A UNIFIED VIEW OF DEEP METRIC LEARNING VIA GRADIENT ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Loss functions play a pivotal role in deep metric learning (DML). A large variety of loss functions have been proposed in DML recently. However, it remains difficult to answer this question: what are the intrinsic differences among these loss functions? This paper answers this question by proposing a unified perspective to rethink deep metric loss functions. We show theoretically that most DML methods in deep metric learning, in view of *gradient equivalence*, are essentially weight assignment strategies of training pairs. Based on this unified view, we revisit several typical DML methods and disclose their hidden drawbacks. Moreover, we point out the key components of an effective DML approach which drives us to propose our weight assignment framework. We evaluate our method on image retrieval tasks, and show that it outperforms the state-of-the-art DML approaches by a significant margin on the CUB-200-2011, Cars-196, Stanford Online Products and In-Shop Clothes Retrieval datasets.

1 INTRODUCTION

Deep metric learning (DML) approaches learn to project images to a discriminative embedding space via deep neural networks. The embedded vectors of similar samples are closer while that of dissimilar samples are further. We call a method *pair-based* when its loss function can be expressed in terms of pairwise cosine similarities [1]. Most DML methods belong to this category, such as contrastive (Hadsell et al. (2006); Sohn (2016)), triplet (Hoffer & Ailon (2015)), quadruplet (Law et al. (2013)), lifted structure (Oh Song et al. (2016)), N-pairs (Sohn (2016)), binomial deviance (Yi et al. (2014)), histogram (Ustinova & Lempitsky (2016)) and angular (Wang et al. (2017)) losses.

One critical problem of these pair-based methods is how to discover and harness informative pairs, especially hard negative pairs because they pose two challenges for metric learning: (1) The number of negative pairs is second polynomial to the dataset size, thus utilizing all of the them is time-consuming and infeasible. (2) As we will show in our experiment section, training with numerous easy pairs causes performance degradation if involving these easy instances in the training process.

Existing DML methods propose different strategies to mitigate this problem. Contrastive loss neglects the negative pairs whose dot products are below a given threshold, while triplet loss only utilizes the negative pairs whose similarities are higher than that of the positive minus a fixed margin. However, contrastive is too absolute while triplet is too sensitive. Both cannot make full use of all the informative pairwise relations as stated in Hoffer & Ailon (2015); Wu et al. (2017). In contrast to employing a fixed margin, some other pair-based methods with log-exp formulation, like binomial deviance and N-pairs losses, solve this problem in a relative subtle way which will be detailed in Section 4.

In this paper, we establish a unified view of deep metric learning by theoretically proving all pair-based methods are equivalent to weight assignment strategies (Section 3). Thus, the intrinsic difference of these pair-based methods is that they assign pairs different weight coefficients, which leads us to rethink existing DML methods and discover their concrete pros and cons.

To seek an effective DML method, we posit two key factors of an suitable weight assignment (Section 5): (1) Assigning zero weight coefficients to easy pairs, since such pairs are already well taken

[1] For simiplicity, we use cosine similarity instead of euclidean distance, since we assume the embedding vector is L_2 normalized.

care of by the current model. Assigning easy pairs with nonzero weight may lead performance degradation. (2) Assigning weight to a pair by considering its absolute similarity and relative similarity compared with other pairs sharing the same anchor.

To the best of our knowledge, none of existing methods satisfy these two factors simultaneously. Instead, we propose a novel weight assignment strategy (Section 5.2) to solve this problem. It distinguishes the uninformative pairs through valid triplet based hard mining (VTHM) and assigns suitable weights to pairs via relative and absolute similarity based weight assignment (RAW).

To demonstrate the effectiveness of our proposed unified weight assignment framework, we conduct experiments on CUB-200-2011 (Wah et al. (2011)), Cars-196 (Krause et al. (2013)), Stanford Online Product (Oh Song et al. (2016)), and In-Shop Clothes Retrieval (Liu et al. (2016b)) datasets for the task of image retrieval (Section 6). Experimental results show that our framework improves the state-of-the-art performance by a large margin. Moreover, existing methods, such as binomial deviance and lifted structure loss, can also be improved when they get rid of the side effect of easy samples guided by our framework.

