Unsupervised Hyperbolic Metric Learning

Jiexi Yan[1], Lei Luo[3], Cheng Deng[1]*, Heng Huang[2,3]
[1]School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2]Department of Electrical and Computer Engineering, University of Pittsburgh, PA 15260, USA
[3]JD Finance America Corporation, Mountain View, CA 94043, USA
{jxyan1995,luoleipitt, chdeng.xd}@gmail.com, heng.huang@pitt.edu

Abstract

Learning feature embedding directly from images without any human supervision is a very challenging and essential task in the field of computer vision and machine learning. Following the paradigm in supervised manner, most existing unsupervised metric learning approaches mainly focus on binary similarity in Euclidean space. However, these methods cannot achieve promising performance in many practical applications, where the manual information is lacking and data exhibits non-Euclidean latent anatomy. To address this limitation, we propose an Unsupervised Hyperbolic Metric Learning method with Hierarchical Similarity. It considers the natural hierarchies of data by taking advantage of Hyperbolic metric learning and hierarchical clustering, which can effectively excavate richer similarity information beyond binary in modeling. More importantly, we design a new loss function to capture the hierarchical similarity among samples to enhance the stability of the proposed method. Extensive experimental results on benchmark datasets demonstrate that our method achieves state-of-the-art performance compared with current unsupervised deep metric learning approaches.

1. Introduction

Learning a precise distance metric for similarity measurement is a key ingredient of various computer vision tasks, such as face recognition [13, 50], image classification [5, 53, 6], and person re-identification [52]. Therefore, metric learning has aroused much attention and many classical methods have been proposed in the past decades [6, 45, 13]. With the resurgence of deep neural networks, Deep Metric Learning (DML) has emerged as a powerful tool in many practical applications [34, 30, 2, 29, 44]. It targets at seeking a reliable embedding space by virtue of nonlinear deep neural networks, where a well-designed metric loss function brings positive samples closer to anchors, but pushes negative samples far away from the anchors.

Most of the existing DML methods usually use large-scale data for training. They can be roughly divided into two categories: structure-learning methods and hard mining methods. For the former, the crucial point is to construct a proper loss function that plays a key role in many well-known DML methods. To this end, numbers of objectives [5, 34, 30, 40, 35, 29, 32, 20], including commonly-used contrastive loss [5], triplet loss [34] and lifted structure loss [30], have been reported to mine underlying similarity relationships among training data in the literature. While the second category, *i.e.*, hard mining approaches, intends to enhance the discriminative ability of the learned embedding by sampling meaningful hard examples. Since training with numerous easy examples may suffer from inefficiency and poor performance, hard sample mining has become a prevalent technique in DML [30, 15, 12, 10, 9, 36].

However, in real-world tasks, supervised DML methods are often inapplicable since the labeled data is not available. To address this issue, many unsupervised deep learning algorithms have been introduced [48, 51, 17, 49], which attempt to learn the inherent structure of training data without using explicitly-provided labels. A common unsupervised DML manner mines potential sample relationship by an auxiliary algorithm such as clustering, and then utilizes the learned pair-wise information as input to perform the DML task. For example, MOM [18] exploits a random walk process to discover the neighborhood of unlabeled data in the manifold space and the Euclidean space to excavate the pairwise information. Compared with the ground truth, the learned pairwise information usually contains label noise, which makes the DML stage unstable. Therefore, how to discover more semantic information as supervision is still a big challenge. Moreover, inspired by self-supervised learning [17, 49], TAC-CCL [24] integrates self-supervised module into the common unsupervised DML framework to boost the performance. Nonetheless, this algorithm ignores the latent metric information of unlabeled data. It is not

*J.Y. and L.L. made equal contributions, C.D. is corresponding author.

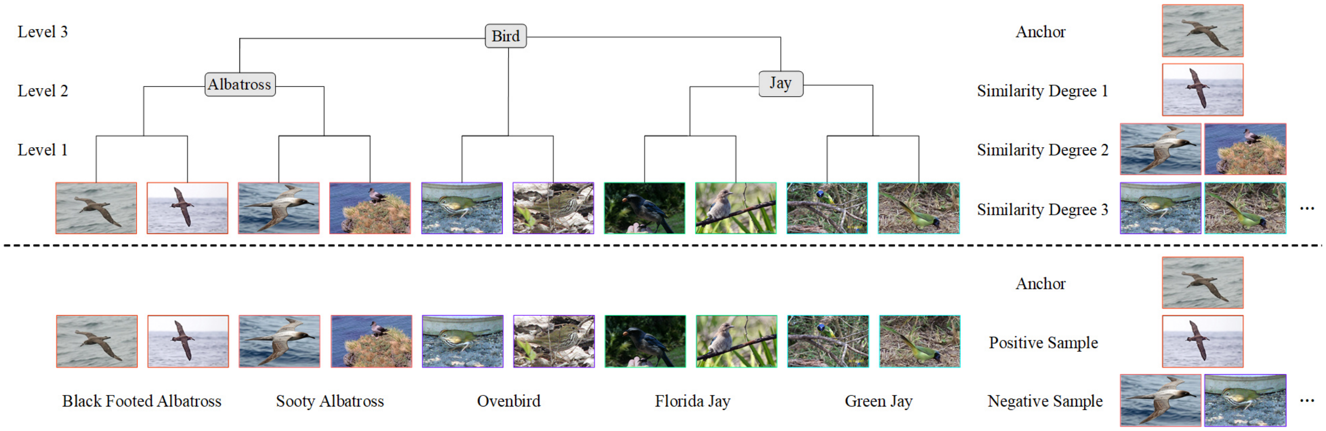

Figure 1. The brief description of different ways to excavate and use similarity information in CUB dataset. In conventional deep metric learning (bottom), given an anchor, we can only address positive pairs and negative pairs as supervision, and all negative pairs will get the same similarity. In fact, due to the complicated hierarchy, negative samples of the same anchor have different similarity degrees. To tackle this problem, our method (top) excavates this hierarchical information and give different negative pairs different similarity degrees.

