Hypergraph-Induced Semantic Tuplet Loss for Deep Metric Learning

Jongin Lim[1, 2] Sangdoo Yun[3] Seulki Park[1] Jin Young Choi[1]
[1]ASRI, Dept. of ECE., Seoul National University
[2]Samsung Advanced Institute of Technology [3]NAVER AI Lab
{ljin0429,seulki.park,jychoi}@snu.ac.kr sangdoo.yun@navercorp.com

Abstract

In this paper, we propose Hypergraph-Induced Semantic Tuplet (HIST) loss for deep metric learning that leverages the multilateral semantic relations of multiple samples to multiple classes via hypergraph modeling. We formulate deep metric learning as a hypergraph node classification problem in which each sample in a mini-batch is regarded as a node and each hyperedge models class-specific semantic relations represented by a semantic tuplet. Unlike previous graph-based losses that only use a bundle of pairwise relations, our HIST loss takes advantage of the multilateral semantic relations provided by the semantic tuplets through hypergraph modeling. Notably, by leveraging the rich multilateral semantic relations, HIST loss guides the embedding model to learn class-discriminative visual semantics, contributing to better generalization performance and model robustness against input corruptions. Extensive experiments and ablations provide a strong motivation for the proposed method and show that our HIST loss leads to improved feature learning, achieving state-of-the-art results on three widely used benchmarks. Code is available at https://github.com/ljin0429/HIST.

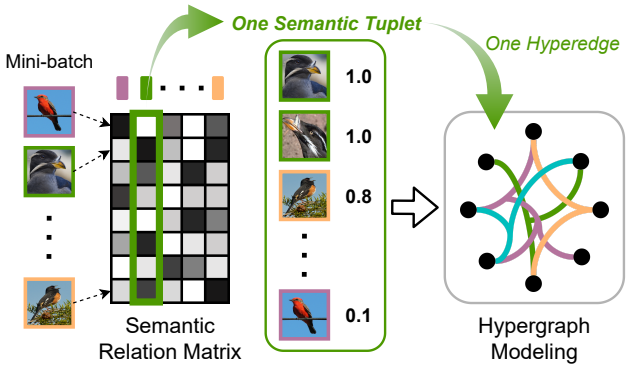


Figure 1. Our HIST loss utilizes multilateral semantic relations between every sample and class (marked by color) for a given mini-batch. A semantic tuplet is defined for a class (*e.g.*, green) and represents the sample's semantic relations to the class. Inside the semantic tuplet, positive samples have definite relation values ($= 1$), and negative samples have soft relation values (≤ 1) based on their likelihood of belonging to the class. Each semantic tuplet is then modeled as a hyperedge. In this hypergraph, we formulate a node classification objective. By leveraging multilateral semantic relations, HIST loss enables the embedding network to capture important visual semantics suitable for deep metric learning.

1. Introduction

Deep metric learning has been extensively studied for a variety of visual tasks, such as image retrieval [29, 37, 46], face recognition [25, 34, 48], person re-identification [4, 50], and few-shot learning [36, 39, 42]. The aim of deep metric learning is to train a deep embedding network to yield discriminative features whereby the embedded features from semantically similar images are close to each other while those from dissimilar ones are far apart. This discerning capability of the embedding network is mainly achieved through loss functions, and many attempts have been made to design optimal loss functions for deep metric learning.

Conventionally, pair-based losses (*e.g.*, Contrastive [5, 14], Triplet [19, 34], and N-pair [37] losses) have been employed. These minimize the feature distances of positive pairs while maximizing those of negative pairs. However, because not all data pairs are informative, pair-based losses often result in bad convergence [21, 29]. For reliable performance, pair-based losses require elaborate sample mining [15, 17, 49, 53], adding a computational burden. Alternative options are proxy-based [1, 13, 21, 29, 40, 60] and classification-based [26, 32, 44, 45, 48, 54] losses, which have demonstrated fast convergence and good performance. However, as they associate each data sample only with representative parameters (*i.e.*, proxies or classification weights), neither proxy-based nor classification-based losses can leverage relations between data samples, which can limit the quality of the learned features.

Recently, to resolve the above limitations, several graph-based losses [7, 35, 55, 60] have been proposed that leverage relations between data samples via graph modeling. These

methods construct a graph between data samples within a mini-batch and then formulate graph-based learning objectives. Although they have shown promising performance improvements, these graph-based losses have inherent limitations. Since each edge in the graph can only connect two nodes, lessons of graph-based losses are limited to a bundle of pairwise relations. Furthermore, each edge is defined by the feature distance or self-attention [41] and is determined regardless of the classes of the two samples. That is, graph-based losses only consider pairwise feature relations and cannot take advantage of class semantic relations. Intuitively, learning from multilateral relations between sample and class, *i.e.*, relations among samples from the same class and similar-looking samples from different classes, must be helpful for understanding class-discriminative visual semantics, leading to improved feature learning.

In this work, we propose **H**ypergraph-**I**nduced **S**emantic **T**uplet (**HIST**) loss, a novel loss function for deep metric learning that leverages multilateral semantic relations between every sample and every class within a mini-batch via hypergraph modeling[1]. Concretely, such semantic relations are given by the proposed *semantic tuplets*. As shown in Figure 1, the semantic tuplets are expressed by a semantic relation matrix with learnable elements where each row indicates the relation of each sample to every class in the mini-batch, and each column represents the relation of each class to every sample in the mini-batch. Thus, the semantic tuplets represent the multilateral semantic relations between every sample and every class by the learnable matrix. To fully exploit these multilateral semantic relations, we introduce hypergraph modeling whereby each semantic tuplet is modeled by a hyperedge. In this hypergraph, we formulate a node classification problem employing a hypergraph neural network (HGNN) [8] and define HIST loss as the node classification loss. This formulation utilizing HGNN allows our HIST loss to benefit from the rich multilateral semantic relations provided by the proposed semantic tuples beyond pairwise feature relations.

