Deep Metric Learning via Adaptive Learnable Assessment

Wenzhao Zheng[1,2,3], Jiwen Lu[1,2,3,*], Jie Zhou[1,2,3,4]
[1]Department of Automation, Tsinghua University, China
[2]State Key Lab of Intelligent Technologies and Systems, China
[3]Beijing National Research Center for Information Science and Technology, China
[4]Tsinghua Shenzhen International Graduate School, Tsinghua University, China
zhengwz18@mails.tsinghua.edu.cn; lujiwen@tsinghua.edu.cn; jzhou@tsinghua.edu.cn

Abstract

In this paper, we propose a deep metric learning via adaptive learnable assessment (DML-ALA) method for image retrieval and clustering, which aims to learn a sample assessment strategy to maximize the generalization of the trained metric. Unlike existing deep metric learning methods that usually utilize a fixed sampling strategy like hard negative mining, we propose a sequence-aware learnable assessor which re-weights each training example to train the metric towards good generalization. We formulate the learning of this assessor as a meta-learning problem, where we employ an episode-based training scheme and update the assessor at each iteration to adapt to the current model status. We construct each episode by sampling two subsets of disjoint labels to simulate the procedure of training and testing and use the performance of one-gradient-updated metric on the validation subset as the meta-objective of the assessor. Experimental results on the widely used CUB-200-2011, Cars196, and Stanford Online Products datasets demonstrate the effectiveness of the proposed approach.

1. Introduction

Developing an effective metric to measure similarities of examples is at the core of many computer vision tasks. Generally, the distance of two points can be represented as the Euclidean distance in an embedding space, and deep metric learning utilizes deep neural networks [15, 19, 32, 39] to learn discriminative embeddings of images, so that samples from the same class have similar representations while samples from different classes have dissimilar representations. Recently a variety of deep metric learning methods have been proposed in the literature and demonstrate great power in various tasks, such as person re-identification [3, 31, 45, 57], face recognition [16, 22, 30], image set classification [23] and image retrieval [20, 27, 36].

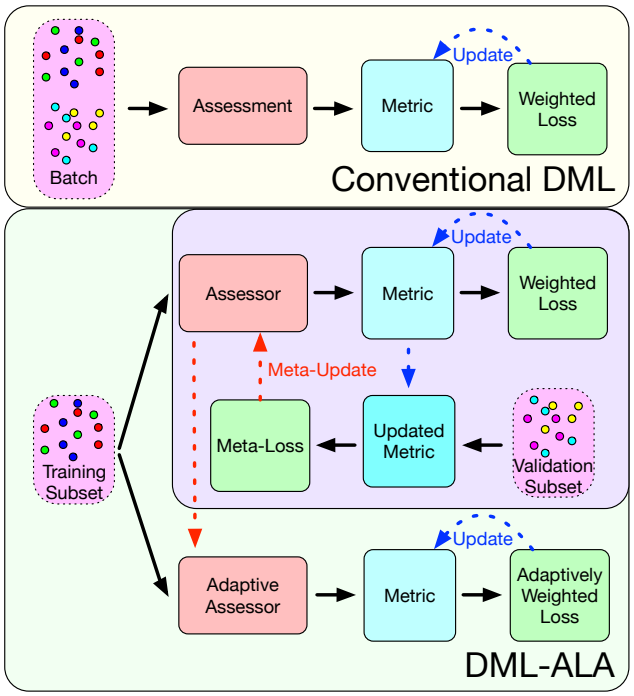


Figure 1. Flow-chart of our DML-ALA and comparisons with conventional deep metric learning (DML) methods. The proposed DML-ALA employs a simultaneously trained assessor to perform sampling instead of a hand-crafted sampling strategy. At each iteration, the training of our model consists of three stages: 1) updating the metric once using the weighted loss on the training subset, 2) training the assessor to maximize the performance of the *updated* metric on the validation subset, and 3) training the *original* metric using examples weighted by the trained assessor. Note that we only use the updated mehtric for the training of the assessor and discard it after each iteration.

Losses in metric learning are usually defined over two or more examples with a certain class structure called a "tuple". The number of m-tuples that can be formed from N examples has $O(N^m)$ complexity, rendering it inefficient to utilize all of them equally even for datasets of modest

*Corresponding author

sizes. There have been many studies exploring an efficient sampling strategy [10, 14, 17, 25, 30, 50]. Most existing methods utilize a hand-crafted sampling strategy, which is pre-defined based on some prior knowledge. However, during training the model is being updated constantly, and thus a fixed sampling strategy may not be effective at all stages. For example, the widely used hard mining strategy mines hard tuples that affect training the most, but it also ignores a variety of easy samples that might be helpful at the beginning [56]. This raises a natural question: *how to choose the appropriate sampling strategy at different training stages?*

In this work, we provide a positive solution to answer this question. We propose an adaptive learnable assessment (ALA) method which performs sampling adaptively to maximize the generalization ability of the trained metric, where the flow-chart is shown in Figure 1. The conventional hard mining strategy equals re-weighting each tuple by 0 or 1 according to a criterion regarding its hardness. We extend it by adopting a soft weighting scheme that generates weights between 0 and 1. Considering that inputs for the training of the metric are a sequence of tuples, we propose a sequence-aware assessor that is able to incorporate knowledge refined from previous inputs and the current model status. In addition, we argue that the success of existing deep metric learning methods has been impeded by overfitting, as verified in [43]. Inspired by this, we formulate the learning of the proposed assessor as a meta-learning problem with a meta-objective of maximizing the generalization. To achieve this, we employ an episode-based training scheme [42] and construct each episode with two subsets of disjoint labels to simulate the training set and test set partition. The metric trained in this manner works along with the assessor to seek the direction of good generalization. Experimental results on the CUB-200-2011 [44], Cars196 [18], and Stanford Online Products [36] datasets show that the proposed ALA improves the performance of existing methods in both image retrieval and clustering tasks.