so much an unsupervised DML method as a plug-and-play self-supervised module extended to DML.

To the best of our knowledge, as shown in Figure 1, images often contain representatives of multiple classes in real-world applications. For example, suppose an anchor in *Green Jay*, instance in *Florida Jay* and *Ovenbird* are both negative samples, but the anchor is more similar to the point of *Florida Jay* than one of *Ovenbird*. Because all samples in *Florida Jay* and *Green Jay* belong to the same second class *Jay*. Previous approaches typically relying on binary labels indicating whether the image pairs are similar or not only address a small subset of similarity relations [21]. Due to the powerful performance of deep learning with labeled data, such supervised DML methods can sometimes obtain good enough results. However, lacking explicitly-provided labels, the performance of these unsupervised DML methods with binary supervision is severely degraded in facing some specific scenarios. On the other hand, most existing DML methods prefer to use Euclidean embeddings to facilitate calculation. However, recent research has proven that many types of data from a multitude of fields (*e.g.* Network Science and Computer Vision) exhibit a highly non-Euclidean latent anatomy [1]. In such cases, these DML methods based on Euclidean space obviously do not provide the most powerful or meaningful geometrical representations of data. As a result, to improve the model performance, it is extremely important and challenging to capture the complicated structure that implicitly exists in real data.

In this work, we propose a novel unsupervised DML method, dubbed Unsupervised Hyperbolic Metric Learning with Hierarchical Similarity, which can effectively excavate the inherent semantic information from unlabeled data. Considering the hierarchical relations between images shown in Figure 1, we first embed the data points from original Euclidean space into Hyperbolic space, which induces a new Hyperbolic DML framework. Specifically, we use hierarchical clustering to generate pseudo hierarchical labels rather than binary labels as supervision for DML task as illustrated in Figure 1. And then, we design a novel loss function to enhance the stability of the model using the inherent richer similarity information discovered by hierarchical clustering. It should be noted that the proposed loss takes the similarity degrees of data pairs into account. Thus, it can well characterize the multi-level relations in the learned hyperbolic embedding space, which is suitable for dealing with triplet supervision task. Our contributions can be summarized as follows:

- We propose the first hyperbolic unsupervised deep metric learning framework, which can well capture the hierarchical structure of data by conducting hierarchical clustering in Hyperbolic embedding space.
- We design a new metric loss function for hierarchical relations. Unlike existing metric losses which are only interested in binary similarity, our loss aims to discover richer similarity information in unsupervised manner by taking full advantages of the learned hierarchical labels.
- Our proposed model achieves the state-of-the-art performances on clustering and retrieval tasks over three benchmark datasets, including CARS196, CUB-200-2011 and Stanford Online Products.

2. Related Work

In this section, we review the basic facts about deep metric learning and hyperbolic geometry.

2.1. Deep Metric Learning

With the significant progress of deep learning, a number of deep metric learning approaches have been proposed to learn non-linear mappings of input images [5, 34, 46, 37]. Many recent deep metric learning methods are built on pair-based [14, 40, 30, 42], optimized by computing the pairwise similarities between instances in the embedding space, and Proxy-based [27, 32, 20], guided by comparing each sample with proxies. Generally, pair-based methods can be cast into a unified weighting formulation through General Pair Weighting (GPW) framework [42]. Hard example mining is another often-used technique to speed up convergence and enhance the discriminative power of feature embeddings in deep metric learning [6, 15, 12, 36]. In addition, considering the limitation of mini-batch training, where only a mini-batch of instances is accessible at each iteration, Cross-batch memory (XBM) [43] provides a memory bank for the feature embeddings of past iterations. To this end, the informative pairs can be identified across the dataset instead of a mini-batch. However, most of the mentioned methods can only deal with binary similarity.

2.2. Hyperbolic Geometry

Hyperbolic geometry is a non-Euclidean geometry, which drops the parallel line postulate while keeping the remaining four of the five of the postulates of Euclidean geometry. In contrast with the hypersphere $\mathbb{S}^d$ and the Euclidean space $\mathbb{R}^d$, the hyperbolic space $\mathbb{H}^d$ can be constructed using various isomorphic models such as the Poincaré half-plane model and the Poincaré ball. The d-dimensional Poincaré ball $\mathbb{D}^d_\tau$ is a model of the hyperbolic space $\mathbb{H}^d$ with curvature τ. Intuitively, hyperbolic spaces can be thought of as continuous versions of trees, which makes it suitable for constructing hierarchical structure information as shown in Figure 2. Hence, trees can be embedded with arbitrarily low error into the Poincaré model of hyperbolic geometry.