We validate our method on three public benchmarks for deep metric learning, CUB-200-2011 [43], CARS-196 [22], and Stanford Online Products [31]. In the experiments, we present extensive ablation studies and parameter analyses to demonstrate the effectiveness of the proposed components. In particular, we show that our HIST loss directs the embedding model to attend to meaningful object regions rather than background or distracting noises, contributing to better generalization performance and model robustness against input corruptions. The main results show that a standard embedding network trained with our HIST loss significantly outperforms state-of-the-art methods for all benchmarks.

[1] A hypergraph is a generalization of a graph where each hyperedge can connect more than two nodes.

2. Related Work

Pair-based losses. Triplet loss [19, 34] is a seminar example, which aims to shorten the distance from the positive pair while increasing that from the negative pair. As extensions of Triplet loss, N-pair [37], Lifted Structure [31],and Tuplet Margin [52] losses considers multiple negative samples. Multi-Similarity [46] considers every pair of data in a mini-batch and assigns weights to each pair based on similarities. However, the pair-based losses empirically suffer from slow convergence [21, 29].

Proxy-based losses. The key idea of this group of losses is to infer proxies and associate each data sample with proxies instead of other data samples. ProxyNCA [29] first introduced the concept of proxy and presented the proxy-based training scheme built upon Neighborhood Component Analysis (NCA) [11]. Manifold Proxy [1] improves the performance by adopting a manifold-aware distance. ProxyNCA++ [40] enhances the performance of ProxyNCA with assorted training techniques. Recently, Proxy Anchor [21] has shown promising results, which takes each proxy as an anchor and computes the loss from the proxy perspective. However, since the proxy-based losses associate each data sample only with proxies, they cannot leverage relations between data samples.

Classification-based losses. This group of losses employs a classifier to train the model like a classification task. A recent line of work [26, 32, 44, 45, 48, 54] has shown that elaborately designed classification losses can yield competitive results. Specifically, Normalized Softmax [54] combined with a balanced sampling strategy has shown promising results. SoftTriple [32] utilizes multiple classifiers to classify each data sample. However, in the above methods, each sample is classified individually, and relations between data samples are not considered. In contrast, we utilize the hypergraph-based classifier that leverages rich relations among multiple data samples.

Graph-based losses. Group Loss [7] computes a similarity matrix representing the pairwise similarity between all data samples in a mini-batch and utilizes Label Propagation (LP) [2, 58, 59] on the similarity matrix. On the other hand, ProxyGML [60] constructs a directed bipartite graph to model the relations between all the proxies and data samples in a mini-batch, and then, utilizes a variant of LP. Recently, IBC [35] constructs a fully connected graph for mini-batch samples and classifies each sample employing Message Passing Network [10]. However, graph modeling can only formulate pairwise relations between data samples. Unlike graph modeling, the hypergraph can effectively formulate higher-order relations between multiple data samples by enclosing multiple nodes within a hyperedge. To the best of our knowledge, we firstly introduce hypergraph modeling into a deep metric learning loss function.

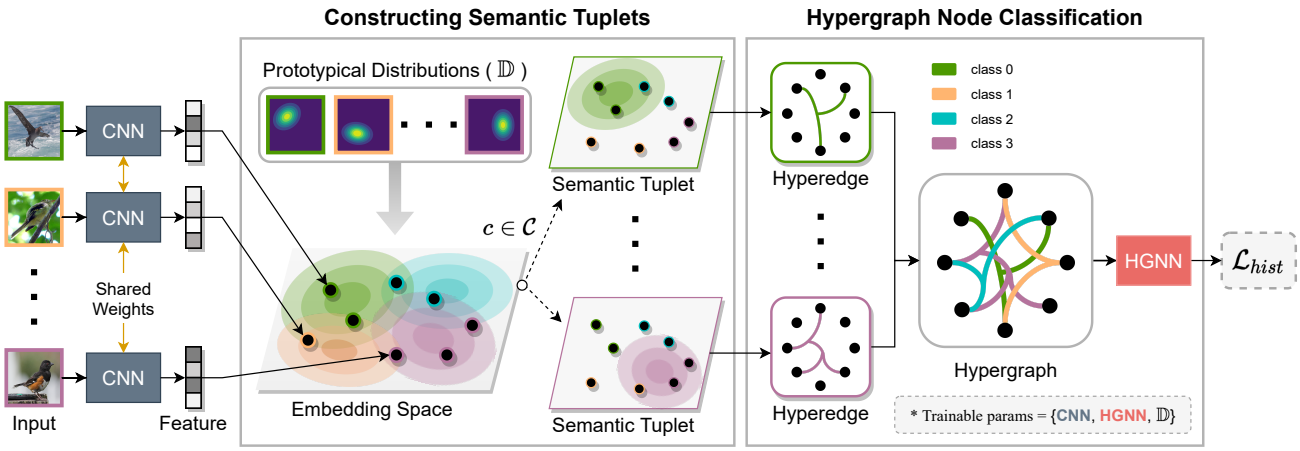


Figure 2. The overall pipeline of Hypergraph-Induced Semantic Tuplet (HIST) loss. HIST loss consists of two main steps: semantic tuplets construction and hypergraph node classification. Given a mini-batch, we construct a semantic tuplet for each class, a tuple of samples that have semantic relations to that class, based on the feature distribution in the embedding space. Then, we form a hypergraph where each hyperedge stands for a semantic tuplet and connects the corresponding nodes in the semantic tuplet all at once. In this hypergraph, we formulate a hypergraph node classification objective, employing a hypergraph neural network (HGNN).

3. Method

3.1. Overview

Consider a CNN model that maps an input image $\mathbf{x}_i$ to a D-dimensional feature $\mathbf{z}_i \in \mathbb{R}^D$ as

$$\mathbf{z}_i = E(\mathbf{x}_i; \mathbf{\Theta}), \tag{1}$$

where $\mathbf{\Theta}$ denotes the overall network parameters. Given a labeled training set with C classes, our goal is to train the model $E(.)$ towards yielding a discriminative feature embedding. Formally, we let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ denote a set of N training images and $\mathcal{Y} = \{y_1, y_2, \cdots, y_N\}$ denote a set of corresponding labels where $y_i \in \{1, 2, \cdots, C\}$ indicates one of C classes. We adopt a mini-batch training and our HIST loss leverages rich correlations among samples in the mini-batch provided by a hypergraph modeling. Specifically, we consider a *randomly* sampled mini-batch $\mathcal{B} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_b}$, consisting of N_b images and the corresponding labels. In addition, we let $\mathcal{C} \subset \{1, 2, \cdots, C\}$ denote a subset of classes included in the mini-batch $\mathcal{B}$.