2. Related Work

Deep Metric Learning: The training of a representative deep metric learning method involves two essential components: sampling and updating. There are two trends of recent progress regarding loss function and sampling strategy. The first trend of works designs different losses to consider various information buried beneath training samples [2, 6, 12, 30, 34, 36, 47, 48, 53]. For example, the triplet loss [5, 30, 46] pushes away the distance of a negative pair to be larger than that of a positive pair. Sohn [34] extended the triplet loss to an N-pair loss which pushes away N-1 negatives in an (N+1)-tuple all at once. Ustinova *et al.* [40] presented a histogram loss to punish the overlap between similarity distributions of positive and negative pairs.

The other trend of works aims at exploring for an effective sampling strategy to train the metric. The quality of samples used in the metric learning process has a crucial influence not only on the convergence speed of training but more importantly on the performance of the method. A widely used approach is the hard negative mining strategy [10, 14, 17, 30, 54], which essentially under-samples the training set for false positive samples that provide the most information. Hard mining may cause a distribution shift due to the under-sampling [52], motivating some works to consider other sampling frameworks to avoid sampling only the hard ones [7, 8, 10, 24, 25, 37, 50, 51, 55, 56]. For example, Wu *et al.* [50] proposed to select samples uniformly based on distances. Movshovitz *et al.* [25] proposed to use proxies to efficiently represent a set of samples, reducing the sampling complexity dramatically. Duan *et al.* [8] and Zhao *et al.* [55] trained a generator to synthesize hard samples in an adversarial manner. Zheng *et al.* [56] utilized linear interpolation to generate hardness-aware synthetics. However, all these methods utilize fixed pre-defined sampling strategies which assume some prior knowledge, and thus cannot flexibly adapt to the current model status.

Meta-Learning: Recently deep learning [15, 19, 32, 39] has shown great power and enabled machines to outperform humans in various tasks. The main issue hindering further development is the demand for numerous training data and massive computing resources. As an attempt to address this issue, meta-learning [1, 4, 9, 26, 29, 33, 38, 42] aims to learn a higher-level model (meta-learner) to instruct the learning process of the original model (learner) to adapt to new tasks rapidly. For example, Vinyals *et al.* [42] proposed an episode-based training strategy to simulate the process of one-shot learning, which is used to train a matching network to directly map a few labelled samples and an unlabelled sample to its label. Finn *et al.* [9] proposed a model-agnostic meta-learning algorithm to learn a set of initial parameters enabling a model to adapt to a new task quickly.

Inspired by recent works in meta-learning, we design a learnable assessor as the meta-learner and utilize it to train the metric adaptively to maximize the generalization ability. To achieve this, we employ an episode-based training scheme [42], where we construct each episode to simulate the procedure of training and testing. Different from most existing meta-learning methods, we train a meta-learner to perform sampling, which is shown in [50] to have a significant effect in deep metric learning.

3. Proposed Approach

In this section, we first introduce the basic ideas of deep metric learning and review the conventional hard mining sampling strategy. Then, we present our adaptive learnable assessor considering the sequential information of training examples. Finally, we propose an efficient approach to learn the assessor simultaneously by maximizing the generaliza-

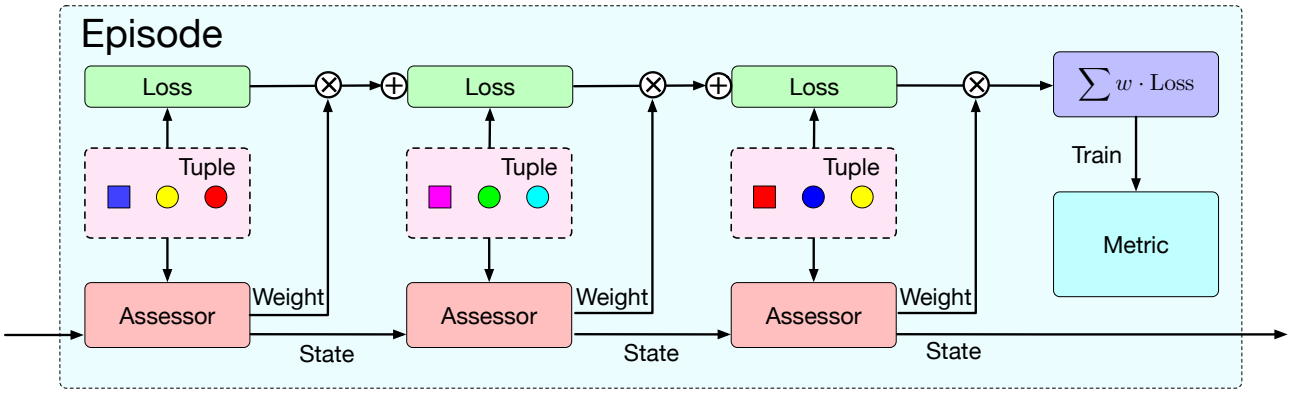


Figure 2. Illustration of the proposed sequence-aware learnable assessment. For each loss over a tuple, the assessor generates an adaptive weight combining information about this tuple's structure with the knowledge of previous inputs and current model status. To achieve this, a latent state is passed through the assessor over the whole training process, containing information learned from previous experience.

tion ability of the trained metric.

3.1. Problem Formulation

Suppose we have a set of samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ and their corresponding class labels $\mathbf{L} = [l_1, l_2, \cdots, l_N]$. The objective of deep metric learning is to learn an embedding function $f(x;\theta)$ which maps a sample from the original space to an n-dimensional embedding (metric) space so that in this space samples from the same class form a cluster far away from the other samples. More concretely, we measure the distance between two examples by computing the Euclidean distance between them in the embedding space:

$$D(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{y}_i, \mathbf{y}_j; \theta) = ||\mathbf{y}_i - \mathbf{y}_j||_2, \tag{1}$$

where $\mathbf{y} = f(\mathbf{x};\theta)$ is the learned embedding of $\mathbf{x}$. The objective of deep metric learning can be formulated as:

$$\min_{\theta} \begin{cases} d(\mathbf{y}_i, \mathbf{y}_j; \theta) & , \ if \ \ l_i = l_j \\ -d(\mathbf{y}_i, \mathbf{y}_j; \theta) & , \ if \ \ l_i \neq l_j. \end{cases} \tag{2}$$

Deep metric learning methods usually utilize a deep network as the embedding function $f(x;\theta)$, where θ represents the parameters of the network. The network is trained towards (2) by minimizing a well-designed loss function:

$$\theta^* = \arg\min_{\theta} \sum_{\mathbf{T} \in \boldsymbol{\mathcal{T}}} L(\mathbf{T}; f_\theta), \tag{3}$$

where $\mathbf{T} = \{\mathbf{y}_i\} \in \boldsymbol{\mathcal{T}}$ is a tuple composed of several examples with a certain class structure.

