score. Surprisingly, we find that neither angular similarity (*i.e.*, feature/proxy normalization) nor angular margin is necessary. Further, we identify two important aspects that are essential for pair-based proxy-free PSL: (1) pair sampling, which affects how accurately it can approximate the desideratum in Eq. 1; (2) similarity score, which should be consistent across training and testing. In Section 3, we discuss how we use a simple design to address these issues.

3. An Embarrassingly Simple PSL Framework

We aim for SimPLE to be as simple as possible without introducing additional assumptions or priors. We formulate pair-based proxy-free PSL as a pair classification problem, which yields the following naive loss formulation:

$$\mathcal{L}_{\text{n}} = \mathbb{E}_{\{\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2\}\sim\mathcal{D}}\Big\{ y_p \cdot \log\big(1+\exp(-\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)-b)\big) + (1-y_p)\cdot\log\big(1+\exp(\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)+b)\big)\Big\} \tag{5}$$

where $y_p = 1$ if $\tilde{\boldsymbol{x}}_1$ and $\tilde{\boldsymbol{x}}_2$ are from the same class, and $y_p = 0$ otherwise. This is essentially a binary logistic regression without classifiers (*i.e.*, binary cross entropy). The advantage of such a formulation can be better understood from its decision boundary $\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)+b=0$. When $\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)$ is larger than $-b$, then $\tilde{\boldsymbol{x}}_1$ and $\tilde{\boldsymbol{x}}_2$ are predicted to the same class. Otherwise, they are predicted as a negative pair.

Similarity score. Cosine similarity (or angular similarity) has been the *de facto* standard in open-set face recognition [13, 37, 67–69], speaker verification [9, 41, 66] and image retrieval [48]. Despite its popularity, angular similarity introduces an assumption that features are supposed to be discriminative on the unit hypersphere. However, Eq. 1 does not necessitate angular similarity. As long as the similarity score is consistent across training and testing, then we can expect it to generalize well. We start with the simplest case without any assumption – the inner product as the similarity score: $\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2) = \langle\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2\rangle = \|\tilde{\boldsymbol{x}}_1\|\cdot\|\tilde{\boldsymbol{x}}_2\|\cdot\cos(\theta_{\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2})$. However, the sign of the inner product completely depends on the angle between two features. When the angle is smaller than $\frac{\pi}{2}$, then increasing the similarity can trivially become increasing the feature magnitude. When the angle is larger than $\frac{\pi}{2}$, then decreasing the similarity can also trivially become decreasing the feature magnitude. We find it to be a strong assumption to use $\frac{\pi}{2}$ as the sign boundary. Therefore, we remove such an assumption by adding an angular bias:

$$\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2) = \|\tilde{\boldsymbol{x}}_1\|\cdot\|\tilde{\boldsymbol{x}}_2\|\cdot\big(\cos(\theta_{\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2}) - b_\theta\big) \tag{6}$$

where b_θ is learned directly from data and stays constant during inference. How does this angular bias term differ from the bias term in Eq. 5? We write down the decision boundary for the new similarity functions:

$$\underbrace{\|\tilde{\boldsymbol{x}}_1\|\cdot\|\tilde{\boldsymbol{x}}_2\|\cdot\cos(\theta_{\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2})}_{\text{Inner product similarity}} - \underbrace{\|\tilde{\boldsymbol{x}}_1\|\cdot\|\tilde{\boldsymbol{x}}_2\|\cdot b_\theta}_{\text{Data-dependent bias}} + \underbrace{b}_{\text{Constant bias}} = 0$$

which is not equivalent to the decision boundary induced by the inner product similarity. The data-dependent bias serves a different role to the constant bias, and also removes a prescribed assumption in inner product. Our experiments show that removing this assumption is important and leads to consistently better performance. One delicate difference to the angular similarity is that the angular bias is redundant since $\|\tilde{\boldsymbol{x}}_1\|\cdot\|\tilde{\boldsymbol{x}}_2\|\cdot b_\theta$ also becomes some fixed constant and can be trivially merged to b with $\|\tilde{\boldsymbol{x}}_1\| = \|\tilde{\boldsymbol{x}}_2\| = 1$.

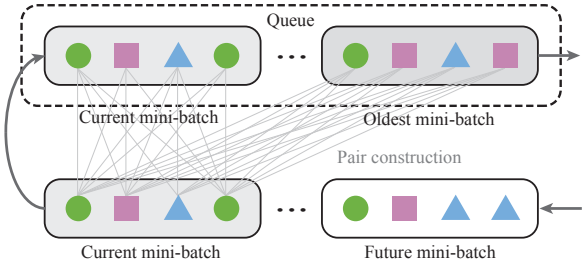
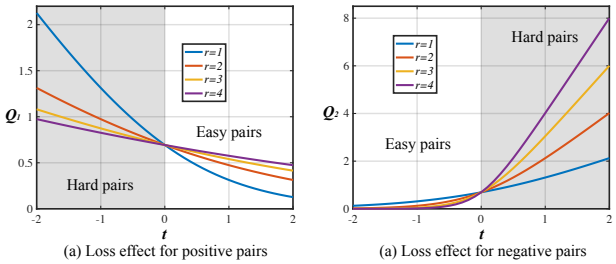


Figure 3: Illustration of SimPLE's pair construction.

Pair sampling. How to construct pairs is arguably one of the most important factors in determining the performance of proxy-free learning [77]. We consider two aspects of pair sampling: pair coverage and pair importance.