2 Related Work

Pair-based DML. *Siamese* network (Hadsell et al. (2006)) that learns embedding via contrastive loss, is one of the first pair-based DML methods. It pulls positive pairs as close as possible, while keeps negative pairs farther than a give distance. Triplet loss (Hoffer & Ailon (2015)) is based on triplets that consists of one positive pair and one negative pair sharing the same anchor point and targets learn an embedding space where the similarity of the negative pair is lower than that of the positive pair by a given margin. Inspired by triplet loss, methods using quadruplets emerged, e.g., PDDM (Law et al. (2013)) and histogram loss (Ustinova & Lempitsky (2016)).

Oh Song et al. (2016) argue that contrastive and triplet losses have not exploited all the pairwise relations of samples in one mini-batch, and propose lifted structure loss to utilize all the pairwise relations. However, it only subsamples approximately equal number of negative pairs of examples as positive randomly, thus, abandons a large number of informative negative pairs. Yi et al. (2014) propose binomial deviance loss using binomial deviance to evaluate the cost between labels and similarities, and pay more attention on hard pairs.

Hard Mining and Sampling The importance of hard mining in DML has been realized recently (Schroff et al. (2015); Harwood et al. (2017); Wu et al. (2017); Ge et al. (2018)), since most pairs, especially negative pairs, are uninformative and cannot boost the model further. Schroff et al. (2015) propose semi-hard mining, which only uses semi-hard triplets whose negative pair is farther than its positive pair. Such valid semi-hard triplets are scarce thus semi-hard mining needs a large batch-size, 1800 in the paper, to seek informative pairs. Harwood et al. (2017) provide a framework named smart mining to search hard samples from the whole dataset, which suffers from off-line computation burden. HTL (Ge et al. (2018)) builds a hierarchal tree of all the classes to find hard negative pairs. Wu et al. (2017) discuss the importance of sampling, and propose a sampling approach named *distance weighted sampling* which sample negative examples uniformly according to similarity.

Compared with these methods above only focusing on sampling, we propose a more generalized view: weight assignment view, and sampling is a special category of weight assignment whose weights are 0 or 1. Unlike some heuristically designed pair-based methods, our DML method is *carefully designed* driven by the instruction of the unified weight assignment view.

3 A Unified Weight Assignment View

Notations. Let $\boldsymbol{x}_i \in \mathbb{R}^{d\times 1}$ be a real valued instance vector and $y_i \in \{1, 2 \ldots, C\}$ denote the label of $\boldsymbol{x}_i$. Then we have the instance matrix $\boldsymbol{X} \in \mathbb{R}^{m\times d}$ and the label vector $\boldsymbol{y} \in \{1, 2, \ldots, C\}^m$ for m training samples. An instance $\boldsymbol{x}_i$ is projected to a unit sphere in an l dimension space by $f(\cdot; \boldsymbol{\theta}) : \mathbb{R}^d \to S^{l-1}$, where f is a learned function (a neural network) parameterized by $\boldsymbol{\theta}$.

In this paper, we use the cosine similarity $s_{i,j} =< f(\boldsymbol{x}_i; \boldsymbol{\theta}), f(\boldsymbol{x}_j; \boldsymbol{\theta}) >$ to measure the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, resulting an $m \times m$ similarity matrix $\boldsymbol{S}$ whose element at (i, j) is $s_{i,j}$. Furthermore, P_i (N_i) denotes the index set of positive (negative) samples of the anchor point $\boldsymbol{x}_i$ (i.e.,

$P_i = \{j|y_j = y_i \wedge j \neq i\}$, and $N_i = \{j|y_j \neq y_i\}$). Given a *pair-based* DML loss $\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})$, it can be expressed as a function in terms of $\boldsymbol{S}$: $\mathcal{H}(\boldsymbol{S}, \boldsymbol{y})$ according to the definition of pair-based.

To better understand our theorem, we define *gradient equivalence* as following:

Definition 3.1. At the t-th iteration, given two loss functions $\mathcal{F}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})$ and $\mathcal{G}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})$, if

$$\left.\frac{\partial \mathcal{F}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_t = \left.\frac{\partial \mathcal{G}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_t, \tag{1}$$

where $\left.\frac{\partial \mathcal{F}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_t$ is the partial gradient of $\mathcal{F}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ at $\boldsymbol{\theta}_t$, then we call $\mathcal{F}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})$ and $\mathcal{G}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})$ are *gradient equivalent* at the t-th iteration.

Gradient equivalence indicates these two functions have equal derivatives w.r.t. the model parameters at the t-th iteration, thus two gradient equivalent loss functions are the same for model training with gradient-based optimizer such SGD and Adam. Optimizing one loss function is equivalent to optimizing the other.