For example, the conventional triplet loss acts on a tuple of three samples (which is also called a triplet). A triplet $\mathbf{T} = \{\mathbf{y}, \mathbf{y}^+, \mathbf{y}^-\}$ is composed of an anchor point $\mathbf{y}$, a positive point $\mathbf{y}^+$ which is from the same class as the anchor, and a negative point $\mathbf{y}^-$ which is from a different class. The triplet loss aims at increasing the distance between the anchor and negative to be larger than the distance between the anchor and positive by a fixed margin m:

$$L(\mathbf{T}(\mathbf{y}, \mathbf{y}^+, \mathbf{y}^-)) = [d(\mathbf{y}, \mathbf{y}^+)^2 - d(\mathbf{y}, \mathbf{y}^-)^2 + m]_+, \tag{4}$$

where $[\cdot]_+ = max(\cdot, 0)$ is the hinge function.

Given N training samples, the set of triplets $\boldsymbol{\mathcal{T}}$ has the complexity size $O(N^3)$, making it inefficient to utilize all of them equally. A widely used technique is the hard mining strategy, which mines the hard triplets in a batch and ignores the easy ones since they provide little information for the network. One simple way to obtain a hard triplet $\mathbf{T}_{hard} = \{\mathbf{y}_h, \mathbf{y}_h^+, \mathbf{y}_h^-\}$ in a batch is to find a negative $\mathbf{y}_h^-$ with the smallest distance from the anchor $\mathbf{y}_h$:

$$\mathbf{y}_h^- = \arg\min_{\mathbf{y}_h^-} d(\mathbf{y}_h, \mathbf{y}_h^-). \tag{5}$$

We see from (4) and (5) that hard triplets lead to substantial loss and thus provide abundant information for training. The training of a network equipped with the hard mining strategy can be represented as:

$$\begin{aligned} \theta^* &= \arg\min_{\theta} \sum_{\mathbf{T} \in \boldsymbol{\mathcal{T}}_{hard}} L(\mathbf{T}; f_\theta) \\ &= \arg\min_{\theta} \sum_{\mathbf{T} \in \boldsymbol{\mathcal{T}}} \mathbb{1}_{\boldsymbol{\mathcal{T}}_{hard}}(\mathbf{T}) L(\mathbf{T}; f_\theta), \end{aligned} \tag{6}$$

where $\mathbb{1}_{\boldsymbol{\mathcal{T}}_{hard}}(\mathbf{T})$ is an indicator function which equals 1 when $\mathbf{T} \in \boldsymbol{\mathcal{T}}_{hard}$ and 0 otherwise.

3.2. Sequence-Aware Learnable Assessment

Suppose we randomly sample N tuples sequentially from the training set. We divide this sampled sequence $\mathbf{T}^N \in \boldsymbol{\mathcal{T}}^N$ into batches and use them to train the network by mini-batch gradient descent. The hard mining strategy can be seen as assigning a weight to each sample in the sequence, which equals 1 for hard tuples and 0 otherwise.

We go beyond the hard mining strategy and define a sample assessment strategy $\mathbf{S} \in \boldsymbol{\mathcal{S}}$ to be a mapping which maps a tuple sequence $\mathbf{T}^N \in \boldsymbol{\mathcal{T}}^N$ to a weight sequence $(w_1, w_2, \cdots, w_N) \in \boldsymbol{\mathcal{R}}^N$ where each $w_i \in (0, 1)$. We de-

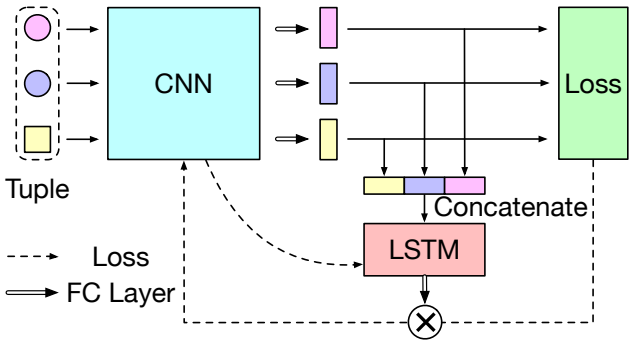


Figure 3. The network architecture of the proposed DML-ALA. We add a fully connected layer after a CNN network as the metric. The assessor is composed of an LSTM module and a fully connected layer. The embeddings of a tuple are concatenated and then taken as inputs of the assessor.

fine the training using assessment strategy $\mathbf{S}$ as:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} S_i(\mathbf{T}^N) L(\mathbf{T}_i; f_\theta), \tag{7}$$

where S_i denotes the ith output of sample assessment strategy $\mathbf{S}$ and $\mathbf{T}_i$ denotes the ith example in the sequence $\mathbf{T}^N$. We argue that $\boldsymbol{\mathcal{S}}$ includes a variety of sampling strategies. For example, we can represent the hard mining strategy as $\mathbf{S}^h(\mathbf{T}^N) = \{\mathbb{I}_{\boldsymbol{\mathcal{T}}_{hard}}(\mathbf{T_i})\} \in \boldsymbol{\mathcal{S}}$.