Recently, there has been significant research interest on hyperbolic geometry. For example, hyperbolic embeddings have become a popular technique to model real data with tree structure from network science [28]. In order to capture the natural hierarchies of data, hyperbolic embeddings have been successfully integrated into neural networks in the field of computer vision [25, 26, 19] and natural language processing [39]. In particular, hyperbolic (Graph Convolutional) neural networks [4, 11] have been proposed to lead to more faithful embeddings and accurate models. These developments construct the analogs of familiar layers in hyperbolic spaces, *i.e.*, the core neural network operations are conducted in a model of hyperbolic space.

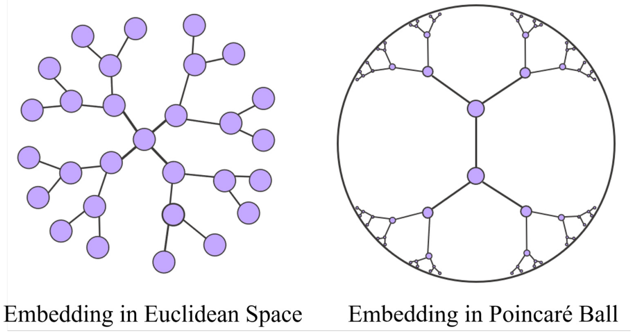

Figure 2. The brief comparison between embedding of trees in Euclidean space (left) and Poincaré Ball (Right). In Poincaré Ball, purple curves are same length geodesics, *i.e*. ”straight lines”.

3. Methodology

3.1. Overview

We present a novel hyperbolic deep metric learning method named Unsupervised Hyperbolic Deep Metric Learning with Hierarchical Similarity which provides a hyperbolic DML model towards unlabeled data by mining and using hierarchical similarity information. Our network structure is shown in Figure 3. The model can be divided into two modules, the hyperbolic metric learning module and the hierarchical clustering module.

Given a training set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ without explicitly-provided labels, we first extract image features to build a hyperbolic metric space $\mathcal{Z} = \{\mathbf{z}_i = f(\mathbf{x_i}|\theta)\}_{i=1}^n$ through the hyperbolic metric learning module initialized by pre-train model. And then, we conduct hierarchical clustering on the learned hyperbolic metric space. According to the hierarchical clustering result $\mathcal{H}$, similarity degree $\mathcal{S} = \{s_{ij}\}_{i,j=1}^n$ of sample pairs will be calculated. Using similarity $\mathcal{S}$ as supervision, we can fine-tune the hyperbolic metric learning module guided by our proposed new loss with hierarchical similarity. Throughout this paper, $\|\cdot\|$ denotes the l_2-norm of a vector.

3.2. Hyperbolic Metric Learning

In many real-world applications, only raw data without any extra supervised information (*e.g*., explicitly-provided labels) can be available. In this scenario, how to discover richer similarity from data itself becomes important. Considering the intrinsic semantic structure of data described in Figure 1, we hope to derive a new metric learning framework to capture such hierarchical similarity. The negative curvature of the hyperbolic space is widely known to accurately capture parent-child relationships [28, 11, 7]. Inspired by this principle, hierarchical relations between training samples call for the use of hyperbolic geometry in our method. Therefore, we introduce a hyperbolic metric learn-

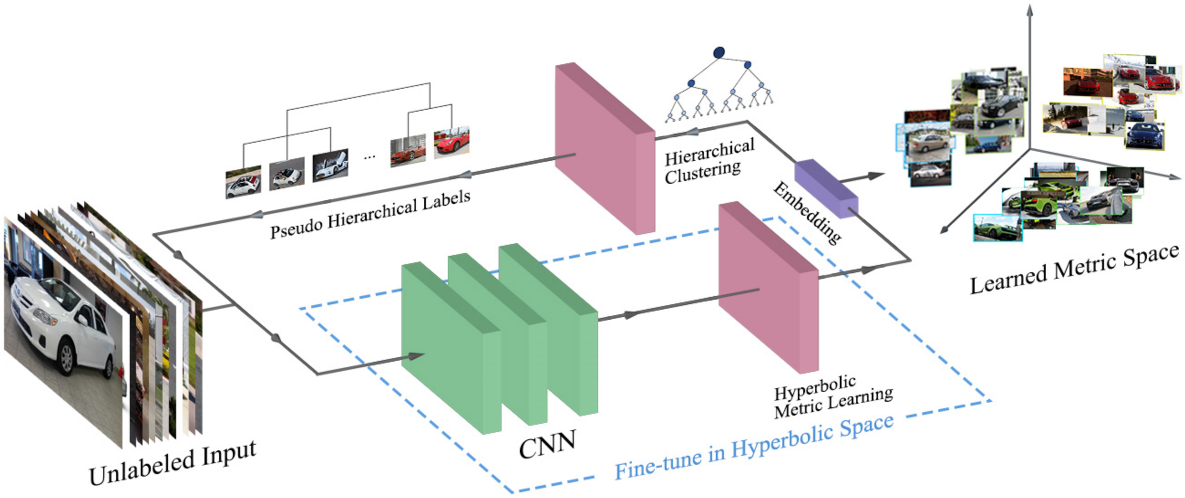
Pseudo Hierarchical Labels
...
Hierarchical Clustering
Embedding
Learned Metric Space
Unlabeled Input
CNN
Hyperbolic Metric Learning
Fine-tune in Hyperbolic Space

Figure 3. The hyperbolic unsupervised deep metric learning framework of our proposed metehod. As the figure shows, our network consists of two modules, hierarchical clustering module and hyperbolic metric module, which can be trained simultaneously. In each iteration, the hierarchical clustering module conducts hierarchical clustering over the learned metric space, and then the metric module is trained with the supervision of the pseudo hierarchical labels yielded by the clustering process. For the initialization, we use the pre-trained model to extract feature vectors for clustering.

ing framework that benefits from the expressiveness of both metric neural networks and hyperbolic embeddings.