Because it is impossible to enumerate all the pair combinations for a large dataset, we seek to enlarge the coverage of pairs. The size of mini-batches also limits the pair coverage. To address this, we maintain a queue of samples encoded by a moving-averaged encoder [18] and then form pairs from samples in the queue. Specifically, we use a first-in-first-out queue where the oldest mini-batch is dequeued as the



(a) Loss effect for positive pairs (a) Loss effect for negative pairs

Figure 4: The effect of r for hard pair mining.

current mini-batch is enqueued. We denote the size of the mini-batch as m and the size of the queue as q. We can form $m \cdot q$ pairs in total. We note that the samples in the queue are encoded by a moving-averaged encoder instead of the original encoder. The moving-averaged encoder is updated by $\boldsymbol{\theta}_q \leftarrow \eta\boldsymbol{\theta}_q + (1-\eta)\boldsymbol{\theta}$ where η is the moving average parameter, $\boldsymbol{\theta}_q$ are the parameters of the moving-averaged encoder and $\boldsymbol{\theta}$ are the parameters of the current encoder that is trained with back-propagation.

With a sufficient number of pairs, we now consider how to weight them based on their importance. We implement the pair reweighting in the loss function. The way we construct pairs will inevitably result in a highly imbalanced number of positive and negative pairs. To address this problem, we first introduce a weighting hyperparameter to balance the importance of positive and negative pairs, yielding

$$\mathcal{L}_{\text{b}} = \mathbb{E}_{\{\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2\}\sim\mathcal{D}}\Big\{\alpha\cdot y_p\cdot\log\left(1+\exp(-\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)+b)\right) + (1-\alpha)\cdot(1-y_p)\cdot\log\left(1+\exp(\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)+b)\right)\Big\} \tag{7}$$

where α is a hyperparameter for balancing positive and negative pairs. Then we consider the final problem of hard pair mining. We note that hard pair mining is highly nontrivial without angular similarity (*i.e.*, feature and proxy normalization). For example, if we multiply the similarity score by a scaling parameter (*i.e.*, simply replace $\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)$ with $r\cdot\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)$ in Eq. 7), this parameter will not have the same effect of hard pair mining as SphereFace2 [74]. This is because the network can trivially learn to decrease the feature magnitude and r will be compensated by the decreased magnitude, whereas features are normalized in [74]. This phenomenon suggests that the effect of hard pair mining within positive and negative pairs tends to cancel out each other in our formulation (without angular similarity).

To address this critical problem, we propose a simple yet novel remedy – perform hard pair mining in a reverse direction for positive and negative pairs. Specifically, we seek a hyperparameter that simultaneously controls the hard pair mining for both positive and negative pairs. As it gets larger, the loss function focuses more on easy pairs within positive pairs, and at the same time, focuses more on hard pairs within negative pairs. The core idea is that as long as the mining directions are reversed for positive and negative pairs, then their effect will no longer cancel out each other. To this end, we multiply $\frac{1}{r}$ to the similarity score in the loss of positive pairs (instead of r), and simultaneously multiply r by the similarity score in the loss of negative pairs. We arrive at the final form of the loss function below:

$$\mathcal{L}_{\text{f}} = \mathbb{E}_{\{\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2\}\sim\mathcal{D}}\Big\{\alpha\cdot y_p\cdot\log\left(1+\exp\left(-\frac{1}{r}(\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)+b)\right)\right) + (1-\alpha)\cdot(1-y_p)\cdot\log\left(1+\exp\left(r(\mathcal{S}(\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_2)+b)\right)\right)\Big\} \tag{8}$$

where r is a hyperparameter that scales the loss curve with respect to the similarity score. Specifically, larger r corresponds to more importance on easy positive pairs and hard negative pairs. We define $Q_1(t) = \log(1+\exp(-t/r))$ and $Q_2(t) = \log(1+\exp(r\cdot t))$, and then plot their curves to illustrate how they achieve hard pair mining of reverse directions. For the function $Q_1(t)$, the loss focuses more on easy pairs as r gets larger. For the function Q_2, the loss focuses more on hard samples as r gets larger.

Simplicity and significance of SimPLE. With similarity score, pair coverage and pair importance taken into account, we end up with a surprisingly simple formulation in Eq. 8 which only requires simple modifications from standard binary cross-entropy. Most importantly, SimPLE completely drops the dependency on angular similarity and margin while still achieving state-of-the-art performance on almost all open-set recognition problems. We believe this method is significant since it opens up new possibilities for PSL and also demonstrates that angular similarity and margin are no longer requisite to achieve state-of-the-art performance.

4. Discussions and Insights

SimPLE closes the training-testing gap. One of the most challenging problems in open-set recognition is the gap between training and testing. Almost all previous innovations were made towards bridging this gap. For example, angular margin [13, 36, 37, 67, 69, 74] is widely adopted in proxy-based learning such that the training target can be closer to the testing scenario. Recently, SphereFace2 [74] was proposed to further bridge this gap by switching from triplet-based learning to pair-based learning, because only pair comparison is performed during testing. However, the use of proxies still prevents SphereFace2 from closing this gap, and moreover, SphereFace2 remains heavily dependent on angular similarity and margin. Our work can actually be viewed as a novel proxy-free generalization of SphereFace2. By dropping the use of class proxies, angular similarity and margin, SimPLE takes one step further towards closing the gap between training and testing in open-set recognition.

SimPLE as a general framework. SimPLE gives a simple yet working variant for pair-based proxy-free learning, but

more importantly, SimPLE identifies a few critical design aspects (*e.g.*, similarity score, pair coverage, pair importance) to achieve PSL's desideratum and opens new possibilities. For example, the optimal similarity score is yet to be designed and how to effectively incorporate hard pair mining without the use of angular similarity remains an open problem. Solving any of these open problems might easily lead to better loss functions in pair-based proxy-free learning.