Most existing methods utilize hand-crafted sampling strategies, which usually assume some prior knowledge and cannot adapt to the model at different stages. For example, the hard mining strategy may be effective at the beginning, but the number of hard samples decreases as the training proceeds and little supervision can be further provided. Also, the under-sampling of the hard mining strategy may cause a distribution shift, harming the generalization ability.

To address this problem, we propose a sequence-aware learnable sample assessment strategy, which adaptively generates a weight for each tuple to best benefit training of the metric considering knowledge about the current model status, as shown in Figure 2. In practice, the tuple sequence $\mathbf{T}^N$ is usually generated progressively, so we do not see the whole sequence until the last step. We instead consider a subset of $\boldsymbol{\mathcal{S}}$ and define a learnable assessor A which takes as inputs a tuple $\mathbf{T}$ and a state variable $\mathbf{h}$, and outputs a real number $w \in (0, 1)$, i.e., $A(\mathbf{T}, \mathbf{h}; \phi) = w$, where ϕ is the parameters. We also assume that the assessor A determines a state transformation function $\mathbf{H}_A : \mathbf{h} \mapsto \mathbf{H}(\mathbf{h}, \mathbf{T}; \phi)$. The assessor A naturally induces a sample assessment strategy:

$$\mathbf{S}^A(\mathbf{T}^N) = \{A(\mathbf{T}_i, \mathbf{h}_{i-1}; \phi_i)\} \in \boldsymbol{\mathcal{S}}, \tag{8}$$

where $\mathbf{T}_i$ is the ith tuple in the sequence $\mathbf{T}^N$, $\mathbf{h}_{i-1} = \mathbf{H}(\mathbf{h}_{i-2}, \mathbf{T}_{i-1}; \phi_{i-1})$ is the state variable at step $i-1$, and ϕ_i is the parameters of assessor A at step i.

The state variable $\mathbf{h}$ encodes information from previous states, making the generated weights aware of the order of $\mathbf{T}^N$. It passes knowledge about previous input tuples and model status through training, enabling the assessor to interact with the metric. The assessor and the transformation function are also updated throughout training, capable of adapting to different training stages and model status.

We exploit a long short-term memory (LSTM) [11] network to integrate both the assessor and state transformation function. Having obtained a tuple of embeddings, we first concatenate them into a vector and use it as the input of the LSTM. At each step, the LSTM network takes in this concatenated vector and outputs a vector on the basis of a latent state cell which is simultaneously refined to incorporate knowledge learned from this step. We add a fully connected layer with a sigmoid activation function following the LSTM network to map the output vector to a real number $w \in (0, 1)$ as the assessed weight. The state variable is hidden inside the LSTM module, so in the context of a sequence $\mathbf{T}^N$, we can omit it from the assessor input for brevity (i.e., $w = A(\mathbf{T}; \phi)$). The training using assessor A can be represented as:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} A(\mathbf{T}_i; \phi) L(\mathbf{T}_i; f_\theta). \tag{9}$$

The proposed sequence-aware learnable assessor can preserve information from previous training process and exploit it to determine the current strategy. In addition, the assessor interacts with the metric and updates itself to produce adaptive weights that can best benefit the following training process. Figure 3 shows the network architecture of the proposed DML-ALA.

3.3. Adaptive Meta-Training of the Assessor

With a learnable assessor, we adaptively customize the training of the metric model. However, the learning of such a assessor is not trivial. Directly minimizing (9) with respect to ϕ leads to a trivial solution of $A(\mathbf{T}; \phi^*) = 0, \forall \mathbf{T} \in \boldsymbol{\mathcal{T}}$. We present an efficient meta-learning based approach to simultaneously learn the assessor in the training process by maximizing the generalization ability of the trained metric, as shown in Figure 1.

The assessor plays a vital role in the training process. It acts more like an optimizer for the metric, guiding the training direction. Furthermore, the assessor itself is learnable. The learning of the assessor is a learning problem at a higher level, which we formulate as a meta-learning problem.

The success of existing deep metric learning methods has been impeded by over-fitting. Real images usually vary widely in the aspects of background, illumination, pose, etc. However, intra-class variations are usually discouraged by the general objective of metric learning (2), leading to a metric with poor generalization ability.

Algorithm 1: DML-ALA

Input: Training image set, labels, learning rates α and β, episode size m, iteration number T, and iteration number for assessor at each episode K.

Output: Parameters of metric θ, and parameters of assessor ϕ.

```
for iter = 1, 2, ..., T do
    Construct an episode of m samples and form two sets of tuples {T}_tr and {T}_va.
    Perform one gradient update to θ and obtain θ' following (10).
    for iter = 1, 2, ..., K do
        Update assessor parameters φ following (12).
    end for
    Update metric parameters θ with the updated assessor parameters φ* following (13).
end for
return θ and φ.
```

This issue is hard to tackle by designing a loss function, which would probably be contradictory with (2). Instead, we propose to train an assessor to maximize the generalization ability of learned metric. We achieve this by exploiting the idea of episode-based training [42]. At each training iteration, we construct an episode by sampling two subsets of M and N examples with disjoint labels. We denote them as the training subset and validation subset. We then form two sets of tuples $\{\mathbf{T}\}_{tr}$ and $\{\mathbf{T}\}_{va}$ from the respective subsets.

We design one episode to simulate the procedure of training and testing. Our goal is to seek a sample assessment strategy to maximize the metric performance on the *validation* subset, *after* utilizing it to update the metric on the *training* subset. At each iteration, we first perform one gradient update to θ using (9) and obtain the updated parameters θ':

$$\begin{aligned}\theta' &= \theta - \alpha\nabla_\theta \sum_{\mathbf{T}\in\{\mathbf{T}\}_{tr}} A(\mathbf{T};\phi)L(\mathbf{T};f_\theta) \\ &= \theta - \alpha \sum_{\mathbf{T}\in\{\mathbf{T}\}_{tr}} A(\mathbf{T};\phi)\nabla_\theta L(\mathbf{T};f_\theta), \quad (10)\end{aligned}$$

where α is the learning rate of the metric.