There are several well-studied models of hyperbolic geometry, which endow Euclidean space with a hyperbolic metric. Following the majority of existing works, we consider the Poincaré ball model of hyperbolic space, which corresponds to a Riemannian manifold with a particular metric tensor. The Poincaré ball model is defined by the manifold $\mathbb{D}_\tau^d = \{\mathbf{x} \in \mathbb{R}^d : \tau\|\mathbf{x}\| < 1, \tau \geq 0\}$, where additional hyperparameter τ denotes the curvature of Poincaré ball. In this model, the induced distance between any two points $\mathbf{z}_i, \mathbf{z}_j \in \mathbb{D}_\tau^d$ is given by the following expression [28]:

$$d_{\mathbb{D}}(\mathbf{z}_i, \mathbf{z}_j) = \cosh^{-1}\left(1 + 2\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{(1-\|\mathbf{z}_i\|^2)(1-\|\mathbf{z}_j\|^2)}\right). \tag{1}$$

Then, we add the hyperbolic network layer at the end of the original deep metric learning model (*i.e*. convolutional neural network with a full connected layer) to map the input features from $\mathbb{R}^n$ to the hyperbolic manifold $\mathbb{D}_\tau^n$ via the "exp" mapping, which is given by:

$$\mathbf{z} = \exp^\tau(\mathbf{x}) := \tanh\left(\sqrt{\tau}\|\mathbf{x}\|\right)\frac{\mathbf{x}}{\sqrt{\tau}\|\mathbf{x}\|}. \tag{2}$$

In this module, we use Euclidean operations in most layers (*i.e*. convolutional neural network with a full connected layer), and utilize the "exp" map to move from the Euclidean to hyperbolic space at the end of the network.

3.3. Hierarchical Similarity Generation

For better guiding the hyperbolic metric learning module, we hope to discover richer relation information rather than binary similarity. Hierarchical clustering is an effective and often-used tool for discovering meaningful representations of data. As shown in Figure 4, in hierarchical clustering, data points are arranged as the leaves of a multi-layered tree structure with internal nodes representing meaningful and potentially overlapping sub-clusters of the data. To this end, we conduct hierarchical clustering in the learned hyperbolic space.

In each merging step of hierarchical clustering, we calculate the distance between any two sub-clusters as:

$$d_{ab} = \frac{1}{n_a n_b}\sum_{\mathbf{z}_i^a \in C_a, \mathbf{z}_j^b \in C_b}\|\mathbf{z}_i^a - \mathbf{z}_j^b\|, \tag{3}$$

where $\mathbf{z}_i^a$, $\mathbf{z}_j^b$ are samples in the sub-cluster C_a, C_b respectively, and n_a, n_b represent the number of samples in C_a, C_b respectively. The closest two sub-clusters will be grouped together and become a new sub-cluster.

After hierarchical clustering, we can get the distance relationship between all sub-clusters. According to distance calculated by Eq. (3), the similarity levels of these sub-clusters will be derived through setting distance threshold δ. With distance threshold δ, sub-clusters whose distance is less than δ will be aggregated. For example, in Figure 4, we set distance threshold $\{5, 10, 15\}$, and obtain three similarity levels. Under different similarity levels, data is divided into different sub-clusters, *e.g*., data is composed of

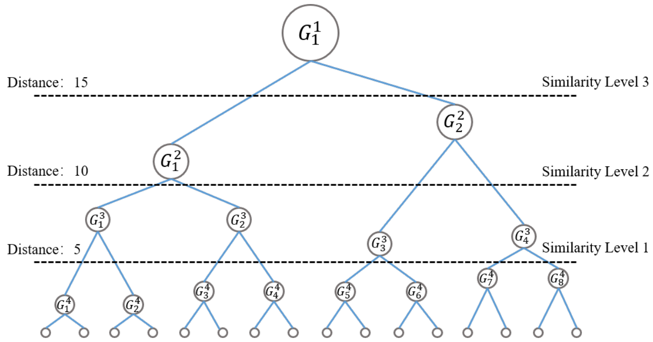

Figure 4. A sample index tree as result of hierarchical clustering for illustration. In each merging step, two clusters will be merged into a new cluster according to the distance between clusters. Therefore, we can set different distance threshold to obtain similarity level. For example, we set distance threshold $\{5, 10, 15\}$, and get three similarity levels. In similarity 3, the data is divided into two big clusters G_1^2, G_2^2.

two sub-clusters in the similarity level 3. To this end, we define similarity degree s_{ij} to measure the similarity between two samples $\mathbf{z}_i$ and $\mathbf{z}_j$ as:

$$s_{ij} = L_k, \tag{4}$$

where $L_k \in \{1, 2, \cdots, K\}$ represents the similarity level, in which $\mathbf{z}_i$ and $\mathbf{z}_j$ are merged into the same sub-cluster.