Theorem 3.1. For any pair-based loss $\mathcal{L}(\boldsymbol{X}, \boldsymbol{y}; \boldsymbol{\theta})$, there exists a loss function gradient equivalent to $\mathcal{L}$ with the following formulation:

$$\mathcal{F}(\boldsymbol{S}, \boldsymbol{y}) = \sum_{i=1}^{m} \left(\sum_{k \in N_i} w_{i,k} s_{i,k} - \sum_{j \in P_i} w_{i,j} s_{i,j} \right), \tag{2}$$

where $w_{i,j} \geq 0$ denoting the weight assigned to pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$.

Proof. Since $\mathcal{L}$ is pair-based, $\mathcal{L}$ can be expressed as a function of $\boldsymbol{S}$: $\mathcal{H}(\boldsymbol{S}, \boldsymbol{y})$. Thus the partial gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{\theta}$ at t-th iteration is:

$$\begin{aligned}
\left.\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\right|_t &= \sum_{i=1}^{m} \sum_{j>i}^{m} \left(\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,j}} \frac{\partial s_{i,j}}{\partial \boldsymbol{\theta}} \right) \Bigg|_t = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,j}}\right|_t \left.\frac{\partial s_{i,j}}{\partial \boldsymbol{\theta}}\right|_t \\
&= \frac{1}{2} \sum_{i=1}^{m} \left(\sum_{k \in N_i} \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,k}}\right|_t \left.\frac{\partial s_{i,k}}{\partial \boldsymbol{\theta}}\right|_t + \sum_{j \in P_i} \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,j}}\right|_t \left.\frac{\partial s_{i,j}}{\partial \boldsymbol{\theta}}\right|_t \right) \\
&= \sum_{i=1}^{m} \left(\sum_{k \in N_i} \frac{1}{2}(1 - 2I_{i,k}) \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,k}}\right|_t \left.\frac{\partial s_{i,k}}{\partial \boldsymbol{\theta}}\right|_t - \sum_{j \in P_i} \frac{1}{2}(1 - 2I_{i,j}) \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,j}}\right|_t \left.\frac{\partial s_{i,j}}{\partial \boldsymbol{\theta}}\right|_t \right),
\end{aligned} \tag{3}$$

where $I_{i,j} = 1$ when $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$ is a positive pair, otherwise 0.

Let $w_{i,j} = \frac{1}{2}(1 - 2I_{i,j}) \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,j}}\right|_t$, then in Equation (2),

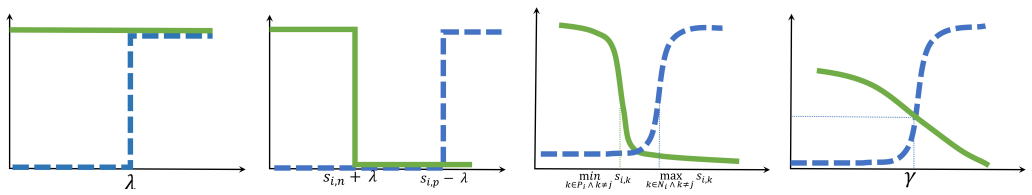
$$\mathcal{F}(\boldsymbol{S}, \boldsymbol{y}) = \sum_{i=1}^{m} \left(\sum_{k \in N_i} \frac{1}{2}(1 - 2I_{i,k}) \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,k}}\right|_t s_{i,k} - \sum_{j \in P_i} \frac{1}{2}(1 - 2I_{i,j}) \left.\frac{\partial \mathcal{H}(\boldsymbol{S}, \boldsymbol{y})}{\partial s_{i,j}}\right|_t s_{i,j} \right) \tag{4}$$

is gradient equivalent to $\mathcal{L}$. □

Theorem 3.1 unifies all existing pair-based approaches as weight assignment scheme and sheds light on the dark side of them. In the next section, we unveil their vital drawbacks that are undetectable if only analyzing their loss functions by studying their respective weight assignments.

4 Rethinking Existing Methods

In this section, we revisit several classic pair-based DML loss functions: contrastive, triplet, binomial deviance and lifted structure losses in the weight assignment scenario.



(a) Contrastive loss (b) Triplet loss (c) Lifted structure loss (d) Binomial deviance loss

Figure 1: Weight coefficient vs. pairwise cosine similarity. The solid green lines show the weight assigned for positive pairs, the dotted blue for negative pairs.