5. Experiments and Results

We evaluate SimPLE with multiple open-set recognition problems, including face recognition, image retrieval, and speaker verification. We adopt the standard training and testing protocols, network configurations, and optimization strategy, so that our results can be transparently and fairly compared to previous methods. The detailed experimental settings are given in the corresponding subsections.

5.1. Open-set Face Recognition

Experimental setup. We generally follow the data processing and augmentation strategy from [26]. Specifically, the face images are cropped based on the 5 face landmarks detected by MTCNN [83] or RetinaFace [12] using similarity transformation. The cropped image is resized to 112×112, and RGB pixels are normalized to [-1, 1]. In training mode, random cropping, rescaling, and photometric jittering are applied to the face images with a probability of 0.2, while horizontally flipping is applied with the probability of 0.5.

We first evaluate the design of SimPLE by performing ablation studies. SFNet-64 [37] and MS1MV2 [13, 16] are adopted as the backbone and training set, respectively. The validation set is constructed by combining LFW [20], AgeDB-30 [46], CALFW [85], and CPLFW [84], containing 12,000 positive and 12,000 negative pairs. SimPLE models are trained with different r and α. The equal error rates (EER) and the true positive rates at different false positive rates (TPR@FAR) on the validation set are reported.

Ablation: hyperparameter r, α, and b_θ. Hyperparameter r is used to control the strength of sample mining. When $r = 1$, SimPLE is equivalent to vanilla binary cross-entropy. As r increases, SimPLE focuses more on the hard negative pairs. Table 2 shows that SimPLE yields large performance gains when $r > 1$ is used. With $r = 3$ and $\alpha = 0.001$, SimPLE yields 3.23% EER, which outperforms the best result with $r = 1$ (3.72%) by a considerable margin.

We also perform a similar ablation study with different b_θ, and the results are given in the Appendix. In general, $b_\theta = 0.3$ or 0.4 works well for all the experiments. We also observe that higher α is usually paired with higher r for the best performance. With optimal r and α pairs, SimPLE performs equally well. Therefore, we fix $r = 3$, $\alpha = 0.001$, and $b_\theta = 0.3$ in the following experiments.

r	α	EER (↓)	TPR@FAR=1e-4	TPR@FAR=1e-3	TPR@FAR=1e-2
1	0.0002	3.98	87.88	90.56	93.78
1	0.0005	3.72	89.23	92.20	94.49
1	0.001	3.85	88.43	90.91	94.11
1	0.002	4.2	85.45	89.89	93.46
2	0.0005	3.35	90.5	92.38	**94.93**
2	0.001	3.38	88.84	92.10	94.55
2	0.002	3.34	90.36	92.25	94.61
3	0.0005	3.38	89.80	92.00	94.81
3	0.001	3.28	**91.07**	**92.45**	94.80
3	0.002	**3.23**	89.62	92.27	94.84

Table 2: Ablation study of r and α for SimPLE (%).

Ablation: score functions. To investigate the importance of score functions, we run experiments for SimPLE using cosine similarity or generalized inner product (*i.e.*, Eq. 6) as the score function. Experimental results show that the generalized inner product leads to a significantly lower EER over cosine similarity (with optimal hyperparameters), *i.e.* 3.23% *vs.* 4.81%. The results suggest that a proper score function plays a key role in the success of SimPLE.

In our early experimentation, we also attempted to incorporate generalized inner product into the proxy-based framework. However, we did not manage to obtain meaningful results (as shown in the Appendix). This further shows that our SimPLE framework is promising in the sense that it can easily adopt various score functions.

Comparison with previous methods. For a comprehensive comparison, we conduct experiments under three different settings: (A) SFNet-20 trained with VGGFace2 dataset [4] (8.6K subjects), (B) SFNet-64 trained with MS1MV2 dataset (85.7K subjects), and (C) IResNet-100 trained with MS1MV2 dataset. The goal is to explore SimPLE under different network capacities and data scales. The evaluations are performed on IARPA Janus Benchmark (IJB) [42, 76]. This is a challenging dataset since it contains mixed-quality samples, *e.g.* low-quality video frames from surveillance cameras and high-quality images. For setting A and B, we train the face models of different methods using their released code, which ensures all methods use the same training recipes except loss functions. For setting C, we directly use the released models or results reported in their published papers, since they represent the current best performance.

Setting A: small model and training set. We first explore SimPLE in a relatively lightweight setting. As can be seen from Table 3, SimPLE outperforms all competitors by large margins in both verification and identification tasks. In particular, SimPLE respectively outperforms SphereFace2 by 7.38% and 8.30% in TAR@FAR=1e-5 and TPIR@FPIR=1e-2 on IJB-B dataset. Similar performance gains can also be observed on the IJB-C dataset, and it shows that SimPLE is effective for low-capacity architectures and small-scale training sets. Using cosine similarity as score function, SimPLE yields inferior results, which is consistent with the perfor-