3.4. Loss Function

Loss function plays a key role in many deep learning methods. However, most existing metric loss can only deal with binary supervision. By taking full advantage of the learned hierarchical labels in the previous subsection, we design a new log-ratio loss that aims to approximate the ratio of similarity degrees by the ratio of distances in the learned hyperbolic embedding space. Given triplet sample $\{\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_l\} \in \mathcal{S}$, the log-ratio loss can be defined as:

$$\mathcal{L}(i, j, l) = \left(\log \frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{\|\mathbf{z}_i - \mathbf{z}_l\|} - \log \Omega^{s_{ij} - s_{il}} \right)^2, \tag{5}$$

where $\Omega > 0$ is the hyperparameter to trade off the similarity degree. The loss function (5) contains two items: the first item is the log ratio of distance between sample pairs and the second item is the log ratio of corresponding similarity degree. The similarity between $\mathbf{z}_i$ and $\mathbf{z}_j$ is represented by $\Omega^{s_{ij}}$, where Ω denotes the base of similarity set manually. Through this new loss, the distance of sample pairs in hyperbolic embedding space will approximately equal to the ratio between their similarity distance. As shown in Figure 5, our proposed loss function can assign different distance thresholds to negative samples to make full use of richer similarity information. Algorithm 1 details the iterating procedure of our proposed method.

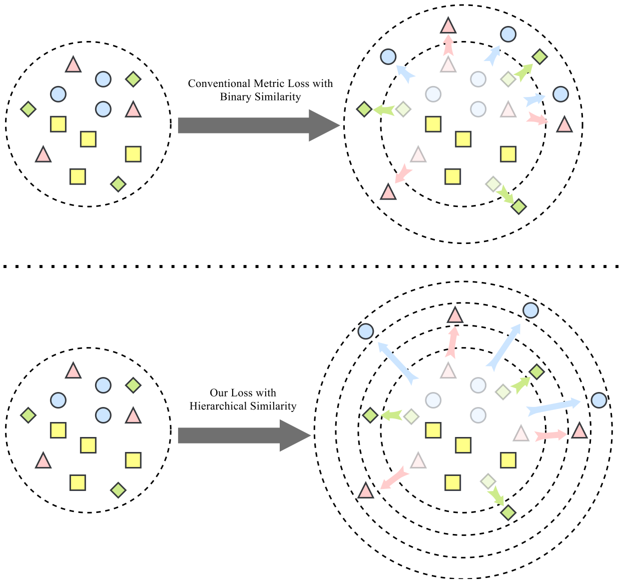
Conventional Metric Loss with Binary Similarity
Our Loss with Hierarchical Similarity

Figure 5. The comparison of our proposed loss (bottom) with conventional metric loss with binary similarity (top). Note that the size of circle is determined by parameter margin in conventional metric loss and hyperparameter Ω in our loss. Conventional metric loss can only push negative samples equally far, but our loss is able to push away negative samples with different margin guided by the intrinsic similarity level.

Algorithm 1 Unsupervised Hyperbolic Deep Metric Learning Algorithm

```
Input:
    Training dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$;
    Hyperparameter $\Omega$;
    The number of similarity level $K$;
    The number of epochs $N$.
Output:
    Best hyperbolic metric model $f(\mathbf{x}_i|\theta)$.
Pre-train and initialize the parameters $\theta$.
for $epoch = 1, 2, \cdots, N$ do
    Conduct hierarchical clustering in the hyperbolic metric space according to Eq. (3);
    Produce triplet inputs with the learned hierarchical similarity according to Eq. (4);
    Optimize the parameters $\theta$ using Eq. (5) in the hyperbolic DML module;
end for
return $\theta$.
```

3.5. Supervised extension of our method

Our method can be easily extended to supervised version. For better scalability, we define new adaptive hierarchical margin which can be integrated into any metric loss function. Given a training data $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ with labels $\mathcal{Y} = \{y_1, y_2, \cdots, y_n\}$, we first extract image features to form a hyperbolic metric space $\mathcal{Z} = \{\mathbf{z}_i = f(\mathbf{x_i}|\theta)\}_{i=1}^{n}$ through the hyperbolic metric learning module with binary similarity. Within labels, we can calculate the similarity distance between a-th class and b-th class

$$d_{ab} = \frac{1}{n_a n_b} \sum_{\mathbf{z}_i^a \in C_a, \mathbf{z}_j^b \in C_b} \|\mathbf{z}_i^a - \mathbf{z}_j^b\|, \tag{6}$$

and average inner similarity distance of a-th class

$$d_a = \frac{1}{n_a^2 - n_a} \sum_{\mathbf{z}_i^a, \mathbf{z}_j^a \in C_a} \|\mathbf{z}_i^a - \mathbf{z}_j^a\|, \tag{7}$$

where $\mathbf{z}_i^a$, $\mathbf{z}_j^b$ are samples in the cluster C_a, C_b respectively, and n_a, n_b represent the number of samples in C_a, C_b respectively.