Contrastive Loss. Hadsell et al. (2006); Chopra et al. (2005) propose Siamese network, which encourages positive pairs to be closer, and negative pairs to be further when their cosine similarity is higher than a fixed threshold. Its loss function formulates as below:

$$\mathcal{L}_{contrast} := (1 - I_{i,j})[s_{i,j} - \lambda]_+ - I_{i,j} s_{i,j}, \tag{5}$$

where λ is the threshold. According to Equation (4), its weight assignment can be formulated as below:

$$w_{i,j} = I_A(\{\boldsymbol{x}_i, \boldsymbol{x}_j\}), \tag{6}$$

where I_A is indicator function of a subset $A = \{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} | j \in P_i\} \bigcup \{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} | s_{i,j} > \lambda \wedge j \in N_i\}$. The curves of weight assigned to pairs vs. similarities are shown in Figure (1a). The weight of any positive pair is 1. The weight of a negative pair is 1 if its similarity is higher than λ, otherwise is 0.

From the weight assignment, we find one drawback of contrastive loss that it treats all selected pairs equally, while neglecting their different hardness.

Triplet Loss. Hoffer & Ailon (2015) learn a discriminative model based on a triplet, which consists of an anchor $\boldsymbol{x}_i$, a positive instance $\boldsymbol{x}_p$ and negative instance $\boldsymbol{x}_n$:

$$\mathcal{L}_{triplet} := [s_{i,n} - s_{i,p} + \lambda]_+, \tag{7}$$

where λ is the given margin. Weight assignment of triplet loss is:

$$w_{i,j} = I_B(\{\boldsymbol{x}_i, \boldsymbol{x}_j\}), \tag{8}$$

where I_B is indicator function of a subset $B = \{\{\boldsymbol{x}_i, \boldsymbol{x}_p\} | s_{i,n} - s_{i,p} + \lambda > 0\} \bigcup \{\{\boldsymbol{x}_i, \boldsymbol{x}_n\} | s_{i,n} - s_{i,p} + \lambda > 0\}$. The weight curves of triplet loss are exhibited in Figure (1b), where we find only pairs in B are assigned with weight 1.

Two problems come to light when we rethink triplet approach from our unified view: One is the same with contrastive loss in assigning equal weights to selected pairs without considering their different hardness. Moreover, for triplet loss, whether a positive pair is selected into B only depends on *one randomly sampled* negative pair, and vice versa. This makes triplet loss tend to miss large amount of informative pairs and to be unstable during the training process.

Lifted Structure Loss. Oh Song et al. (2016) propose lifted structure loss whose objective is to exploit all the pairwise distances in a mini-batch, but the original version of lifted structure approach has not utilized all the negative pairs of one anchor. Hermans* et al. (2017) put forward a generalized form of lifted structure loss which leverages all anchor-positive and anchor-negative pairs as following:

$$\mathcal{L}_{lifted} := \frac{1}{m} \sum_{i=1}^{m} \left\{ \log \left[\sum_{k \in P_i} \exp(-s_{i,k}) \right] - \log \left[\sum_{k \in N_i} \exp(\lambda - s_{i,k}) \right] \right\}_+ \tag{9}$$

When the hinge function w.r.t. $\boldsymbol{x}_i$ in above equation returns nonzero value, we get its weight $w_{i,j}$ assignment by calculating the gradient of $\mathcal{L}_{lifted}$ w.r.t. $s_{i,j}$ and $s_{i,k}$:

$$w_{i,j} = \frac{\exp(-s_{i,j})}{\sum_{k \in P_i} \exp(-s_{i,k})} = \frac{1}{\sum_{k \in P_i} \exp(s_{i,j} - s_{i,k})}, \qquad if \quad j \in P_i \tag{10}$$

$$w_{i,j} = \frac{\exp(s_{i,j})}{\sum_{k \in N_i} \exp(s_{i,k})} = \frac{1}{\sum_{k \in N_i} \exp(s_{i,j} - s_{i,k})}, \qquad if \quad j \in N_i \tag{11}$$

When the hinge function returns zero value, the weights assigned to all involved pairs are zeros.

Its weight distribution is illustrated in Figure (1c). From Equation (11) and Figure (1c), we find that lifted structure has important improvement compared to contrastive and triplet approaches: It assigns pairs with different weight coefficients based on their similarities, while contrastive and triplet methods only think of pairs or triplets at instance level and assign pairs with 0 or 1 weight values. However, lifted structure is still confronted with two drawbacks from our weight assignment perspective: (1) It evaluates the overall state of all pairs with the same anchor to determine whether assigning all these pairs with zero or nonzero weights. In consequence, it ignores some informative pairs or assigns nonzero weights to numerous uninformative pairs. (2) In Equation (11), the weight of one pair only depends on its relative hardness $(s_{i,j} - s_{i,k})$ compared with other pairs, while neglects its *absolute similarity*.