Figure 2 shows the overall pipeline of our HIST loss. Given a mini-batch, we define a *semantic tuplet* for each class $c \in \mathcal{C}$, resulting a total of $|\mathcal{C}|$ semantic tuplets. Unlike previous tuplet losses [31, 37, 52], in which a tuplet is defined for each anchor image, our semantic tuplet is defined for each class c and consists of samples that have semantic relations to class c, such as images of class c and images of other classes that are likely to belong to class c. To model such semantic relations, we introduce a set of learnable distributions, dubbed *prototypical distributions* (see Section 3.2), and construct the semantic tuplets based on these prototypical distributions (see Section 3.3). We then formulate a hypergraph modeling where each hyperedge stands for a semantic tuplet and connects the corresponding nodes in the semantic tuplet all at once (see Section 3.4). In this hypergraph, we perform node classification using HGNN [8] (see Section 3.5). Consequently, our HIST loss leverages rich semantic relations provided by semantic tuplets through hypergraph message passing of HGNN. It should be noted that the entire computation of the HIST loss is fully differentiable, and the overall parameters are jointly trained in an end-to-end manner. In Section 3.6, we discuss the underlying rationales of HIST loss.

3.2. Learning Prototypical Distributions

In this section, we present a set of learnable distributions $\mathbb{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_C\}$, which we refer to *prototypical distributions*, that aim to model the true feature distribution. Concretely, each prototypical distribution $\mathcal{D}_c$ is assigned for each class c to represent the entire features of c, *i.e.*, $\mathcal{Z}_c = \{\mathbf{z}_i | \mathbf{x}_i \in \mathcal{X}, y_i = c\}$. The real-world data includes intra-class variations such as poses, viewpoints, and backgrounds. To handle such intra-class variations, each $\mathcal{D}_c$ is realized with two learnable parameters, mean $\boldsymbol{\mu}_c \in \mathbb{R}^D$ and covariance $\mathbf{Q}_c \in \mathbb{R}^{D \times D}$, which represents class centroid and intra-class variations, respectively. In practice, for computational efficiency, we formulate a diagonal covariance matrix where each $\mathbf{Q}_c$ is simplified with a D-dimensional vector $\mathbf{q}_c \in \mathbb{R}^D$ as $\mathbf{Q}_c = \text{diag}(\mathbf{q}_c)$. Hence, the overall parameters of $\mathbb{D}$ are denoted as $\mathbf{\Phi} = \{(\boldsymbol{\mu}_c, \mathbf{q}_c)\}_{c=1}^{C}$, which can be jointly trained by back-propagation.

Now, we formalize our distribution loss $\mathcal{L}_D$, which ensures that each prototypical distribution well captures the true feature distribution. Inspired by NCA [11], we asso-

ciate each sample in $\{\mathbf{z}_i\}_{i=1}^{N_b}$ with the *nearest* prototypical distribution and maximize the probability of correct association. Unlike NCA, since we associate each sample with a distribution, the squared Mahalanobis distance is employed for the distance metric where the distance between $\mathbf{z}_i$ and $\mathcal{D}_c$ is defined as $d_m^2(\mathbf{z}_i, \mathcal{D}_c) = (\mathbf{z}_i - \boldsymbol{\mu}_c)^\top \mathbf{Q}_c^{-1}(\mathbf{z}_i - \boldsymbol{\mu}_c)$. We consequently define each sample probability P_i, *i.e.*, the probability that $\mathbf{z}_i$ is associated with the correct prototypical distribution $\mathcal{D}_+$, as

$$P_i = \frac{\exp(-\tau d_m^2(\mathbf{z}_i, \mathcal{D}_+))}{\sum_{\mathcal{D}_c \in \mathbb{D}} \exp(-\tau d_m^2(\mathbf{z}_i, \mathcal{D}_c))}, \tag{2}$$

where $\tau > 0$ is a temperature scaling factor [18]. Then, our distribution loss $\mathcal{L}_D$ is given by the negative log-likelihood of the overall sample probability for the mini-batch $\mathcal{B}$ as

$$\mathcal{L}_D = \frac{1}{N_b} \sum_{i=1}^{N_b} -\log P_i. \tag{3}$$

This supervision aims at making prototypical distributions well represent the true feature distribution. The better representation power helps to improve the quality of semantic tuplets, which will be presented in the following section.

3.3. Constructing Semantic Tuplets

The notation of tuplet was first introduced to extend the triplet by exploring multiple negatives [31, 37, 52], and each tuplet provides pairwise supervisions for minimizing feature distances of positive pairs while maximizing those of negative pairs. In contrast, we define a semantic tuplet for each class c, consisting of multiple samples that share semantic relations to the class c. Furthermore, our semantic tuplets are modeled by a hypergraph and used for the hypergraph node classification, providing multilateral semantic relations rather than pairwise supervisions.

Formally, we construct a semantic tuplet $\mathcal{S}(c)$ for each class $c \in \mathcal{C}$ from the mini-batch $\mathcal{B}$. Consequently, a total of $|\mathcal{C}|$ semantic tuplets are constructed. Using the prototypical distributions $\mathbb{D}$, the semantic tuplets are expressed by the semantic relation matrix $\mathbf{S} \in [0,1]^{N_b \times |\mathcal{C}|}$ of which ij-th element is given by

$$\mathbf{S}_{ij} = \begin{cases} 1 & \text{if } y_i = \mathcal{C}_j, \\ e^{-\alpha d_m^2(\mathbf{z}_i, \mathcal{D}_{\mathcal{C}_j})} & \text{otherwise}, \end{cases} \tag{4}$$

where $\mathcal{C}_j$ denotes the j-th class in $\mathcal{C}$, and α is a positive scalar that controls the reflection ratio of negative samples.