Given an anchor $\mathbf{z}_i$, the positive margin with positive sample $\mathbf{z}_j$ is defined as

$$M_p = d_a + \gamma, \tag{8}$$

and the negative margin with negative sample $\mathbf{z}_l$ is represented as

$$M_n = d_{ab} - d_a + \gamma, \tag{9}$$

where γ is a constant parameter following [12].

4. Experiments

This section describes our setting for the experiment and reports the performance of our algorithm compared to existing methods. More experimental results on the supervised version of our method are given in supplementary materials.

4.1. Datasets

The proposed approach is tested on the following standard benchmark datasets for image retrieval. For the retrieval task, we use the standard performance metric Reall@K, *i.e*. computing the percentage of testing samples which have at least one example from the same category in K nearest neighbors. We do not use ground-truth labels in all experiments.

- **CARS196** [23] includes 16,185 images of 196 car types. 8,054 images in the first 98 classes are used for training, and 8,131 images in the remaining 98 classes are used for testing.
- **CUB-200-2011** [41] contains 11,788 images of 200 bird species. We use the first 100 classes (5,864 images) for training and the other 100 classes (5,924 images) for testing.
- **Stanford Online Products** (SOP) [30]: is composed of 12,053 images of 22,634 products from eBay.com. We use the first 11,318 products with 59,551 images and the other 11,316 products with 60,502 images for training and testing, respectively.

4.2. Implementation Details

We utilized the Pytorch deep learning package [31] in all experiments. Following the standard data pre-processing paradigm, we normalized the input images into 256×256 at first, and then cropped them to 227×227. For data augmentation, random cropping and random horizontal mirroring were performed before training. For the metric network, the GoogLeNet [38] pre-trained on ILSVRC 2012-CLS [33] was adopted as a backbone network for fair comparison. Moreover, on top of the network following the global pooling layer, a fully-connected layer was added with random initialization. We fixed the feature embedding size to 512 and and set the batch size to 80. Adam optimizer [22] is used in all experiments and the weigh decay is set to $1e^{-5}$. We set the curvature of Poincaré ball $\tau = 1$.

4.3. Performance Comparisons with State-of-the-art Methods

To evaluate the performance of our proposed method, we compare it with the state-of-the-art unsupervised methods on image retrieval tasks. Self-supervised transformed attention consistency method (denoted by TAC-CCL) [24], the invariant and spreading instance feature method (denoted by Instance) [49] and the mining on manifolds (MOM) [18] are current state-of-the-art methods for unsupervised metric learning. In addition three other methods introduced in Instance can be adopted for unsupervised DML task: DeepCluster [3], NCE (Noise-Contrastive Estimation) [47] and Examplar [8]. All of these methods use the GoogLeNet [38] as the backbone encoder. We include the results of these methods for comparisons.

Following the conventional paradigm of unsupervised DML, we adapt some classical DML methods such as Contrastive loss [5], Triplet loss [16] and Lifted Structure loss [30] from supervised metric learning to unsupervised metric learning using k-means clustering to assign pseudo labels. These methods can be regarded as baseline.

Table 1 and Table 3 present the experimental results of our proposed method and compared methods on the CUB, Cars, and SOP datasets. Note that bold numbers represent the results of our proposed method. It is obvious that our method significantly outperforms state-of-the-art unsupervised DML methods on all the datasets, which demonstrates the effectiveness of our proposed approach. Considering that TAC-CCL is a plug-and-play self-supervised module for DML that can be easily combined with our method, it is fairer to compare our method with TAC-CCL (baseline). On the Cars196 dataset, our method outperforms the current state-of-the-art TAC-CCL (baseline) method by 4.7% and even improves the Recall@8 by 3.6% compared with TAC-CCL. On the CUB dataset, our method improves the Recall@1 by 5.0% and is even competitive to some supervised metric learning methods. Compared with these meth-

Method	SOP		
	R@1	R@10	R@100
Baseline + CL [5]	58.6	74.3	86.7
Baseline + TL [16]	59.4	74.8	87.0
Baseline + LL [30]	60.3	74.2	86.9
Exampler	45.0	60.3	75.2
NCE	46.6	62.3	76.8
DeepCluster	34.6	53.6	66.8
MOM	43.3	57.2	73.2
Instance	48.9	64.0	78.0
TAC-CCL(Baseline)	62.5	76.5	87.2
TAC-CCL	63.9	77.6	87.8
Ours	**65.1**	**78.2**	**88.3**

Table 1. Recall@K performance on SOP in comparison with other methods.

ods only using binary similarity, our method can capture richer hierarchical similarity information, which boosts our performance in an unsupervised manner.

4.4. Ablation Studies

In this section, we conduct several ablation studies to demonstrate the effectiveness of different components in our proposed method.

4.4.1 Impact of different similarity levels

The proposed log-ratio loss is based on hierarchical clustering in the hyperbolic metric space. How to construct appropriate hierarchical similarity according to the results of hierarchical clustering is essential. Therefore, the number of similarity level K is a critical parameter for the proposed method. We conduct the following ablation experiment on the CUB data to study the impact of K. We select different K to derive corresponding hierarchical clusters as shown in Table 2. The results for Recall@1, 2, 4, 8 are presented in Figure 6. We can see that unsupervised metric learning performance increases with the number of similarity degree since it contains richer similarity with enhanced discriminative power. However, excessive hierarchical information may degrade the performance because the intrinsic structure of data is not so complicated.