	IJB-B						IJB-C					
	1:1 Verification TAR @ FAR			1:N Identification TPIR @ FPIR			1:1 Verification TAR @ FAR			1:N Identification TPIR @ FPIR		
Method	1e-6	1e-5	1e-4	top 1	1e-2	1e-1	1e-6	1e-5	1e-4	top 1	1e-2	1e-1
NormFace [68]	32.53	68.20	82.24	91.17	58.85	78.99	65.64	76.31	86.15	92.09	70.60	81.43
SphereFace [36, 37]	40.11	75.44	87.43	92.97	67.70	84.87	73.79	83.02	90.37	94.19	78.18	86.90
CosFace [67, 69]	40.77	73.66	85.51	91.96	67.97	82.77	70.43	80.21	88.75	93.09	75.36	84.90
ArcFace [13]	40.15	76.52	87.50	92.26	70.25	85.02	74.32	82.49	90.17	93.79	78.22	86.71
Circle Loss [61]	36.56	72.81	86.51	91.41	65.58	83.73	69.69	80.66	89.67	92.96	75.41	85.63
CurricularFace [21]	22.16	63.35	88.23	92.66	47.59	84.93	35.54	76.49	91.10	93.73	54.13	85.77
SphereFace2 [74]	40.19	77.13	87.95	92.36	72.14	87.32	75.38	83.38	90.82	93.24	80.03	87.54
SimPLE (cosine)	40.90	63.09	80.86	91.02	58.06	76.94	51.72	69.49	84.40	92.22	62.10	77.92
SimPLE	**47.00**	**84.51**	**90.72**	**93.19**	**80.44**	**89.18**	**82.34**	**88.62**	**92.92**	**94.51**	**85.66**	**90.84**

Table 3: Comparison on IJB-B and IJB-C. We use SFNet-20 as the backbone architecture and VGGFace2 as the training set. Results are in % and higher number indicates better performance.

	IJB-B						IJB-C					
	1:1 Verification TAR @ FAR			1:N Identification TPIR @ FPIR			1:1 Verification TAR @ FAR			1:N Identification TPIR @ FPIR		
Method	1e-6	1e-5	1e-4	top 1	1e-2	1e-1	1e-6	1e-5	1e-4	top 1	1e-2	1e-1
NormFace [68]	40.56	75.30	90.22	92.49	64.62	88.19	70.17	85.88	92.69	93.70	77.97	89.81
SphereFace [36, 37]	**48.83**	86.66	94.36	94.84	76.35	93.20	83.57	92.79	95.82	96.07	87.74	94.47
CosFace [67, 69]	37.82	82.99	94.20	94.69	70.61	93.03	78.01	92.29	95.87	95.91	84.59	94.53
ArcFace [13]	41.02	86.16	**94.82**	94.88	77.92	**93.79**	84.47	93.25	**96.25**	96.12	88.80	**95.08**
Circle Loss [61]	41.65	82.76	94.09	94.64	74.63	92.83	81.18	91.59	95.83	95.77	84.56	94.15
CurricularFace [21]	43.76	85.55	94.61	94.82	76.01	93.37	83.35	92.95	96.11	96.04	87.88	94.76
SphereFace2 [74]	40.31	85.89	94.04	94.59	78.05	93.02	84.60	92.37	95.74	95.81	88.87	94.52
SimPLE	46.67	**90.34**	94.49	**95.15**	**83.56**	93.62	**88.49**	**93.48**	95.91	**96.36**	**91.88**	94.76

Table 4: Comparison on IJB-B and IJB-C. We use SFNet-64 as the backbone architecture and MS1MV2 as the training set.

	IJB-B						IJB-C					
	1:1 Verification TAR @ FAR			1:N Identification TPIR @ FPIR			1:1 Verification TAR @ FAR			1:N Identification TPIR @ FPIR		
Method	1e-6	1e-5	1e-4	top 1	1e-2	1e-1	1e-6	1e-5	1e-4	top 1	1e-2	1e-1
SphereFace [36, 37]	47.33	90.14	94.87	95.13	82.57	94.30	87.86	94.36	96.25	96.45	91.68	95.36
CosFace [67, 69]	43.67	88.83	95.23	95.35	80.50	94.49	85.29	94.33	96.62	96.53	90.69	95.61
ArcFace [13]	43.43	90.40	95.02	95.14	81.36	94.26	86.00	94.49	96.39	96.47	91.91	95.51
CurricularFace† [21]	-	-	94.86	-	-	-	-	-	96.15	-	-	-
BroadFace† [29]	40.92	89.97	94.97	-	-	-	85.96	94.59	96.38	-	-	-
SCF-ArcFace† [31]	-	90.68	94.74	-	-	-	-	94.04	96.09	-	-	-
SphereFace2 [74]	41.53	89.92	95.02	95.24	83.46	94.36	87.63	94.49	96.42	96.41	92.08	95.47
MagFace+ [43]	42.32	90.36	94.51	94.81	83.65	93.87	90.24	94.08	95.97	96.02	91.95	95.06
AdaFace [26]	46.78	90.04	**95.67**	**95.54**	80.73	**95.07**	89.74	**94.87**	**96.89**	96.75	92.12	**96.20**
SimPLE	**49.87**	**91.13**	94.78	**95.54**	**85.92**	94.28	**90.30**	94.34	96.27	**96.81**	**92.88**	95.49

Table 5: Comparison on IJB-B and IJB-C. We use IResNet-100 as the backbone architecture and MS1MV2 as the training set. '-' indicates that neither the model is released nor the result is reported in their paper. † Results are obtained from their papers.

mance on the validation set. The results indicate that a lot more small insights (*e.g.* margin) are required before it can achieve competitive performance.

Setting B and C: larger model and training set. These experiments are designed to investigate if SimPLE can benefit from larger models and training sets. Again, the comparison is conducted on the IJB datasets and the results are given in Table 4 and Table 5. We observe that SimPLE achieves competitive results on IJB datasets under both settings. Compared to other methods, SimPLE improves more at low accept rates, *e.g.* FPR=1e-6, 1e-5, and FPIR=1e-2. The results validate that SimPLE can benefit from a stronger backbone and more training data.