Binomial Deviance Loss. Yi et al. (2014) propose the binomial deviance approach, whose loss function is:

$$\mathcal{L}_{binomial} \coloneqq \sum_{i=1}^{m} \left\{ \frac{1}{|P_i|} \sum_{k \in P_i} \log\left[1 + \exp\left(\alpha(\gamma - s_{i,k})\right)\right] + \frac{1}{|N_i|} \sum_{k \in N_i} \log\left[1 + \exp\left(\beta(s_{i,k} - \gamma)\right)\right] \right\}, \quad (12)$$

where γ is a hyper-parameter which serves as a soft threshold similar as λ in contrastive loss, α and β represent the sensitivities of positive and negative pairs to the threshold γ, respectively.

Its weight assignment strategy follows:

$$w_{i,j} = \frac{1}{|P_i|} \frac{\alpha \exp\left(\alpha\left(\gamma - s_{i,j}\right)\right)}{1 + \exp\left(\alpha\left(\gamma - s_{i,j}\right)\right)}, \qquad if \quad j \in P_i \quad (13)$$

$$w_{i,j} = \frac{1}{|N_i|} \frac{\beta \exp\left(\beta\left(s_{i,j} - \gamma\right)\right)}{1 + \exp\left(\beta\left(s_{i,j} - \gamma\right)\right)}, \qquad if \quad j \in N_i \quad (14)$$

We display its weight distribution in Figure (1d). Binomial deviance approach utilizes the sigmoid function to replace the unit step function in contrastive loss, thus provides smoother weights. The main drawback of binomial deviance is similar with lifted structure: It assigns all pairs with nonzero weights, especially the numerous uninformative negative pairs, which brings side effect to model training. Another drawback is that it solely takes the pair's *absolute similarity* $(s_{i,j} - \gamma)$ into consideration, while ignoring the *relative similarity* compared with other pairs, which is contrary to lifted structure loss.

To summarize, contrastive and triplet approaches are impeded by assigning all hard pairs with the same weight without being aware of their different hardness. Lifted structure and binomial deviance approaches mitigate the problem by assigning pairs with weights dynamically based on pairwise similarities. However, they suffers from serious side effects *e.g.*, numerous easy pairs, only considering the absolute or relative similarities of pairs.

5 Weight Assignment Design

Theorem 3.1 bridges all the pair-based DML methods with weight assignment strategies of pairs, which instructs that designing a powerful DML approach is equivalent to seeking a suitable weight assignment. From the review of existing DML methods, we argue that an effective weight assignment strategy should have two desirable properties: (1) Not involving *uninformative pairs* in the learning process. (2) Assigning weight coefficients to pairs base on their both *absolute* and *relative similarities*. In the following, we elaborate how to design such a DML method.

5.1 Valid Triplet Based Hard Mining

As the model converges, most pairs, especially negative pairs, have been well addressed and cannot further improve the model. In fact, these easy pairs without any useful information may bring side effect to the training process. That is to say, only samples that can provide useful information should be involved in the training. Then, it comes the problem mentioned frequently in recent papers: hard mining, which is to exploit informative samples to promote the learning of an effective network. Here we propose our hard mining approach that takes the local distribution of all the pairs

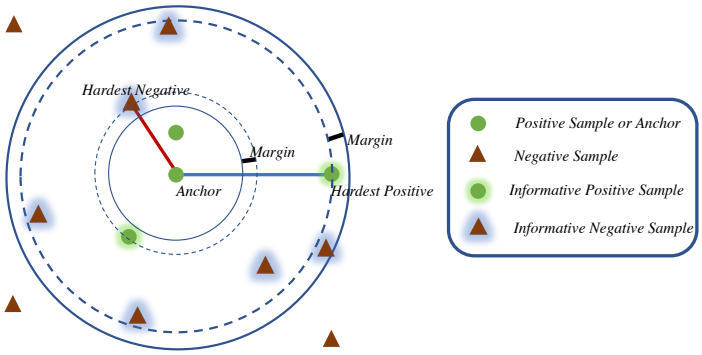


Figure 2: Valid triplet based hard mining: for any anchor point, every positive pair is compared with the hardest negative pair, and vice versa.