4.4.2 Impact of different hyperparameter Ω

In our proposed log-ratio metric loss, we hope to make the ratio between the distance of sample pairs approximately equal to the ratio between their similarity distance. Therefore, assigning appropriate similarity distance to sample pairs guided by similarity degree is also very important in our method. In this ablation study, we discuss the impact of different Ω (*i.e.* the preset base of similarity) to model performance. For example, the hyperparameter Ω ranges from 2, 10, 100 to 1000. The results for Recall@1, 2, 4, 8 are presented in Figure 7. We can see that when $\Omega = 10$, our method achieves the best performance. It is shown that the distance ratio of sample pairs in adjacent similarity degree is approximate to 10 on CUB dataset.

K	hierarchical clusters
1	100
2	$100 \rightarrow 50$
3	$100 \rightarrow 75 \rightarrow 25$
4	$100 \rightarrow 70 \rightarrow 40 \rightarrow 10$

Table 2. The description of different K and corresponding hierarchical clusters.

4.4.3 Performance contributions analysis

In this ablation study, we aim to identify the contribution of each algorithm component on different datasets. Our proposed method contains three major components: hyperbolic layer for DML module, hierarchical clustering, and the new log-ratio-based metric loss with hierarchical similarity. In order to evaluate the effect of different components of the proposed method, we conduct unsupervised DML task on the CUB and CARS datasets using different method configurations: (1) Baseline with classical metric losses; (2) Baseline with classical metric losses + hyperbolic layer; (3) Our method. The experimental results are summarized in Table 4. It finds that both the hyperbolic geometry and the proposed new metric loss with hierarchical similarity significantly improve the performance of baseline.

5. Conclusion

We have proposed an Unsupervised Deep Hyperbolic Metric Learning method. Unlike existing works, our new method takes account into the hierarchical similarity among samples in modeling by virtue of Hyperbolic embedding and hierarchical clustering.In addition, we also presented a novel log-ratio loss function to utilize the hierarchical similarity supervision. We demonstrated that our method outperforms several state-of-the-art methods by a great margin on several standard benchmark datasets.

Acknowledgment

J.Y. and C.D. were supported in part by the National Natural Science Foundation of China under Grant 62071361, and the National Key R&D Program of China under Grant 2017YFE0104100.

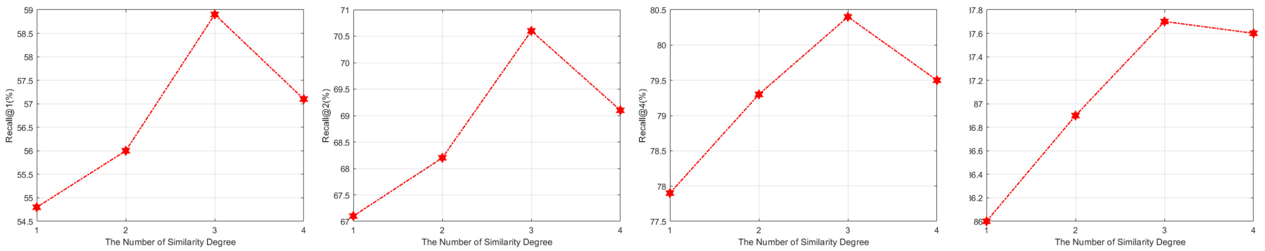

Figure 6. Recall@$k(\%)$ performance on CUB dataset in comparison with different number of similarity degree K.

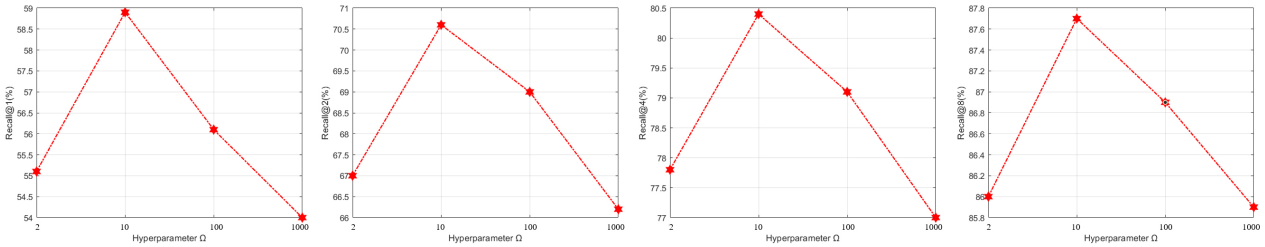

Figure 7. Recall@$k(\%)$ performance on CUB dataset in comparison with different number of Ω.