In Eq (4), each row and column of $\mathbf{S}$ represent a sample in $\mathcal{B}$ and each semantic tuplet for a class in $\mathcal{C}$, respectively. For each semantic tuplet $\mathcal{S}(\mathcal{C}_j)$, positive samples of the class $\mathcal{C}_j$ are assigned definitely, whereas negative samples are assigned with weights determined by the squared Mahalanobis distance from $\mathcal{D}_{\mathcal{C}_j}$. Since $\mathcal{D}_{\mathcal{C}_j}$ models the true feature distribution of the class $\mathcal{C}_j$, these weights reflect their sample likelihood of belonging to the class $\mathcal{C}_j$. Besides, this can be viewed as paying more attention to the harder negative sample, since the harder negative sample, *i.e.*, more closer to $\mathcal{D}_{\mathcal{C}_j}$, is assigned to $\mathcal{S}(\mathcal{C}_j)$ with a greater weight. Therefore, our semantic tuples provide multilateral semantic relations between every sample and class in the mini-batch, and these rich semantic relations are fully utilized through hypergraph modeling.

3.4. Hypergraph Modeling

Here, we first briefly describe hypergraph notations and then present our hypergraph modeling for HIST loss.

In general, a hypergraph is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ consisting of a node set $\mathcal{V}$ and a hyperedge set $\mathcal{E}$. Above all, unlike the graph in which each edge connects only two nodes, a hyperedge can connect multiple nodes that are related to each other. Therefore, higher-order relations between multiple samples can be effectively modeled by a hypergraph. The hypergraph structure can be represented by an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, with entries defined as

$$\mathbf{H}_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

For a node $v_i \in \mathcal{V}$, its degree is defined as $d(v_i) = \sum_{j=1}^{|\mathcal{E}|} \mathbf{H}_{ij}$. For a hyperedge $e_j \in \mathcal{E}$, its degree is defined as $\delta(e_j) = \sum_{i=1}^{|\mathcal{V}|} \mathbf{H}_{ij}$. In addition, $\mathbf{D}_v$ and $\mathbf{D}_e$ denote the diagonal matrices of the node degrees and the hyperedge degrees, respectively.

Now, we derive our hypergraph modeling for HIST loss. Given the mini-batch $\mathcal{B}$, we construct a hypergraph whose nodes and hyperedges denote samples and semantic tuplets, respectively. Concretely, each node v_i corresponds to the sample $\mathbf{x}_i$, and its node feature is assigned by the embedded feature $\mathbf{z}_i$. The overall node features of the hypergraph are represented by the feature matrix $\mathbf{Z} \in \mathbb{R}^{N_b \times D}$, defined as $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_{N_b}]^\top$. In our design, to reflect soft relation between sample and class, each hyperedge connects nodes with soft incidence weights in $[0, 1]$, where the nodes for positive samples in each class are assigned by incidence weights of 1, whereas the nodes for negative samples from other classes are assigned by incidence weights less than 1. To this end, the weighted incidence matrix $\mathbf{H} \in \mathbb{R}^{N_b \times |\mathcal{C}|}$ of the proposed hypergraph is designed by the semantic tuplets denoted as the semantic relation matrix $\mathbf{S} \in \mathbb{R}^{N_b \times |\mathcal{C}|}$ in Eq (4). That is, $\mathbf{H}$ is set to $\mathbf{S}$. Thus, each hyperedge e_j stands for the semantic tuplet $\mathcal{S}(\mathcal{C}_j)$ for j-th class $\mathcal{C}_j$. For each sample in the class $\mathcal{C}_j$, the hyperedge e_j is called *positive hyperedge* having the relation value of one and the others are called *negative hyperedges* having a relation value less than one as in (4).

3.5. HIST Loss

After the hypergraph construction, we formulate a hypergraph node classification objective, employing Hypergraph Neural Network (HGNN) [8]. The node features, *i.e.*, embeddings of mini-batch samples, are updated via hypergraph message passing steps of HGNN, allowing each sample to be classified in consideration of semantic tuplets.

Concretely, we utilize L layers of HGNN, which applies L message passing steps successively. In each step, the l-th layer takes a feature matrix $\mathbf{Z}^{(l)} \in \mathbb{R}^{N_b \times d_l}$ as an input and outputs a feature matrix $\mathbf{Z}^{(l+1)} \in \mathbb{R}^{N_b \times d_{l+1}}$ by propagating messages through the hypergraph $\mathbf{H}$. Formally, given an input feature matrix $\mathbf{Z}^{(0)} = \mathbf{Z}$ and the hypergraph $\mathbf{H}$, HGNN conducts the following layer-wise feature update as

$$\mathbf{Z}^{(l+1)} = \sigma\big(\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-\frac{1}{2}}\mathbf{Z}^{(l)}\mathbf{\Psi}^{(l)}\big), \quad (6)$$

where $l = 0, 1, \cdots, L-1$ and $\mathbf{\Psi}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ denotes a trainable weight matrix for feature transform at l-th layer. Function $\sigma(.)$ denotes a non-linear activation. We let $\mathbf{\Psi}$ denote the overall network parameters of HGNN.

The last layer of HGNN outputs the final representation of each node, of which dimension is set to the number of classes, *i.e.*, $\mathbf{Z}^{(L)} \in \mathbb{R}^{N_b \times C}$. On top of the final representation, we add a softmax activation function on each row of $\mathbf{Z}^{(L)}$ and obtain class predictions for each node, *i.e.*, $\hat{\mathbf{Y}} = \mathrm{softmax}(\mathbf{Z}^{(L)})$, where the i-th row of $\hat{\mathbf{Y}}$ represents the class prediction of node v_i. Then, the cross-entropy loss between the predictions and the ground-truth labels over all nodes in the hypergraph is given as

$$\mathcal{L}_{CE} = -\frac{1}{N_b}\sum_{i=1}^{N_b}\sum_{j=1}^{C}\mathbf{Y}_{ij}\log\hat{\mathbf{Y}}_{ij}, \quad (7)$$

where $\mathbf{Y} \in \mathbb{R}^{N_b \times C}$ denotes the ground-truth label matrix whose i-th row denotes a one-hot vector indicating y_i.