of one anchor into consideration. First, for every sample $\boldsymbol{x}_i$ as the anchor, and the hardest positive sample of $\boldsymbol{x}_i$ is defined as $\boldsymbol{x}_{p^*} := \arg\min_{\boldsymbol{x}_k : k \in P_i} s_{i,k}$, similarly, the hardest negative sample is $\boldsymbol{x}_{n^*} := \arg\max_{\boldsymbol{x}_k : k \in N_i} s_{i,k}$. Second, we compare each negative sample $\boldsymbol{x}_j$ with $\boldsymbol{x}_{p^*}$. It is treated as informative when $s_{i,j} > s_{i,p^*} - \lambda$, and for any positive pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$, it is considered as informative when $s_{i,j} < s_{i,n^*} + \lambda$. The sampling process is shown in Figure (2). We call it valid triplet based hard mining (VTHM), since only triplets involving these pairs may have nonzero triplet loss. The index set of informative positive (or negative) samples of $\boldsymbol{x}_i$ is denoted as $\widetilde{P}_i$ (or $\widetilde{N}_i$).

Compared with the triplet approach selecting one pair only based on another randomly sampled pair, our VTHM selects positive (negative) pair by comparing it with all the negative (positive) pairs with the same anchor, which is stable and exploits more informative pairs. Compared with the hard mining method in (Harwood et al. (2017)), VTHM doesn't need off-line computation and can be directly combined with existing pair-based losses to improve their performances. Compared with the distance weighted sampling in Wu et al. (2017) that targets to select a wide range of negative examples w.r.t. similarity, our method focuses on mining pairs with information.

We assign zero weights to these uninformative pairs which don't belong to $\widetilde{P}_i \bigcup \widetilde{N}_i$. Next, we describe our weight assignment strategy for the informative pairs in $\widetilde{P}_i \bigcup \widetilde{N}_i$.

5.2 Relative and Absolute Similarity Based Weight Assignment (RAW)

Our weight assignment strategy takes advantage of lifted structure and binomial deviance approaches as below:

$$w_{i,j} = \frac{1}{\left[\sum_{k \in \widetilde{P}_i} \exp\left(\alpha\left(s_{i,j} - s_{i,k}\right)\right)\right] + \exp\left(\alpha\left(s_{i,j} - \gamma\right)\right)} \qquad if \quad j \in \widetilde{P}_i \tag{15}$$

$$w_{i,j} = \frac{1}{\left[\sum_{k \in \widetilde{N}_i} \exp\left(-\beta\left(s_{i,j} - s_{i,k}\right)\right)\right] + \exp\left(-\beta\left(s_{i,j} - \gamma\right)\right)} \qquad if \quad j \in \widetilde{N}_i, \tag{16}$$

where α, β, γ are hype-parameters.

In Equation (15), $\sum_{k \in \widetilde{P}_i} \exp\left(\alpha\left(s_{i,j} - s_{i,k}\right)\right)$ captures relative similarity between the positive pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$ and other informative positive pairs, and $\exp\left(\alpha\left(s_{i,j} - \gamma\right)\right)$ utilizes a fixed threshold γ to evaluate its absolute similarity. Analysis is the same for negative pairs. We call it relative and absolute similarity based weight assignment (RAW). Though RAW seems a simple combination of lifted structure and binomial deviance, it is not trivial to derive without our unified view.

To avoid the computation of $w_{i,j}$ for each pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$, we obtain the loss function $\mathcal{L}_{RAW}$ that is gradient equivalent to our proposed weight assignment:

$$\mathcal{L}_{RAW} = \frac{1}{m} \sum_{i=1}^{m} \left\{ \frac{1}{\alpha} \log \left[1 + \sum_{k \in \widetilde{P}_i} \exp\left(-\alpha(s_{i,k} - \gamma)\right) \right] + \frac{1}{\beta} \log \left[1 + \sum_{k \in \widetilde{N}_i} \exp\left(\beta(s_{i,k} - \gamma)\right) \right] \right\} \tag{17}$$

Table 1: Ablation study on CUB-200, Cars-196, SOP, In-shop. Only Recall@1 is given.