We then evaluate the updated model on the *validation* subset and employ the validation loss to train the assessor. More concretely, the meta-training objective of the assessor can be represented as:

$$\begin{aligned}&\min_\phi \sum_{\mathbf{T}'\in\{\mathbf{T}\}_{va}} L(\mathbf{T}';f_{\theta'}) \quad (11)\\ =\ &\min_\phi \sum_{\mathbf{T}'\in\{\mathbf{T}\}_{va}} L\big(\mathbf{T}';f_{\theta-\alpha\sum_{\mathbf{T}\in\{\mathbf{T}\}_{tr}} A(\mathbf{T};\phi)\nabla_\theta L(\mathbf{T};f_\theta)}\big).\end{aligned}$$

Note that this loss is computed over the metric with the updated parameters θ' which is differentiable w.r.t. ϕ.

Ideally, we want to train the assessor A to minimize (11), but to improve the efficiency we only update it for a fixed times K. For each update:

$$\phi \leftarrow \phi - \beta\nabla_\phi \sum_{\mathbf{T}'\in\{\mathbf{T}\}_{va}} L(\mathbf{T}';f_{\theta'}), \quad (12)$$

where β is the meta learning rate of assessor A.

Finally, we update the *original* metric (i.e., f_θ, not $f_{\theta'}$) *once* using the updated assessor A_{ϕ^*}:

$$\theta \leftarrow \theta - \alpha \sum_{\mathbf{T}\in\{\mathbf{T}\}_{tr}} A(\mathbf{T};\phi^*)\nabla_\theta L(\mathbf{T};f_\theta), \quad (13)$$

and use it as the learned metric parameters at this iteration.

We only utilize updated model $f_{\theta'}$ to evaluate the generalization ability of the current optimizer (with assessor A_ϕ) and discard it after each iteration. The metric is optimized using (13) with the updated assessor A_{ϕ^*}, ensuring that the metric is always trained towards good generalization.

We sample each episode randomly from the training set, so the optimization of the metric and assessor can be performed using stochastic gradient descent (SGD). The metric and assessor are updated alternately at each iteration, but can be seen as being trained simultaneously across iterations throughout the whole process. The metric and assessor are coupled with each other, collaborating to seek a representation with good discrimination and generalization ability. Algorithm 1 details the proposed DML-ALA.

3.4. Implementation Details

We implemented our method using the Tensorflow package throughout the experiments. For fair comparisons with most deep metric learning methods, we employed the GoogLeNet [39] model pre-trained on ImageNet ILSVRC dataset [28] followed by a randomly initialized fully connected layer. We set the output embedding size of our method to 512. We implemented the assessor with a two-layer LSTM [11] model and a fully connected layer, where there are 64 hidden units in each layer. We normalized all the images to 256 by 256 as inputs. For training, we performed standard random cropping at 227 by 227 and horizontal random mirror for data augmentation. We set the base learning rate to 10^{-4} for the CNN, 10^{-3} for the last fully connected layer, and 4×10^{-4} for the assessor. At each iteration, we constructed an episode with a training subset of 100 samples and a validation subset of 20 samples and updated the assessor for 3 times. We tuned all the hyperparameters via cross-validation on the training set.

4. Experiments

In this section, we evaluated the proposed framework in both image retrieval and clustering tasks. We conducted

experiments on three widely used benchmark datasets, including the CUB-200-2011 [44], Cars196 [18], and Stanford Online Products [36] datasets.

4.1. Datasets

We followed [36] and evaluated our method under the setting where the training set is disjoint from the test set. We split each dataset into training/test set as described below:

- The CUB-200-2011 dataset [44] is composed of 11,788 images including 200 bird species. We split the images into a training set containing the first 100 species (5,864 images) and a test set containing the rest 100 species (5,924 images).
- The Cars196 dataset [18] is composed of 16,185 images of 196 car makes and models. We split the images into a training set containing the first 98 models (8,054 images) and a test set containing the rest 100 models (8,131 images).
- The Stanford Online Products dataset [36] is composed of 120,053 images of 22,634 online products from eBay.com. We split the images into a training set containing the first 11,318 products (59,551 images) and a test set containing the rest 11,316 products (60,502 images).

4.2. Evaluation Metrics

Following recent works [8, 35, 36] on deep metric learning, we conducted experiments in image retrieval and clustering tasks. We employed Recall@Ks to evaluate our method in the retrieval task, which computes the percentage of images with at least one correct retrieved example from the K nearest neighbors. We employed NMI and F_1 to evaluate our method in the clustering task. The normalized mutual information (NMI) is defined as the ratio of mutual information and the arithmetic mean of entropy of clusters and the ground truth classes, i.e., $\mathrm{NMI}(\Omega, \mathbb{C}) = \frac{2I(\Omega;\mathbb{C})}{H(\Omega)+H(\mathbb{C})}$, where $\Omega = \{\omega_1, \cdots, \omega_K\}$ is a set of clusters and $\mathbb{C} = \{c_1, \cdots, c_K\}$ is a set of ground truth classes. ω_i represents the set of samples assigned to the ith cluster, and c_j represents the set of samples belonging to the jth class. F_1 is defined as the harmonic mean of precision and recall, i.e., $\mathrm{F}_1 = \frac{2PR}{P+R}$.

4.3. Results and Analysis

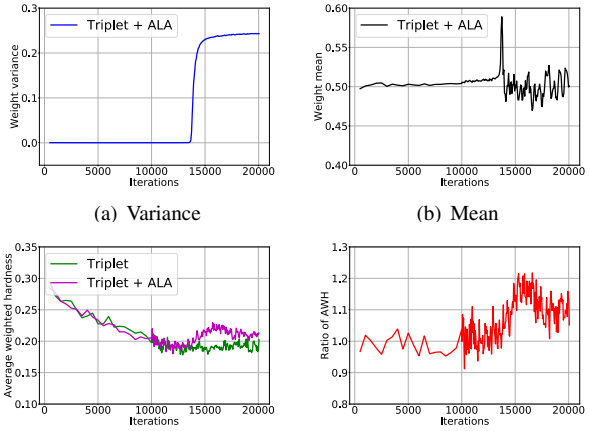
Effect of Episode Construction: We construct the training and validation subsets to simulate the procedure of training and testing so that we can evaluate the generalization ability of the metric. To study the effect of using disjoint labels, we performed an ablation study where both the original triplet loss and our method used random tuples.