Proxy-based vs Proxy-free. Both SphereFace2 and SimPLE are pair-wise learning frameworks, while SimPLE removes the proxy, angular assumption, and margin term. As shown in Tables 4 and 5, the improvement of SimPLE over SphereFace2 suggests that these dominating components might not be necessary in the open-set recognition problem. We hope this observation will encourage researchers to rethink the use of each component in the PSL framework.

We further evaluate our SimPLE model trained with setting C on several high-quality datasets, as given in Table 6. SimPLE achieves the highest accuracies on cross-age and cross-pose datasets, *i.e.* 96.25% on CALFW, 94.00% on CPLFW, and 98.77% on CFP-FP, showing the robustness

Method	LFW	AgeDB	CALFW	CPLFW	CFP-FP
SphereFace [36, 37]	99.78	98.02	95.56	92.11	98.08
CosFace [67, 69]	98.81	98.11	95.76	92.28	98.12
ArcFace [13]	98.83	98.28	95.45	92.08	98.27
CurricularFace [21]	99.80	98.32	96.20	93.13	98.37
BroadFace [29]	**99.85**	**98.38**	96.20	93.17	98.63
SCF-ArcFace [31]	99.82	98.30	96.12	93.16	98.40
SphereFace2 [74]	99.80	98.07	95.38	92.20	98.15
MagFace [43]	99.83	98.17	96.15	92.87	98.46
AdaFace [26]	99.82	98.05	96.08	93.53	98.49
SimPLE	99.78	98.28	**96.25**	**94.00**	**98.77**

Table 6: Comparison on multiple high-quality face datasets. Results are in % and higher number indicates better performance.

Method	Precision@1	R-Precision	MAP@R
Contrastive [17]	68.13	37.24	26.53
Triplet [73]	64.24	34.55	23.69
NT-Xent [5, 51, 58]	66.61	35.96	25.09
ProxyNCA [47]	65.69	35.14	24.21
Margin [77]	63.60	33.94	23.09
Margin/class [77]	64.37	34.59	23.71
N. Softmax [68, 86]	65.65	35.99	25.25
CosFace [67, 69]	67.32	37.49	26.70
ArcFace [13]	67.50	37.31	26.45
FastAP [3]	63.17	34.20	23.53
SNR [82]	66.44	36.56	25.75
MS [71]	65.04	35.40	24.70
MS+Miner [71]	67.73	37.37	26.52
SoftTriple [55]	67.27	37.34	26.51
SimPLE	**68.58**	**37.62**	**26.84**

Table 7: Performance of Image Retrieval on CUB-200-2011.

of SimPLE to varying age and pose. The best performance on the LFW and AgeDB datasets (99.85% and 98.38%) is obtained by BroadFace, which is a hybrid method that combines proxy-based and proxy-free PSL. Our results suggest that the proxy-free PSL paradigm is still worth exploring and should not be ignored for open-set recognition.

5.2. Image Retrieval and Speaker Verification

We evaluate SimPLE on two more open-set recognition problems: image retrieval and speaker verification.

Image Retrieval. We use the codebase in [48], which is a well-known benchmarking toolkit for image retrieval and metric learning. For all the methods, the data processing, training recipes, and testing protocols are nearly the same, except the loss functions. This ensures a fair comparison of different methods. As suggested in [48], we use BN-Inception as the backbone [22] with ImageNet pretraining. The precision at 1 (also known as top-1 / rank-1 accuracy), R-precision, and Mean Average Precision at R (MAP@R) on CUB-200-2011 dataset [65] are reported in Table 7.

Speaker Verification. We adopt the standard train/val/test split given by VoxCeleb2 [10]. The speech recordings are randomly cropped to 3-8 seconds in each mini-batch as data

Method	VoxCeleb1	VoxCeleb1-E	VoxCeleb1-H
Softmax	2.11	2.05	3.76
A-Softmax [36, 37]	2.11	2.11	3.47
AM-Softmax [67, 69]	2.17	2.16	3.49
AAM-Softmax [13]	2.22	2.21	3.55
SimPLE	**1.85**	**1.80**	**3.23**

Table 8: Performance of Speaker Verification on VoxCeleb1.

augmentation. The mini-batch size is set to 512. We use ResNet-34 as the backbone architecture. To learn the networks from scratch, the SGD optimizer is used and the learning rate is initialized at 0.1 and divided by 10 after 30K, 50K, and 60K iterations. The training is completed at 70K iterations. We report the EER on VoxCeleb1, VoxCeleb1-easy, VoxCeleb1-hard in Table 8.

Unsurprisingly, SimPLE achieves consistently competitive results on CUB-200-2011 and VoxCeleb1 datasets (Table 7 and 8). The pipeline of different methods is the same, so the gains can only be attributed to the better PSL loss function. This shows that the applications of SimPLE are not limited to any particular object (face) or data modal (image). It appears to perform well on a variety of open-set recognition problems, *e.g.* generic object or speech data.

6. Related Work and Concluding Remarks

How to learn discriminative representations has been a shared goal of multiple lines of research. We conclude our paper by discussing some highly related work.

Contrastive learning. There has been a rapidly growing interest [5, 6, 15, 18, 51, 64] in learning representations with instance contrast. The core idea is to view a sample and its augmented versions as a class and learn to group their representations while contrasting with other samples. As a popular loss function in this line, InfoNCE [51] uses multi-class cross-entropy while SimPLE uses binary cross-entropy.