	CUB-200	Cars-196	SOP	In-shop
Binomial	64.45	80.78	73.4	84.78
RAW	65.06	81.27	77.0	88.38
VTHM	61.55	76.61	76.82	88.72
Binomial+VTHM	65.34	81.48	77.22	88.87
RAW+DW	65.67	80.70	77.39	88.79
RAW + SemiHard	64.97	80.48	77.12	88.66
RAW+VTHM (ours)	**66.85**	**83.69**	**78.18**	**89.64**

6 EXPERIMENTS

We use PyTorch to implement our model. For the network architecture, we use the Inception network with batch normalization (Ioffe & Szegedy (2015)) pretrained on ImageNet. We add an FC layer at the top of the network following the global pooling layer. L_2 normalization is applied to the embedding vectors. All the input images are first resized to 256×256 and then cropped to 227×227. For data augmentation, we used random crop with random horizontal mirroring for training and only one center crop for testing. The embedding dimension is set to 512. We use the Adam optimizer with a fixed 10^{-5} learning rate for all experiments. We conduct the experiments on four standard datasets: CUB-200-2011 (Wah et al. (2011)), Cars196 (Krause et al. (2013)), Stanford Online Products (SOP) (Oh Song et al. (2016)) and In-shop Slothes (In-shop) (Liu et al. (2016b)) . We follow the data split protocol proposed in Oh Song et al. (2016). For every mini-batch, we randomly choose a certain number of classes, and then randomly sample K instances from each class. We set $K = 5$ for CUB and Cars196, $K = 4$ for SOP and In-shop. The margin λ in VTHM is 0.1, and $\alpha = 2$, $\beta = 50$ and $\gamma = 0.5$ in both Equation (13) and Equation (17). Our proposed method is verified on image retrieval task and evaluated by the standard performance metric: Recall@K as Oh Song et al. (2016).

6.1 ABLATION STUDY

We conduct an ablation study by comparing the following methods: **Binomial** applies binomial deviance loss to all the pairs in the mini-batch. **VTHM** gives equal weights to the pairs selected by our VTHM strategy. **Binomial** and **RAW** assign weights to all the pairs via binomial deviance and RAW respectively. **Binomial+VTHM** utilizes the weight assignment of binomial deviance to the selected pairs by our hard mining. **RAW+VTHM** represents our full method that assigns weights to informative pairs via Equation (17) with VTHM. **RAW+DW** and **RAW+SemiHard** denote methods that assign weight given by RAW to pairs sampled through distance weighted sampling (Wu et al. (2017)) and semi-hard mining (Schroff et al. (2015)) respectively. For simplicity, only Recall@1 are reported as shown in Table 1.

Sampling. From this table, VTHM leads to improvement on all datasets for Binomial and RAW. It offers 3.8% improvement on SOP and 4.4% on In-shop for binomial deviance loss. For RAW, it increases the Recall@1 by more than 2% on CUB-200 and Cars-196, while other sampling methods like semi-hard and distance weight sampling do not have such a positive impact. This is because VTHM mines as many informative pairs as possible. Therefore, we conclude that though Binomial and RAW have already assigned easy pairs with small weight values, these samples still do harms to the learning process because of their large number.

Weight Assignment. In Table 1, we find that Binomial+VTHM outperforms VTHM on all datasets simultaneously through assigning pairs with weight base on their absolute similarity. Moreover, our RAW+VTHM further improves the performance by considering the relative hardness of pairs. For instance, on Cars-196, Binomial+VTHM achieves 6% higher Recall@1 than VTHM, and RAW+VTHM further increases 2%.

Table 2: Recall@$K(\%)$ performance on CUB-200 and Cars-196.

	CUB-200-2011						Cars-196					
Recall@K	1	2	4	8	16	32	1	2	4	8	16	32
HDC	53.6	65.7	77.0	85.6	91.5	95.5	73.7	83.2	89.5	93.8	96.7	98.4
Clustering	48.2	61.4	71.8	81.9	-	-	58.1	70.6	80.3	87.8	-	-
ProxyNCA	49.2	61.9	67.9	72.4	-	-	73.2	82.4	86.4	87.8	-	-
Smart Mining	49.8	62.3	74.1	83.3	-	-	64.7	76.2	84.2	90.2	-	-
Margin	63.6	74.4	83.1	90.0	94.2	-	79.6	86.5	91.9	95.1	97.3	-
HTL	57.1	68.8	78.7	86.5	92.5	95.5	81.4	88.0	92.7	95.7	97.4	99.0
ABIER	57.5	68.7	78.3	86.2	91.9	95.5	82.0	89.0	93.2	96.1	97.8	98.7
RAW+VTHM	**66.85**	**77.84**	**85.8**	**91.29**	**94.94**	**97.42**	**83.69**	**90.27**	**94.53**	**97.16**	**98.65**	**99.36**

Table 3: Recall@$K(\%)$ performance on SOP and In-shop.