Finally, our HIST loss is defined as the weighted sum of the two loss terms, the distribution loss $\mathcal{L}_D$ and the hypergraph node classification loss $\mathcal{L}_{CE}$, as follows

$$\mathcal{L}_{hist} = \mathcal{L}_D + \lambda_s\mathcal{L}_{CE}, \quad (8)$$

where $\lambda_s > 0$ is a scaling parameter to balance the two loss values. Note that the entire HIST loss is fully differentiable, allowing back-propagation from end to end. During training, the overall parameters, *i.e.*, $\mathbf{\Theta}$ of the CNN model $E(.)$, $\mathbf{\Phi}$ of prototypical distributions $\mathbb{D}$, and $\mathbf{\Psi}$ of HGNN, are jointly trained by minimizing $\mathcal{L}_{hist}$. After training, only the CNN model $E(.)$ is used for subsequent tasks such as image retrieval and clustering.

3.6. Rationales

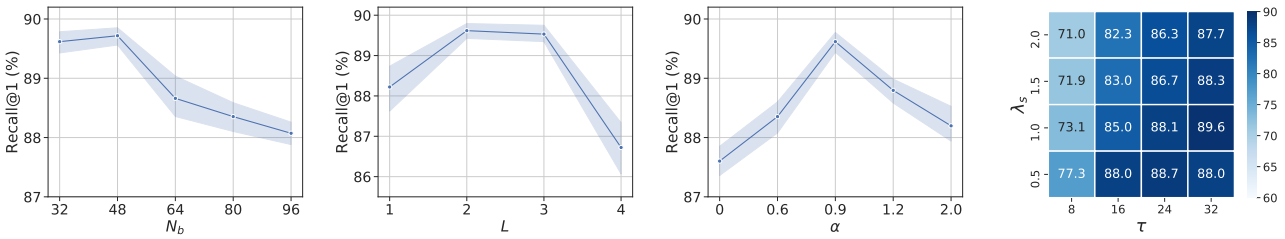
In this section, we examine the underlying rationales of our HIST loss. The key to HIST loss is the hypergraph modeling for semantic tuplets and the node classification objective using HGNN. In essence, each layer of HGNN is a weighted aggregation of the node features connected by the hyperedges. Hence, HGNN makes the node features within the same hyperedge similar. If a hyperedge has negative samples with high incidence weights, their final representations of HGNN will become more similar to those of positive samples and thus hard to be distinguished. To properly discriminate the negative samples from the positive ones, learning should proceed in the direction that each sample (node) does not belong to *negative hyperedges*, *i.e.*, in the direction of reducing its semantic relations (incidence weights) to *negative hyperedges*. In consequence, the CNN model $E(.)$ is enforced to make each sample's feature be far from the feature distributions of the other classes corresponding to *negative hyperedges*, resulting in more discriminative features. Furthermore, to distinguish each samples in the same semantic tuplet, our HIST loss would guide the CNN model $E(.)$ to capture important visual semantics. As a result, the embedding network trained with HIST loss attends well to the meaningful object region rather than background or distracting noises and demonstrates robustness to input corruptions, which will be validated in Section 4.3.

4. Experiments

4.1. Experimental Setup

Datasets and metrics. Experiments were conducted on three widely used benchmarks for deep metric learning: CUB-200-2011 [43], CARS-196 [22], and Stanford Online Products (SOP) [31]. We split the datasets into training and test sets, according to the standard settings [27,31]. We then conducted image retrieval and clustering on the test sets. It should be noted that there were no overlapping classes between the training and test splits, *i.e.*, retrieval and clustering were performed for *unseen* classes. To evaluate the retrieval performance, we adopted the Recall@K (R@K) metric. To evaluate the clustering quality, we applied K-means clustering on the embedding feature vectors of all test samples and computed Normalized Mutual Information (NMI) based on the clustering result. To ensure statistical robustness, we conducted 10 independent runs and reported 95% confidence intervals for the results.

Implementation details. For a fair comparison with previous works, we followed the *standard evaluation settings* for deep metric learning [21, 31, 32, 46]. Specifically, input images were resized to 224×224. During training, images were augmented using random resized cropping and horizontal flipping. During testing, images were resized to 256×256, and then cropped to 224×224 at the center. Following the convention, we considered BN-Inception [20] and ResNet-50 [16] pre-trained on ImageNet [6] as our backbone network, and the results were compared for the



(a) Impact of N_b (b) Impact of L (c) Impact of α (d) Impact of τ and λ_s

Figure 3. Impact of hyper-parameters. We evaluate Recall@1 (%) for different hyperparameter values on CARS-196 dataset. For (a), (b), and (c), the shaded areas represent 95% confidence intervals.

same backbone network. On top of the backbone network, one fully-connected layer was attached to adjust the dimensionality of the embedding vector, where the size of the embedding vector was set to 512. For all experiments, we used two layers of HGNN with a hidden dimension of 512, and we set the mini-batch size to 32. The hyper-parameters α, τ, and λ_s were determined empirically. In addition, we also conducted experiments following the *MLRC evaluation settings* [30] to increase the reliability of the evaluation.

4.2. Parameter Analysis

To validate the efficacy of our HIST loss, we analyzed the impact of hyper-parameters using the CARS-196 dataset. For all analyses, we followed the standard evaluation settings and evaluated the retrieval performance (R@1) using the ResNet-50 backbone network.

Impact of N_b. As our HIST loss exploits relations between samples in the mini-batch, we investigated the impact of the mini-batch size. Figure 3a shows the results of HIST loss with $N_b \in \{32, 48, 64, 80, 96\}$. Notably, HIST loss showed reliable performance regardless of the mini-batch size and worked well with a small mini-batch. This is attributed to the fact that our semantic relations are determined by the prototypical distributions, reflecting the true feature distribution, and hence, are less influenced by the mini-batch size. While the performance slightly improved when $N_b = 48$, we set $N_b = 32$ for efficiency.

Impact of L. Figure 3b shows how the performance varies with the number of HGNN layers L. Our HIST loss generally showed superior performance regardless of the number of HGNN layers. We observed that the performance dropped when $L = 4$, which was due to over-smoothing [23, 24]. The best performance was achieved when $L = 2$, which confirms that proper message passing steps help to enhance the quality of the learned features.