Table 1. Results using different tuple settings on CUB-200-2011.

Method	NMI	F_1	R@1	R@2	R@4
Triplet (random)	48.3	14.5	34.7	47.0	58.3
ALA (random)	**56.6**	**25.5**	**44.4**	**58.4**	**70.9**
Triplet (disjoint)	49.8	15.0	35.9	47.7	59.1
ALA (disjoint)	**58.7**	**26.3**	**46.3**	**60.1**	**72.4**

Table 2. Results on the training and test set of CUB-200-2011.

Method	NMI	F_1	R@1	R@2	R@4
Triplet (training)	76.5	53.0	65.2	72.5	79.9
ALA (training)	**79.3**	**56.1**	**66.5**	**74.3**	**81.0**
Triplet (testing)	49.8	15.0	35.9	47.7	59.1
ALA (testing)	**58.7**	**26.3**	**46.3**	**60.1**	**72.4**



(a) Variance (b) Mean

(c) Hardness (d) Ratio

Figure 4. Weight analysis of ALA (triplet loss) on CUB-200-2011.

Table 1 shows that ALA using a random validation subset still boosts the performance of the original method, but with a smaller margin compared to that using disjoint tuples. The reason is that the assessor is less restricted due to joint labels and each episode cannot precisely simulate the training and test set partition. This illustrates that both the adaptive assessment and the use of disjoint subsets contribute to the performance improvement.

Alleviation of Overfitting: Table 2 shows the training and testing performance of the triplet loss with/without ALA on the CUB-200-2011 dataset. We see that with comparable training performance, our proposed ALA achieves much better results on the test set. This verifies that the proposed ALA can alleviate overfitting to some extent.

Analysis of Assessed Tuple Weights: We conducted experiments with the triplet loss on the CUB-200-2011 dataset to analyze the assessed tuple weights. Figures 4(a) and 4(b) show the weight variance and mean in each iteration. We observe that in the beginning our ALA treats all the samples almost equally, but learns to assign different weights as training proceeds. This suggests that the sampling strategy mainly influences the latter half of training, when further training of the model requires more challenging tuples.

Table 3. Comparisons with existing sampling methods on the CUB-200-2011 dataset.

Method	NMI	F_1	R@1	R@2	R@4	R@8
Rand-disjoint	49.8	15.0	35.9	47.7	59.1	70.0
Semi-hard	53.4	17.9	40.6	52.3	64.2	75.0
Smart mining	58.1	-	45.9	57.7	69.6	79.8
Dis-weighted	56.3	25.4	44.1	57.5	70.1	80.5
DAML	51.3	17.6	37.6	49.3	61.3	74.4
DVML	55.5	25.0	43.7	56.0	67.8	76.9
HDML	55.1	21.9	43.6	55.8	67.7	78.3
DE-DSP	53.7	19.8	41.0	53.2	64.8	-
ALA	**58.7**	**26.3**	**46.3**	**60.1**	**72.4**	**82.6**

Table 4. Comparisons with existing sampling methods on the Cars196 dataset.

Method	NMI	F_1	R@1	R@2	R@4	R@8
Rand-disjoint	52.9	17.9	45.1	57.4	69.7	79.2
Semi-hard	55.7	22.4	53.2	65.4	74.3	83.6
Smart mining	58.2	-	56.1	68.3	78.0	85.9
Dis-weighted	58.3	25.4	59.4	72.3	81.6	87.2
DAML	56.5	22.9	60.6	72.5	82.5	89.9
DVML	61.1	28.2	64.3	73.7	79.2	85.1
HDML	59.4	27.2	61.0	72.6	80.7	88.5
DE-DSP	55.0	22.3	59.3	71.3	81.3	-
ALA	**61.7**	**29.6**	**67.2**	**78.4**	**86.6**	**92.0**

Table 5. Comparisons with existing sampling methods on the Stanford Online Products dataset.

Method	NMI	F_1	R@1	R@10	R@100
Rand-disjoint	86.3	20.2	53.9	72.1	85.7
Semi-hard	86.7	22.1	57.8	75.3	88.1
Dis-weighted	87.9	23.4	58.9	77.2	89.6
DAML	87.1	22.3	58.1	75.0	88.0
DVML	89.0	31.1	66.5	82.3	91.8
HDML	87.2	22.5	58.5	75.5	88.3
DE-DSP	87.4	22.7	58.2	75.8	88.4
ALA	**89.7**	**35.4**	**68.6**	**83.1**	**91.9**

To show in one aspect what triplets our ALA assigns larger weights, we define the average weighted hardness (AWH) as $\frac{1}{n}\sum_{i=1}^{n} w_i \frac{d(\mathbf{y}_i,\mathbf{y}_i^+)}{d(\mathbf{y}_i,\mathbf{y}_i^-)}$, where $\frac{d(\mathbf{y}_i,\mathbf{y}_i^+)}{d(\mathbf{y}_i,\mathbf{y}_i^-)}$ is the ratio of distances between the positive and negative pair in each triplet, and w_i is the assessed weight. The AWH reflects the average hardness level of weighted tuples. Figure 4(c) shows the AWH of ALA and the original method in each iteration, and figure 4(d) shows the ratio of the two. We can see that the AWH tends to decrease, but ALA assigns larger weights to harder tuples as training proceeds to keep AWH at a high level. This is reasonable since it is beneficial to train the model with samples of increasing hardness levels [14, 56].

Comparisons with Existing Sampling Methods: We compared the proposed ALA with existing sampling methods, including random sampling of disjoint tuples, semi-hard negative mining [30], smart mining [14], distance-weighted sampling [50], DAML [8], DVML [21], HDML [56], and DE-DSP [7]. We equipped the widely used triplet loss with these sampling methods and conducted experiments under the same settings (e.g., CNN architecture and batch construction).