	SOP				In-shop					
Recall@K	1	10	100	1000	1	10	20	30	40	50
Clustering	67.0	83.7	93.2	-	-	-	-	-	-	-
HDC	69.5	84.4	92.8	97.7	62.1	84.9	89.0	91.2	92.3	93.1
Margin	72.7	86.2	93.8	98.0	-	-	-	-	-	-
Proxy-NCA	73.7	-	-	-	-	-	-	-	-	-
ABIER	74.2	86.9	94.0	97.8	83.1	95.1	96.9	97.5	97.8	98.0
HTL	74.8	88.3	94.8	98.4	80.9	94.3	95.8	97.2	97.4	97.8
RAW+VTHM	**78.18**	**90.47**	**96.0**	**98.74**	**89.64**	**97.87**	**98.47**	**98.84**	**99.05**	**99.20**

6.2 Comparison with the State-of-the-Art Approaches

We compare the results of our full method (RAW+VTHM) with current state-of-the-art techniques in DML: Clustering (Song et al. (2017)), Proxy-NCA (Movshovitz-Attias et al. (2017)), HDC (Yuan et al. (2016)), Sampling (Wu et al. (2017)), Smart Mining (Harwood et al. (2017)), ABIER (Opitz et al. (2017; 2018)) and HTL (Ge et al. (2018)). Table 2 and Table 3 compare the performances on CUB-200, Cars-196, SOP and In-shop respectively. Our method outperforms the sate-of-the-art results by a large margin on all datasets: Recall@1 increases 3% on CUB-200, 1.7% on Cars-196, 3% on SOP and 6% on In-shop. The results exhibit the effectiveness of our proposed approach.

7 Conclusion

In the work, we focus on loss functions in deep metric learning (DML), and present a unified framework for DML that expresses all pair-based approaches as different weight assignment strategies. We disclose the hidden drawbacks of existing DML methods by analyzing their weight assignment strategies. Moreover, we exploit two key points to design an effective DML approach: (1) avoiding the side effect of easy pairs and assigning weight coefficients to informative pairs based their absolute and (2) relative similarities. We then propose our DML approach under the direction of these two points. We show the importance of these points respectively through an ablation study, Further, we demonstrate through experiments that our approach improves the state-of-the-art performance significantly.

REFERENCES

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546 vol. 1, June 2005. doi: 10.1109/CVPR.2005.202.

Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R Scott. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–285, 2018.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742, 2006. doi: 10.1109/CVPR.2006.100.

Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. pp. 2840–2848, 10 2017.

Alexander Hermans*, Lucas Beyer*, and Bastian Leibe. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*, 2017.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *SIMBAD*, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

M. T. Law, N. Thome, and M. Cord. Quadruplet-wise image similarity learning, Dec 2013. ISSN 1550-5499.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 507–516, New York, New York, USA, 20–22 Jun 2016a. PMLR. URL http://proceedings.mlr.press/v48/liud16.html.

Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016b.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 360–368, 2017. doi: 10.1109/ICCV.2017.47. URL http://doi.ieeecomputersociety.org/10.1109/ICCV.2017.47.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

M. Opitz, G. Waltner, H. Possegger, and H. Bischof. Deep Metric Learning with BIER: Boosting Independent Embeddings Robustly. *arXiv:cs/1801.04815*, 2018.

Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Bier - boosting independent embeddings robustly. In *ICCV*, 2017.

Oren Rippel, Manohar Paluri, Piotr Dollár, and Lubomir D. Bourdev. Metric learning with adaptive density discrimination. *CoRR*, abs/1511.05939, 2015. URL http://arxiv.org/abs/1511.05939.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1857–1865. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6200-improved-deep-metric-learning-with-multi-class-n-pair-loss-objective.pdf.

Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4170–4178. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6464-learning-deep-embeddings-with-histogram-loss.pdf.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Master's thesis, None, 2011.

Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2612–2620, 2017.

Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *CoRR*, abs/1706.07567, 2017. URL http://arxiv.org/abs/1706.07567.

Dong Yi, Zhen Lei, and Stan Z. Li. Deep metric learning for practical person re-identification. *CoRR*, abs/1407.4979, 2014. URL http://arxiv.org/abs/1407.4979.

Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. *CoRR*, abs/1611.05720, 2016. URL http://arxiv.org/abs/1611.05720.