Impact of α. We investigated the impact of the scaling factor α, which controls the reflection ratio of negative samples in Eq (4). Figure 3c shows the results of HIST loss with $\alpha \in \{0, 0.6, 0.9, 1.2, 2.0\}$. When $\alpha = 0$, each semantic tuplet (hyperedge) connected all samples in a mini-batch equally regardless of their semantic relations, propagating useless information between data samples, which led to inferior performance. For $\alpha > 0$, HIST loss showed reliable performance regardless of the α values. While the optimum differed slightly for each dataset, we consistently found that any α around 1 achieved the best performance.

Impact of τ and λ_s. Lastly, we investigated the impact of two hyper-parameters τ and λ_s by varying the values $\tau \in \{8, 16, 24, 32\}$ and $\lambda_s \in \{0.5, 1.0, 1.5, 2.0\}$. Figure 3d demonstrates that our HIST loss is insensitive to the choice of λ_s. Furthermore, the results suggest that any $\tau \geq 16$ yields stable and good performances, which is consistent with the recent argument that large temperature scaling is effective in deep metric learning [21, 40]. Overall, our HIST loss demonstrated reliable and robust performance regardless of the hyper-parameter choices.

4.3. Effectiveness of HIST

Ablation studies. To validate the effectiveness of each component of HIST, we compared HIST with six ablation models, as shown in Table 1. For all ablation models and HIST, we used ResNet-50 as the backbone network and followed the standard evaluation settings. First, as our baseline, we consider the $\mathcal{L}_D$-only model that utilizes $\mathcal{L}_D$ alone without the classification module. Then, for the Single model, a single classification loss is added where each sample is classified individually by a sample classification network instead of the HGNN. The TF-like model extends the Single model by replacing the sample classification network with a Transformer [41]-like classification network. Specifically, the Transformer-like classification network predicts the class label as $\hat{y}_i = \text{softmax}_j(f_Q(\mathbf{z}_i)f_K(\mathbf{z}_j)^T)f_V(\mathbf{z}_i)$, where f_Q, f_K, and f_V are realized with the same number of fc layers as the HGNN. Hence, the TF-like model leverages all pairwise relations in the mini-batch. In addition, D-IBC denotes the IBC [35] model (we used the authors' code) paired with $\mathcal{L}_D$. H-Pos denotes our variant in which each hyperedge connects only positive samples. Lastly, we consider HIST without the distribution loss $\mathcal{L}_D$.

Table 1 shows the retrieval performance of the above models on CARS-196. Compared to the Single, TF-like, and D-IBC models, our HIST model showed a signifi-

Method	Relations	R@1
$\mathcal{L}_D$-only	-	87.3 ± 0.4
+ Single classification:		
Single	-	86.4 ± 0.2
+ Graph-based classification:		
TF-like	Transformer [41]-like attention	87.8 ± 0.3
D-IBC	IBC [35]	87.6 ± 0.3
+ Hypergraph-based classification:		
H-Pos	Only positive samples	87.4 ± 0.2
HIST (w.o. $\mathcal{L}_D$)	Semantic tuplets	88.3 ± 0.2
HIST	Semantic tuplets	**89.6 ± 0.2**

Table 1. Retrieval performance of ablation models on CARS-196.

Input corruption	Single	TF-like	HIST
Additive noise:			
Uniform	83.2 ± 0.1	85.5 ± 0.1	**88.0 ± 0.1**
Gaussian	67.4 ± 0.3	71.3 ± 0.2	**76.2 ± 0.3**
Salt & Pepper	56.2 ± 0.2	58.6 ± 0.2	**68.6 ± 0.2**
Dropping pixels:			
Cutout	73.4 ± 0.3	76.9 ± 0.1	**81.8 ± 0.3**
Dropout	61.3 ± 0.3	65.2 ± 0.2	**72.5 ± 0.4**
Affine transformation:			
Perspective	77.7 ± 0.2	81.0 ± 0.2	**84.3 ± 0.2**
Rotation	69.4 ± 0.3	73.7 ± 0.4	**78.3 ± 0.2**
Degrading image quality:			
JPEG-compression	72.6 ± 0.2	74.0 ± 0.3	**79.7 ± 0.3**
Gaussian blur	64.7 ± 0.2	69.3 ± 0.2	**75.9 ± 0.2**

Table 2. Robustness to *unseen* input corruptions. We evaluated retrieval performance (R@1) with different input corruptions (not used for training) on CAR-196 dataset.

cant performance boost over the $\mathcal{L}_D$-only baseline, which demonstrates the benefits of the proposed hypergraph approach. H-Pos performed better than the baseline, but not the best, which indicates that relations between positive samples only are not sufficient. As discussed in Section 3.6, exploiting the semantic relations of negative samples further improves performance. Lastly, the use of $\mathcal{L}_D$ contributes to the performance improvement of HIST. With $\mathcal{L}_D$, the prototypical distributions better capture the true distribution and improve the quality of semantic tuplets, providing additional performance gain. Further results for other datasets are appended in the supplementary material.

Robustness to input corruptions. Many researchers have shown that deep models are easily fooled by negligible perturbations on the input image [12]. To further demonstrate the effectiveness of HIST, we validated the model robustness to various input corruptions. Specifically, we evaluated the retrieval performance of the embedding network, normally trained on CARS-196, with corrupted test images. As shown in Table 2, we considered nine input corruptions of four types that were not used for training: additive noises (uniform, Gaussian, and Salt & Pepper noises), dropping pixels (cutout and dropout), affine transformation (perspective and rotation), and degrading image quality (JPEG-compression and Gaussian blur). The details of the above corruptions are in the supplementary material.