Class proxy design. Although proxy-based PSL typically updates the class proxies by back-propagation from the loss function, there exist other ways to design class proxies. several methods [19, 32, 39, 54, 79] use fixed classifiers and still obtain satisfactory performance. Liu et al. [34] use stochastic proxies for a large number of categories. Class proxies can also be designed to achieve certain properties [23, 44, 53, 79] (*e.g.*, uniformity [33, 35]). Proxy-free methods bypass the difficulty of designing or learning proxies, but they instead introduce the problem of pair construction and mining. Different pair mining strategies in proxy-free PSL may implicitly inject different inductive biases for the learned features, and the mechanism behind is of great importance. How to combine the advantages of proxy-based and proxy-free methods and achieve a good trade-off remains an open challenge.

Deep metric learning. Traditional metric learning [8, 17, 73, 78] learns a proper distance metric that satisfies non-

negativity and the triangle inequality. More recently, deep metric learning [50, 58, 59] achieves promising performance on image retrieval by using neural networks to learn a feature representation and then put it into a proper distance function. In contrast, PSL drops the requirement to learn a proper distance metric. Recent DML methods [50, 55, 58, 59, 70, 72] are mostly triplet-based, while our SimPLE is pair-based.

Concluding remarks. In this work, we start by rethinking the desideratum of pairwise similarity learning. We then challenge a few common components in current PSL methods, such as angular similarity and margin. We argue that they can be safely removed in pair-based proxy-free frameworks. Following the desideratum, we design a simple yet effective PSL method. Extensive experiments show that SimPLE is able to achieve state-of-the-art performance in a diverse set of open-set recognition tasks.

7. Limitations and Open Problems

SimPLE follows a simple yet intuitive design, and yet is by no means, an optimal one. For example, the hard pair mining for positive and negative pairs is still less straightforward and could be further improved. Moreover, similar to existing proxy-free methods, SimPLE can be quite sensitive to the design of pair construction and mining. Despite some limitations, our paper aims to demonstrate that angular similarity is not the only way to achieve state-of-the-art performance.

While SimPLE exhibits empirical superiority in many tasks, a few open problems remain. First, SimPLE introduces three new hyperparameters: b_θ, r, α, which means there is one more hyperparameter to tune, compared to the well-known margin-based softmax cross-entropy losses like SphereFace, CosFace, and ArcFace. While hyperparameters in SimPLE have clear physical interpretations, it is still desirable to reduce the number of hyperparameters without sacrificing performance. This requires a deeper understanding of PSL's desideratum. Second, SimPLE achieves less significant performance gain with large training sets. This is mainly due to the naive pair construction strategy, which cannot cover all the representative pairs. We expect that advanced pair sampling methods could be an important future direction. Third, our paper presents a desideratum for general PSL and SimPLE is just one possible route to achieve this desideratum. There may exist many other routes that may potentially perform PSL more effectively.

Finally, the PSL problem is in fact very general. learning a common embedding space for multi-modal data pairs (*e.g.*, image-text pairs [56]) can also be viewed as a PSL problem. How to apply various PSL methods (including SimPLE) to multi-modal pretraining is a promising direction to explore.

Acknowledgements

The authors would like to thank colleges from Max Planck Institute for Intelligent Systems at Tübingen for many inspiring discussions. This work was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B, and by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

WL was supported by the German Research Foundation (DFG): SFB 1233, Robust Vision: Inference Principles and Neural Mechanisms, TP XX, project number: 276693517.

AW acknowledges support from a Turing AI Fellowship under grant EP/V025379/1, and the Leverhulme Trust via Leverhulme Centre for the Future of Intelligence.

MJB has received research gift funds from Adobe, Intel, Nvidia, Meta/Facebook, and Amazon.MJB has financial interests in Amazon, Datagen Technologies, and Meshcapade GmbH. While MJB is a consultant for Meshcapade, his research in this project was performed solely at, and funded solely by, the Max Planck Society.

References

[1] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Teva Merlin, Javier Ortega-García, Dijana Petrovska-Delacrétaz, and Douglas A Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004. 1

[2] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *ECCV*, 2020. 2

[3] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, 2019. 8

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *IEEE FG*, 2018. 6

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 8

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 8

[7] Hongjun Choi, Anirudh Som, and Pavan Turaga. Amc-loss: Angular margin contrastive loss for improved explainability in image classification. In *CVPR Workshops*, 2020. 2

[8] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 2, 3, 8

[9] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. In defence of metric learning for speaker recognition. In *Interspeech*, 2020. 4

[10] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *Interspeech 2018*, 2018. 8

[11] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 2008. 1

[12] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020. 6

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[14] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *TPAMI*, 2020. 1

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020. 8

[16] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016. 6

[17] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2, 3, 8

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 4, 8

[19] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: the marginal value of training the last weight layer. *arXiv preprint arXiv:1801.04540*, 2018. 8

[20] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Technical Report*, 2007. 6

[21] 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 *CVPR*, 2020. 7, 8

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 8

[23] Tejaswi Kasarla, Gertjan Burghouts, Max van Spengler, Elise van der Pol, Rita Cucchiara, and Pascal Mettes. Maximum class separation as inductive bias in one matrix. In *NeurIPS*, 2022. 8

[24] Umair Khan and Francisco Javier Hernando Pericás. Unsupervised training of siamese networks for speaker verification. In *Interspeech*, 2020. 2

[25] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *NeurIPS*, 2020. 2

[26] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *CVPR*, 2022. 2, 6, 7, 8

[27] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, 2020. 2

[28] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *CVPR*, 2019. 2

[29] Yonghyun Kim, Wonpyo Park, and Jongju Shin. Broadface: Looking at tens of thousands of people at once for face recognition. In *ECCV*, 2020. 7, 8