Table 6. Applications to various losses on CUB-200-2011.

Method	NMI	F_1	R@1	R@2	R@4	R@8
Lifted	56.4	22.6	46.9	59.8	71.2	81.5
Clustering	59.2	-	48.2	61.4	71.8	81.9
N-pair	60.2	28.2	51.9	64.3	74.9	83.2
Angular	61.0	30.2	53.6	65.0	75.3	83.7
Contrastive	47.2	12.5	27.2	36.3	49.8	62.1
Cont + ALA	**50.6**	**19.3**	**37.3**	**46.5**	**58.2**	**74.0**
Triplet	49.8	15.0	35.9	47.7	59.1	70.0
Triplet + ALA	**58.7**	**26.3**	**46.3**	**60.1**	**72.4**	**82.6**
Margin	58.7	26.6	49.6	62.7	74.1	82.9
Margin + ALA	**66.3**	**35.1**	**61.6**	**73.9**	**83.1**	**89.7**

Table 7. Applications to various losses on Cars196.

Method	NMI	F_1	R@1	R@2	R@4	R@8
Lifted	57.8	25.1	59.9	70.4	79.6	87.0
Clustering	59.0	-	58.1	70.6	80.3	87.8
N-pair	62.7	31.8	68.9	78.9	85.8	90.9
Angular	62.4	31.8	71.3	80.7	87.0	91.8
Contrastive	42.3	10.5	27.6	38.3	51.0	63.9
Cont + ALA	**43.7**	**12.5**	**36.3**	**48.2**	**60.1**	**73.2**
Triplet	52.9	17.9	45.1	57.4	69.7	79.2
Triplet + ALA	**61.7**	**29.6**	**67.2**	**78.4**	**86.6**	**92.0**
Margin	59.7	28.1	71.5	79.3	87.8	91.6
Margin + ALA	**68.5**	**38.4**	**80.5**	**87.9**	**92.8**	**95.9**

Tables 3, 4, and 5 show the experimental results of different sampling methods, where red numbers represent the best results. We observe that the proposed ALA outperforms existing sampling methods. The reason is that the other methods utilize a fixed strategy, while the proposed ALA can simultaneously learn an adaptive assessor to maximize the generalization ability of the learned metric. In particular, although DAML [8] and HDML [56] can exploit more potential of existing samples by generating synthetic samples as complements, our method still achieves better results. Furthermore, we emphasize that ALA does not conflict with these generative methods, which can be integrated to further boost the performance.

Applications to Various Losses: We applied the proposed ALA to three losses for direct comparisons, including contrastive loss [13], triplet loss [49], and margin loss [50]. We also compared our method with other four baseline losses, including lifted structure [36], clustering loss [35], N-pair loss [34], and angular loss [47].

We conducted all the experiments under the same settings for fair comparisons. In particular, for all baseline methods, we sample tuples of disjoint labels in each iteration as previous works (e.g., [47] for triplet loss). Therefore, our framework takes in batches of the same structure as baseline methods as shown in 1. The only difference is

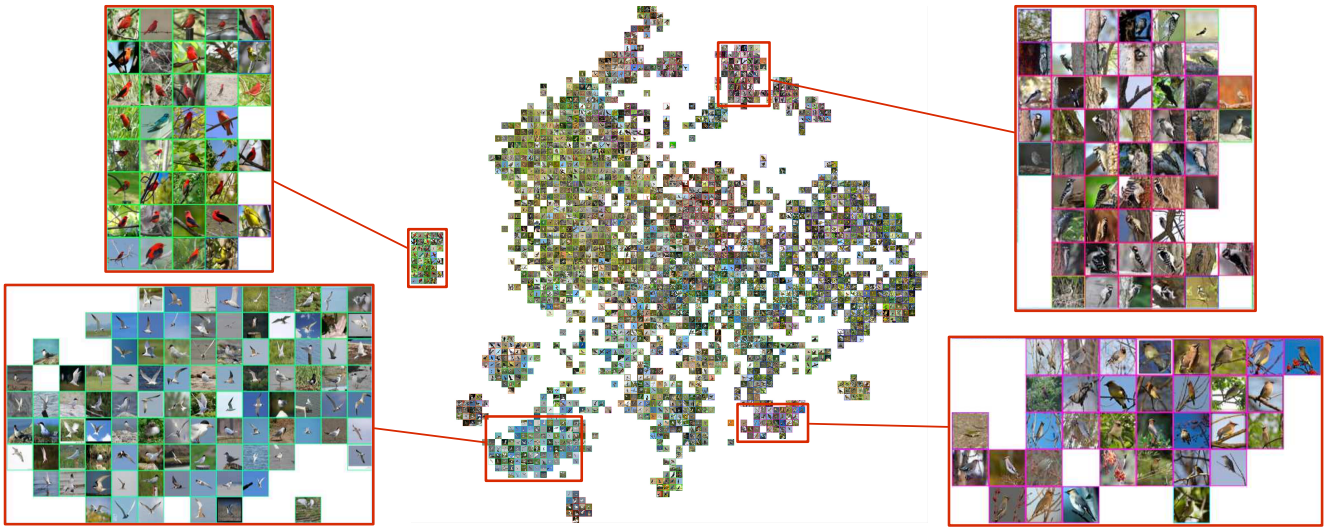


Figure 5. Barnes-Hut t-SNE visualization [41] of the learned embedding (Margin + ALA) on the CUB-200-2011 dataset.

Table 8. Applications to various losses on Stanford Online Products.

Method	NMI	F_1	R@1	R@10	R@100
Lifted	87.2	25.3	62.6	80.9	91.2
Clustering	89.5	-	67.0	83.7	92.2
N-pair	88.1	28.2	67.7	83.8	93.0
Angular	87.8	26.5	67.9	83.2	92.2
Contrastive	82.4	10.1	37.5	53.9	71.0
Cont + ALA	**84.3**	**12.5**	**42.9**	**58.5**	**74.1**
Triplet	86.3	20.2	53.9	72.1	85.7
Triplet + ALA	**89.7**	**35.4**	**68.6**	**83.1**	**91.9**
Margin	87.3	26.7	65.5	80.7	90.9
Margin + ALA	**91.0**	**39.3**	**77.0**	**89.4**	**96.1**

that ALA further splits each batch into two subsets. Also, we discard the temporary updated metric and only update the original metric once during each iteration.