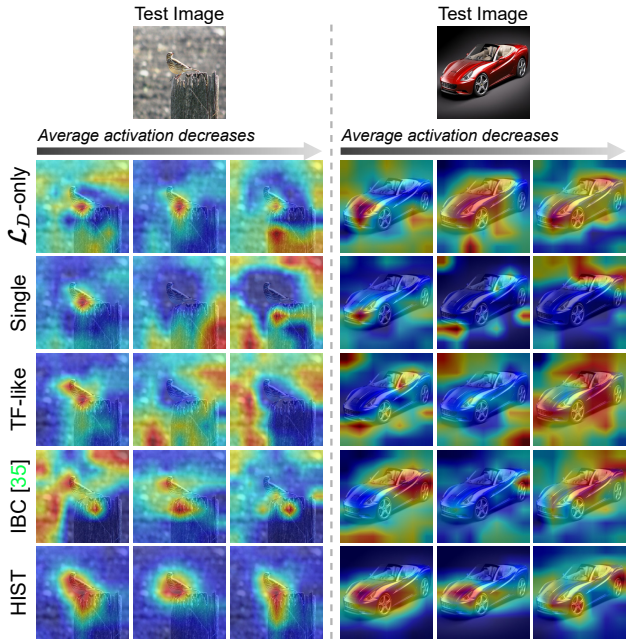


Figure 4. Visualization of three channels of the last feature maps which have the maximal average activation values. More results are appended in the supplementary material.

In Table 2, we compared the results of the Single, TF-like, and HIST models. Our HIST model showed robust and superior performance for all input corruptions, suggesting that the embedding network trained with HIST loss attends to the meaningful area of the input image rather than to distracting noises. In particular, the superiority of HIST is more obvious as the corruption worsens, which further supports the effectiveness of our hypergraph approach compared to the single and graph-based approaches.

Visualization of feature activation maps. To understand the qualitative effect of HIST, we investigated the feature activation maps of the test set images provided by the last convolutional layer of the learned embedding network. In Figure 4, we visualized the top three channels sorted by average activation in descending order. The results show that the embedding network trained with HIST focused better on object regions than the other ablation models, which demonstrates the merit of the hypergraph approach leveraging multilateral semantic relations. Even compared to the recent state-of-the-art graph-based counterpart (IBC [35]; we used the trained model provided by the authors for a fair comparison), HIST showed better semantic focus. Furthermore, the result of HIST for the car image (right side

Method	CUB-200-2011				CARS-196				SOP			
	R@1	R@2	R@4	NMI	R@1	R@2	R@4	NMI	R@1	R@10	R@100	NMI
Methods using BN-Inception:												
HTL^{512} [9]	57.1	68.8	78.7	-	81.4	88.0	92.7	-	74.8	88.3	94.8	-
RLL-H^{512} [47]	57.4	69.7	79.2	63.6	74.0	83.6	90.1	65.4	76.1	89.1	95.4	89.7
MS^{512} [46]	65.7	77.0	86.3	-	84.1	90.4	94.0	-	78.2	90.5	96.0	-
SoftTriple^{512} [32]	65.4	76.4	84.5	69.3	84.5	90.7	94.5	70.1	78.3	90.3	95.9	92.0
GroupLoss^{1024} [7]	65.5	77.0	85.0	69.0	85.6	91.2	94.9	72.7	75.7	88.2	94.8	91.1
CircleLoss^{512} [38]	66.7	77.4	86.2	-	83.4	89.8	94.1	-	78.3	90.5	96.1	-
ProxyAnchor^{512} [21]	68.4	79.2	86.8	-	86.1	91.7	95.0	-	79.1	90.8	**96.2**	-
ProxyGML^{512} [60]	66.6	77.6	86.4	69.8	85.5	91.8	95.3	72.4	78.0	90.6	**96.2**	90.2
DRML^{512} [57]	68.7	78.6	86.3	69.3	86.9	92.1	95.2	72.1	71.5	85.2	93.0	88.1
DAM^{512} [51]	69.1	79.8	87.2	-	86.9	92.1	95.3	-	-	-	-	-
HIST^{512} (Ours)	**69.7**$\pm$0.3	**80.0**$\pm$0.2	**87.3**$\pm$0.2	**70.8**$\pm$0.2	**87.4**$\pm$0.2	**92.5**$\pm$0.3	**95.4**$\pm$0.1	**73.0**$\pm$0.2	**79.6**$\pm$0.2	**91.0**$\pm$0.2	**96.2**$\pm$0.2	**92.2**$\pm$0.3
Methods using ResNet-50:												
N.Softmax^{512} [54]	61.3	73.9	83.5	69.7	84.2	90.4	94.4	74.0	78.2	90.6	96.2	91.0
FastAP^{512} [3]	-	-	-	-	-	-	-	-	76.4	89.0	95.1	-
TML^{512} [52]	62.5	73.9	83.0	-	86.3	92.3	95.4	-	78.0	91.2	**96.7**	-
ProxyAnchor^{512} [21]	69.7	80.0	87.0	-	87.7	92.9	95.8	-	-	-	-	-
ProxyNCA++^{512} [40]	64.7	-	-	-	85.1	-	-	-	79.6	-	-	-
DiVA^{512} [28]	69.2	79.3	-	71.4	87.6	92.9	-	72.2	79.6	91.2	-	90.6
DCML^{512} [56]	68.4	77.9	86.1	71.8	85.2	91.8	96.0	73.9	79.8	90.8	95.8	90.8
S2SD^{512} [33]	70.1	79.7	-	71.6	89.5	**93.9**	-	72.9	80.0	91.4	-	90.8
IBC^{512} [35]	70.3	80.3	87.6	74.0	88.1	93.3	96.2	74.8	**81.4**	91.3	95.9	92.6
HIST^{512} (Ours)	**71.4**$\pm$0.2	**81.1**$\pm$0.3	**88.1**$\pm$0.2	**74.1**$\pm$0.2	**89.6**$\pm$0.2	**93.9**$\pm$0.1	**96.4**$\pm$0.1	**75.2**$\pm$0.3	**81.4**$\pm$0.2	**92.0**$\pm$0.2	**96.7**$\pm$0.1	**92.8**$\pm$0.2

Table 3. Comparisons with state-of-the-arts under the standard evaluation settings. Superscript denotes the embedding dimension. For all compared methods, the results were quoted from the original paper. For our method, we reported the average performance with 95% confidence interval evaluated over 10 independent runs. The best results are marked in **bold**, and the second-best results are underlined.

of Figure 4) shows that the embedding network focused on the entire car in the first channel and then on the specific parts, such as headlights and wheels. This observation implies that our HIST loss guides the embedding network to capture important semantics from the image, contributing to better generalization performance to unseen classes and model robustness against input corruptions, as demonstrated in Tables 1 and 2, respectively.