Method	CARS196				CUB-200-2011			
	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8
Baseline with Contrastive Loss [5]	33.3	44.2	55.5	67.2	48.8	62.4	74.1	84.2
Baseline with Triplet Loss [16]	37.9	49.3	61.4	72.9	50.7	62.4	73.4	82.8
Baseline with LiftedStructure Loss [30]	42.4	53.9	65.5	76.5	53.1	66.1	76.8	85.6
Exampler	36.5	48.1	59.2	71.0	38.2	50.3	62.8	75.0
NCE	37.5	48.7	59.8	71.5	39.2	51.4	63.7	75.8
DeepCluster	32.6	43.8	57.0	69.5	42.9	54.1	65.6	76.2
MOM	35.5	48.2	60.6	72.4	45.3	57.8	68.6	78.4
Instance	41.3	52.3	63.6	74.9	46.2	59.0	70.1	80.2
TAC-CCL(Baseline)	43.0	53.8	65.3	76.0	53.9	66.2	76.9	85.8
TAC-CCL	46.1	56.9	67.5	76.7	57.5	68.8	78.8	87.2
Ours	**47.7**	**58.9**	**70.3**	**80.3**	**58.9**	**70.6**	**80.4**	**87.7**

Table 3. Recall@K performance on CARS and CUB datasets in comparison with other methods.

Method	CARS196				CUB-200-2011			
	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8
Baseline with Contrastive Loss [5]	33.3	44.2	55.5	67.2	48.8	62.4	74.1	84.2
+ Hyperbolic Layer	43.7	55.	66.9	78.1	54.5	68.1	79.0	87.5
Baseline with Triplet Loss [16]	37.9	49.3	61.4	72.9	50.7	62.4	73.4	82.8
+ Hyperbolic Layer	44.7	56.4	67.6	78.3	55.1	67.9	78.3	86.5
Baseline with LiftedStructure Loss [30]	42.4	53.9	65.5	76.5	53.1	66.1	76.8	85.6
+ Hyperbolic Layer	44.3	55.5	66.4	77.5	55.1	66.9	78.0	86.1
Ours	**47.7**	**58.9**	**70.3**	**80.3**	**58.9**	**70.6**	**80.4**	**87.7**

Table 4. The performance of different components from our method on Cars and CUB datasets.

References

[1] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 2

[2] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *ECCV*, pages 730–746. Springer, 2016. 1

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 6

[4] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in neural information processing systems*, pages 4868–4879, 2019. 3

[5] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546, 2005. 1, 3, 6, 7, 8

[6] Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *CVPR*, pages 1153–1162, 2016. 1, 3

[7] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. Representation tradeoffs for hyperbolic embeddings. *ICML*, 80:4460, 2018. 3

[8] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *PAMI*, 38(9):1734–1747, 2015. 6

[9] Yueqi Duan, Lei Chen, Jiwen Lu, and Jie Zhou. Deep embedding learning with discriminative sampling policy. In *CVPR*, pages 4964–4973, 2019. 1

[10] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2780–2789, 2018. 1

[11] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in neural information processing systems*, pages 5345–5355, 2018. 3

[12] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *ECCV*, pages 269–285, 2018. 1, 3, 6

[13] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. In *ICML*, pages 498–505. IEEE, 2009. 1

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742. IEEE, 2006. 3

[15] Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2821–2829, 2017. 1, 3

[16] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015. 6, 7, 8

[17] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *ICML*, pages 2849–2858, 2019. 1

[18] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Mining on manifolds: Metric learning without labels. In *CVPR*, pages 7642–7651, 2018. 1, 6

[19] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *CVPR*, pages 6418–6428, 2020. 3

[20] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, pages 3238–3247, 2020. 1, 3

[21] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2288–2297, 2019. 2

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshop*, pages 554–561, 2013. 6

[24] Yang Li, Shichao Kan, and Zhihai He. Unsupervised deep metric learning with transformed attention consistency and contrastive clustering loss. *ECCV*, 2020. 1, 6

[25] Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. In *NIPS*, pages 12565–12576, 2019. 3

[26] Nicholas Monath, Manzil Zaheer, Daniel Silva, Andrew McCallum, and Amr Ahmed. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *KDD*, pages 714–722, 2019. 3

[27] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017. 3

[28] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, pages 6338–6347, 2017. 3, 4

[29] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *CVPR*, pages 5382–5390, 2017. 1

[30] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *ICCV*, pages 4004–4012, 2016. 1, 3, 6, 7, 8

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, pages 8026–8037, 2019. 6

[32] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *CVPR*, pages 6450–6458, 2019. 1, 3

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 6

[34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 1, 3

[35] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, pages 1857–1865, 2016. 1

[36] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *CVPR*, pages 7251–7259, 2019. 1, 3

[37] 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*, pages 6398–6407, 2020. 3

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 6

[39] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018. 3

[40] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, pages 4170–4178, 2016. 1, 3

[41] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[42] 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*, pages 5022–5030, 2019. 3

[43] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *CVPR*, pages 6388–6397, 2020. 3

[44] Kun Wei, Muli Yang, Hao Wang, Cheng Deng, and Xianglong Liu. Adversarial fine-grained composition learning for unseen attribute-object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3741–3749, 2019. 1

[45] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, pages 1473–1480, 2006. 1

[46] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *CVPR*, pages 2840–2848, 2017. 3

[47] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 6

[48] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016. 1

[49] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pages 6210–6219, 2019. 1, 6

[50] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *ICCV*, pages 814–823, 2017. 1

[51] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, pages 1058–1067, 2017. 1

[52] Liming Zhao, Xi Li, Yueting Zhuang, and Jingdong Wang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, pages 3219–3228, 2017. 1

[53] Jiahuan Zhou, Pei Yu, Wei Tang, and Ying Wu. Efficient online local metric adaptation via negative samples for person re-identification. In *ICCV*, pages 2420–2428, 2017. 1