Tables 6, 7, and 8 show the quantitative results on the CUB-200-2011, Cars196, and Stanford Online Products datasets respectively. We use red numbers to indicate the best results and bold numbers to represent an improvement over the corresponding original methods. We observe that our proposed ALA can uniformly boost the performance of original methods on all the three datasets. In particular, our framework combined with the margin loss outperforms the other baseline methods and achieves the best results. We see that the performance boost is relatively small on the large-sized Stanford Online Products dataset. We think this is because the large amount of training data alleviate the problem of over-fitting, which our method is designed to address.

For a training iteration, our framework (with the triplet loss) takes an average of 0.65s with one GTX 1080 Ti card, which is approximately twice as much as the original method takes (0.31s). However, we emphasize that our ALA is only used for more effective training and introduces no additional workload to computation during evaluation.

Qualitative Results: Figure 5 visualizes the embedding of our method (Margin + ALA) using Barnes-Hut t-SNE [41] on the CUB-200-2011 dataset. We represent the label of each image by the color of the border and enlarge several areas for clarity. We see that our method can effectively group semantically similar images, in spite of subtle visual cues and large variations of viewpoints, poses, etc.

5. Conclusion

In this paper, we have proposed a deep metric learning via adaptive learnable assessment (DML-ALA) method to maximize the generalization ability of the learned metric. By utilizing an episode-based training scheme, we can simulate the procedure of training and testing at one iteration, where we simultaneously train a sequence-aware assessor to instruct the learning process adaptively. Experimental results on three widely used benchmarks have shown that the proposed ALA outperforms existing sampling methods and improves current deep metric learning methods in both image retrieval and clustering tasks. While the proposed sample assessment method is designed for deep metric learning, it can be easily modified to apply to other machine learning approaches where sampling is a vital component.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 61822603, Grant U1813218, Grant U1713214, and Grant 61672306, in part by the Shenzhen Fundamental Research Fund (Subject Arrangement) under Grant JCYJ20170412170602564, and in part by Tsinghua University Initiative Scientfic Research Program.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, pages 3981–3989, 2016.

[2] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, pages 1861–1870, 2019.

[3] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *CVPR*, pages 1320–329, 2017.

[4] Yutian Chen, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando Freitas. Learning to learn without gradient descent by gradient descent. In *ICML*, pages 748–756, 2017.

[5] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, pages 1335–1344, 2016.

[6] Thanh-Toan Do, Toan Tran, Ian Reid, Vijay Kumar, Tuan Hoang, and Gustavo Carneiro. A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning. In *CVPR*, pages 10404–10413, 2019.

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

[8] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *CVPR*, pages 2780–2789, 2018.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[10] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R Scott. Deep metric learning with hierarchical triplet loss. In *ECCV*, pages 269–285, 2018.

[11] Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12:2451–2471, 2000.

[12] Soumyadeep Ghosh, Richa Singh, and Mayank Vatsa. On learning density aware embeddings. In *CVPR*, pages 4884–4892, 2019.

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

[14] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *ICCV*, pages 2840–2848, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[16] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, pages 1875–1882, 2014.

[17] Chen Huang, Chen Change Loy, and Xiaoou Tang. Local similarity-aware deep feature embedding. In *NIPS*, pages 1262–1270, 2016.

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

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[20] Marc T Law, Raquel Urtasun, and Richard S Zemel. Deep spectral clustering learning. In *ICML*, pages 1985–1994, 2017.

[21] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *ECCV*, pages 689–704, 2018.

[22] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, pages 6738–6746, 2017.

[23] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou. Multi-manifold deep metric learning for image set classification. In *CVPR*, pages 1137–1145, 2015.

[24] Jing Lu, Chaofan Xu, Wei Zhang, Ling-Yu Duan, and Tao Mei. Sampling wisely: Deep image embedding by top-k precision optimization. In *ICCV*, pages 7961–7970, 2019.

[25] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368, 2017.

[26] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, pages 2554–2563, 2017.

[27] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Bier - boosting independent embeddings robustly. In *ICCV*, pages 5189–5198, 2017.

[28] 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. *IJCV*, 115(3):211–252, 2015.

[29] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016.

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

[31] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li. Embedding deep metric for person re-identification: A study against large variations. In *ECCV*, pages 732–748, 2016.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, abs/1409.1556, 2014.

[33] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.

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

[35] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *CVPR*, pages 2206–2214, 2017.

[36] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016.

[37] 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.

[38] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

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

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

[41] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *JMLR*, 15(1):3221–3245, 2014.

[42] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

[43] Nam Vo and James Hays. Generalization in metric learning: Should the embedding layer be the embedding layer? *arXiv*, abs/1803.03310, 2018.

[44] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[45] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *CVPR*, pages 1288–1296, 2016.

[46] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014.

[47] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *ICCV*, pages 2593–2601, 2017.

[48] 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.

[49] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10(2):207–244, 2009.

[50] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, pages 2859–2867, 2017.

[51] Xinyi Xu, Yanhua Yang, Cheng Deng, and Feng Zheng. Deep asymmetric metric learning via rich relationship mining. In *CVPR*, pages 4076–4085, 2019.

[52] Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. Correcting the triplet selection bias for triplet loss. In *ECCV*, pages 71–87, 2018.

[53] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *ICCV*, pages 6490–6499, 2019.

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

[55] Yiru Zhao, Zhongming Jin, Guo-jun Qi, Hongtao Lu, and Xian-sheng Hua. An adversarial approach to hard triplet generation. In *ECCV*, pages 501–517, 2018.

[56] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *CVPR*, pages 72–81, 2019.

[57] 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.