4.4. Comparison with State-of-the-arts

Table 3 shows the performance comparison against other state-of-the-art methods under the **standard evaluation settings** [21, 31, 32, 46]. As the backbone network has a huge impact on performance, the results are compared for the same backbone network. In all experiments, the standard model trained with our HIST loss achieved state-of-the-art performance. In particular, compared to the recent graph-based losses such as ProxyGML [60], GroupLoss [7], and IBC [35], our HIST loss clearly showed superior performances for all datasets. The outperforming performance of HIST loss comes from our hypergraph approach that leverages multilateral semantic relations between samples, guiding the embedding network to capture important semantics from the image, as presented in Section 4.3. Moreover, to improve the credibility of our evaluation, we further conducted experiments under the **MLRC evaluation settings** [30], and our HIST loss still achieved state-of-the-art performance (see the supplementary material).

5. Conclusion

In this paper, we proposed Hypergraph-Induced Semantic Tuplet (HIST) loss for deep metric learning that leverages multilateral semantic relations provided by the semantic tuplets via hypergraph modeling. First, we proposed learnable prototypical distributions to automatically construct the semantic tuplets from a mini-batch, avoiding the excessive computational burden for tuple mining. Then, we formulated the hypergraph-based learning objective employing a hypergraph neural network. Compared to previous graph-based losses, our HIST loss takes advantage of multilateral semantic relations beyond pairwise feature relations. By leveraging multilateral semantic relations, HIST loss facilitates the embedding network to attend on meaningful object regions rather than background or distracting noises, contributing to better generalization performance and robustness against input corruptions. Extensive experimental results demonstrated the effectiveness of HIST loss, and a standard model trained with HIST loss achieved state-of-the-art performances, under both standard and MLRC evaluation settings, for three benchmark datasets.

Acknowledgement

This research was supported by IITP grant from Korea government (MSIT): [No.B0101-15-0266, Development of High Performance Visual Big Data Discovery Platform] and [NO.2021-0-01343, AI Graduate School Program (SNU)].

References

[1] Nicolas Aziere and Sinisa Todorovic. Ensemble deep manifold similarity learning using hard proxies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7299–7307, 2019. 1, 2

[2] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006. 2

[3] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1861–1870, 2019. 8

[4] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2017. 1

[5] Sumit Chopra, Raia Hadsell, and Yann 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, pages 539–546. IEEE, 2005. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[7] Ismail Elezi, Sebastiano Vascon, Alessandro Torcinovich, Marcello Pelillo, and Laura Leal-Taixé. The group loss for deep metric learning. In *European Conference on Computer Vision*, pages 277–294. Springer, 2020. 1, 2, 8

[8] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019. 2, 3, 5

[9] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285, 2018. 8

[10] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017. 2

[11] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *Advances in neural information processing systems*, 17, 2004. 2, 3

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 7

[13] Geonmo Gu, Byungsoo Ko, and Han-Gyu Kim. Proxy synthesis: Learning with synthetic classes for deep metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 1

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 1

[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

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[17] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 1

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[19] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015. 1, 2

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 5

[21] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3238–3247, 2020. 1, 2, 5, 6, 8

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 2, 5

[23] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 6

[24] Jongin Lim, Daeho Um, Hyung Jin Chang, Dae Ung Jo, and Jin Young Choi. Class-attentive diffusion network for semi-supervised classification. In *AAAI'21 Proceedings of the Thirty-fifth AAAI Conference on Artificial Intelligence*. AAAI Press, 2020. 6

[25] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 1

[26] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016. 1, 2

[27] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016. 5

[28] Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation for deep metric learning. In *European Conference on Computer Vision*, pages 590–607. Springer, 2020. 8

[29] 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. 1, 2

[30] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020. 6, 8

[31] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016. 2, 3, 4, 5, 8

[32] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458, 2019. 1, 2, 5, 8

[33] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. Simultaneous similarity-based self-distillation for deep metric learning. In *International Conference on Machine Learning*, pages 9095–9106. PMLR, 2021. 8

[34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 2

[35] Jenny Seidenschwarz, Ismail Elezi, and Laura Leal-Taixé. Learning intra-batch connections for deep metric learning. *arXiv preprint arXiv:2102.07753*, 2021. 1, 2, 6, 7, 8

[36] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017. 1

[37] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016. 1, 2, 3, 4

[38] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020. 8

[39] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 1

[40] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 1, 2, 6, 8

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 6, 7

[42] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016. 1

[43] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 5

[44] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018. 1, 2

[45] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 1, 2

[46] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 1, 2, 5, 8

[47] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5207–5216, 2019. 8

[48] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 1, 2

[49] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017. 1

[50] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3415–3424, 2017. 1

[51] Furong Xu, Meng Wang, Wei Zhang, Yuan Cheng, and Wei Chu. Discrimination-aware mechanism for fine-grained representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 813–822, 2021. 8

[52] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6490–6499, 2019. 2, 3, 4, 8

[53] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *Proceedings of the IEEE international conference on computer vision*, pages 814–823, 2017. 1

[54] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. *arXiv preprint arXiv:1811.12649*, 2018. 1, 2, 8

[55] Yanfu Zhang, Lei Luo, Wenhan Xian, and Heng Huang. Learning better visual data similarities via new grouplet non-euclidean embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9918–9927, 2021. 1

[56] Wenzhao Zheng, Chengkun Wang, Jiwen Lu, and Jie Zhou. Deep compositional metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9320–9329, 2021. 8

[57] Wenzhao Zheng, Borui Zhang, Jiwen Lu, and Jie Zhou. Deep relational metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12065–12074, 2021. 8

[58] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004. 2

[59] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003. 2

[60] Yuehua Zhu, Muli Yang, Cheng Deng, and Wei Liu. Fewer is more: A deep graph metric learning perspective using fewer proxies. In *NeurIPS*, 2020. 1, 2, 8