[30] Ibuki Kuroyanagi, Tomoki Hayashi, Kazuya Takeda, and Tomoki Toda. Anomalous sound detection using a binary classification model and class centroids. In *EUSIPCO*, 2021. 2

[31] Shen Li, Jianqing Xu, Xiaqing Xu, Pengcheng Shen, Shaoxin Li, and Bryan Hooi. Spherical confidence learning for face recognition. In *CVPR*, 2021. 7, 8

[32] Zexi Li, Xinyi Shang, Rui He, Tao Lin, and Chao Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *ICCV*, 2023. 8

[33] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, 2018. 8

[34] Weiyang Liu, Rongmei Lin, Zhen Liu, James M Rehg, Liam Paull, Li Xiong, Le Song, and Adrian Weller. Orthogonal over-parameterized training. In *CVPR*, 2021. 3, 8

[35] Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. Learning with hyperspherical uniformity. In *AISTATS*, 2021. 8

[36] Weiyang Liu, Yandong Wen, Bhiksha Raj, Rita Singh, and Adrian Weller. Sphereface revived: Unifying hyperspherical face recognition. *TPAMI*, 2022. 2, 5, 7, 8

[37] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

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

[39] Weiyang Liu, Longhui Yu, Adrian Weller, and Bernhard Schölkopf. Generalizing and decoupling neural collapse via hyperspherical uniformity gap. In *ICLR*, 2023. 2, 8

[40] Weiyang Liu, Yan-Ming Zhang, Xingguo Li, Zhiding Yu, Bo Dai, Tuo Zhao, and Le Song. Deep hyperspherical learning. In *NeurIPS*, 2017. 3

[41] Yi Liu, Liang He, and Jia Liu. Large margin softmax loss for speaker verification. In *Interspeech*, 2019. 4

[42] 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 *ICB*, 2018. 6

[43] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. In *CVPR*, 2021. 2, 7, 8

[44] Pascal Mettes, Elise Van der Pol, and Cees Snoek. Hyperspherical prototype networks. In *NeurIPS*, 2019. 8

[45] Zuheng Ming, Joseph Chazalon, Muhammad Muzzamil Luqman, Muriel Visani, and Jean-Christophe Burie. Simple triplet loss based on intra/inter-class metric learning for face verification. In *ICCV Workshops*, 2017. 2

[46] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *CVPR Workshops*, 2017. 6

[47] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017. 2, 8

[48] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, 2020. 4, 8

[49] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 2

[50] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *CVPR*,

2017. 9

[51] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 8

[52] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015. 2

[53] Federico Pernici, Matteo Bruni, Claudio Baecchi, and Alberto Del Bimbo. Regular polytope networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 8

[54] Federico Pernici, Matteo Bruni, Claudio Baecchi, Francesco Turchini, and Alberto Del Bimbo. Class-incremental learning with pre-allocated fixed classifiers. In *ICPR*, 2021. 8

[55] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*, 2019. 2, 8, 9

[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 9

[57] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2, 3, 4

[58] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016. 2, 8, 9

[59] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 2, 9

[60] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *NeurIPS*, 2014. 2

[61] 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 *CVPR*, 2020. 2, 7

[62] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014. 2

[63] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 2

[64] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 8

[65] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *Technical Report from California Institute of Technology*, 2011. 8

[66] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *ICASSP*, 2018. 4

[67] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE SPL*, 2018. 1, 2, 3, 4, 5, 7, 8

[68] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *ACM-MM*, 2017. 1, 2, 3, 7, 8

[69] 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 *CVPR*, 2018. 2, 3, 4, 5, 7, 8

[70] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *ICCV*, 2017. 2, 9

[71] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, 2019. 2, 8

[72] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In *CVPR*, 2019. 2, 9

[73] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009. 2, 3, 8

[74] Yandong Wen, Weiyang Liu, Adrian Weller, Bhiksha Raj, and Rita Singh. Sphereface2: Binary classification is all you need for deep face recognition. In *ICLR*, 2021. 2, 4, 5, 7, 8

[75] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 2

[76] 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 *CVPR Workshops*, 2017. 6

[77] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017. 2, 4, 8

[78] Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. Distance metric learning with application to clustering with side-information. In *NeurIPS*, 2002. 8

[79] Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? In *NeurIPS*, 2022. 8

[80] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *TPAMI*, 2021. 1

[81] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *ICCV*, 2019. 2

[82] Tongtong Yuan, Weihong Deng, Jian Tang, Yinan Tang, and Binghui Chen. Signal-to-noise ratio: A robust distance metric for deep metric learning. In *CVPR*, 2019. 2, 8

[83] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE SPL*, 2016. 6

[84] 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. 6

[85] 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. 6

[86] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *CVPR*, 2019. 8

Appendix

A. Effect of Hyperparameters

We explore the effect of b_θ under different hyperparameter settings. The results are given in Table 9.

b_θ	r	α	EER (↓)	TPR@FAR=1e-4	TPR@FAR=1e-3	TPR@FAR=1e-2
0.2	1	0.0005	4.58	86.52	90.58	93.55
0.2	1	0.001	4.66	83.78	90.53	93.64
0.2	1	0.002	4.66	83.24	90.40	93.36
0.2	2	0.0001	3.93	88.93	91.63	94.17
0.2	2	0.0002	4.29	87.55	91.03	93.57
0.2	2	0.0005	3.65	88.75	92.15	94.24
0.2	2	0.001	4.64	79.22	88.54	92.87
0.2	2	0.002	4.13	87.93	91.64	94.16
0.2	2	0.005	4.23	84.09	90.19	93.57
0.2	3	0.0002	6.24	73.60	82.07	89.84
0.2	3	0.0005	4.07	82.18	90.20	93.28
0.3	1	0.0002	3.98	87.88	90.56	93.78
0.3	1	0.0005	3.72	89.23	92.20	94.49
0.3	1	0.001	3.85	88.43	90.91	94.11
0.3	1	0.002	4.20	85.45	89.89	93.46
0.3	2	0.0005	3.35	90.5	92.38	94.93
0.3	2	0.001	3.38	88.84	92.10	94.55
0.3	2	0.002	3.34	90.36	92.25	94.61
0.3	3	0.0005	3.38	89.80	92.00	94.81
0.3	3	0.001	3.28	91.07	92.45	94.80
0.3	3	0.002	3.23	89.62	92.27	94.84
0.4	1	0.0002	3.59	85.80	91.68	94.45
0.4	1	0.0005	3.77	85.84	91.35	94.56
0.4	1	0.001	3.71	87.78	91.40	94.36
0.4	2	0.0005	3.58	88.50	92.33	94.94
0.4	2	0.001	3.34	88.67	92.85	94.94
0.4	2	0.002	3.35	89.25	92.12	94.50
0.4	3	0.0005	3.38	90.31	92.10	94.69

Table 9: Ablation study of b_θ, r and α for SimPLE (%).

There are generally two types of hyperparameters in SimPLE: (1) common hyperparameters which are already extensively studied in existing methods, *e.g.*, learning rate, weight decay, size of the memory buffer; (2) our own hyperparameters: b_θ, r, α.

First, SimPLE only has one more hyperparameter than popular margin-based softmax cross-entropy losses. Second, all these hyperparameters are physically interpretable and easy to tune (their feasible range is small). $\alpha \in (0, 1)$ is the balancing ratio between positive and negative pairs, which can be set roughly according to the actual ratio of positive and negative pairs in each batch. $b_\theta \in (0, 1)$ is the angular bias. Setting b_θ around 0.3 works well. $r > 0$ controls the easy/hard sample mining. When r gets larger, the loss focuses more on easy positive pairs and hard negative pairs. We empirically show that $r = 1$ can already achieve good results. More importantly, SimPLE demonstrates consistent performance gain across a wide range of hyperparameter setup (see appendix). The best hyperparameter setup is generally robust to different tasks, datasets, and architectures.

B. Applying Generalized Inner Product to Proxy-based Methods

Since generalized inner product achieves significant improvement in SimPLE, we are interested in how it works in proxy-based methods. Here we apply generalized inner product to two representative proxy-based methods: vanilla cross-entropy loss and CosFace. The formulations are given as follows. Note that $\mathcal{L}_{\text{CosFace}^*}$ is equivalent to $\mathcal{L}_{\text{CE}^*}$ when margin m is 0.

$$\mathcal{L}_{\text{CE}^*} = \log\big(1 + \sum_{i \neq y} \exp(\|\boldsymbol{w}_i\| \cdot \|\tilde{\boldsymbol{x}}\| \cdot (\cos(\theta_{\tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{x}}}) - b_\theta) - \|\boldsymbol{w}_y\| \cdot \|\tilde{\boldsymbol{x}}\| \cdot (\cos(\theta_{\tilde{\boldsymbol{w}}_y, \tilde{\boldsymbol{x}}}) - b_\theta)\big) \tag{9}$$

$$\mathcal{L}_{\text{CosFace}^*} = \log\big(1 + \sum_{i \neq y} \exp(\|\boldsymbol{w}_i\| \cdot \|\tilde{\boldsymbol{x}}\| \cdot (\cos(\theta_{\tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{x}}}) - b_\theta - m) - \|\boldsymbol{w}_y\| \cdot \|\tilde{\boldsymbol{x}}\| \cdot (\cos(\theta_1 \tilde{\boldsymbol{w}}_y, \tilde{\boldsymbol{x}}) - b_\theta)\big) \tag{10}$$

We adopt Setting A in this ablation. b_θ is set from 0 to 0.9 for the vanilla cross-entropy loss. As shown in Table 10, we do not observe improved performance when applying generalized inner product to the proxy-based method. In contrast, our proxy-free SimPLE can enjoy accuracy gains from the generalized inner product, producing substantially better results.

For CosFace, we also try different combinations of b_θ and m. However, the models do not converge even if we use a very small m (*i.e.* 0.1), resulting in chance-level performance. This is also consistent with the observations in a large body of literature, where cosine similarity has become a *de facto* choice.

b_θ	m	EER (↓)	TPR@FAR=1e-4	TPR@FAR=1e-3	TPR@FAR=1e-2
0	0	6.12	19.12	38.23	77.82
0.1	0	6.08	15.48	37.95	80.70
0.2	0	5.99	24.11	52.16	82.88
0.3	0	6.10	21.02	52.77	84.27
0.4	0	6.40	31.58	52.03	84.30
0.5	0	7.00	38.78	63.12	83.08
0.6	0	7.47	35.78	60.34	82.07
0.7	0	8.05	40.47	55.13	80.58
0.8	0	8.91	40.33	54.98	77.36
0.9	0	9.83	34.63	54.72	75.23
0-0.9	0.1	Not converged			
0-0.9	0.2	Not converged			
CE Loss		6.13	37.39	58.05	84.78
SimPLE (cosine)		5.15	61.60	73.04	87.42
SimPLE		4.61	68.92	78.97	90.10

Table 10: Results of applying generalized inner product to proxy-based methods.