

Self-Taught Metric Learning without Labels

Sungyeon Kim¹

Dongwon Kim¹

Minsu Cho^{1,2}

Suha Kwak^{1,2}

Dept. of CSE, POSTECH¹

Graduate School of AI, POSTECH²

{sungyeon.kim, kdwon, mscho, suha.kwak}@postech.ac.kr

<http://cvlab.postech.ac.kr/research/STML/>

Abstract

We present a novel self-taught framework for unsupervised metric learning, which alternates between predicting class-equivalence relations between data through a moving average of an embedding model and learning the model with the predicted relations as pseudo labels. At the heart of our framework lies an algorithm that investigates contexts of data on the embedding space to predict their class-equivalence relations as pseudo labels. The algorithm enables efficient end-to-end training since it demands no off-the-shelf module for pseudo labeling. Also, the class-equivalence relations provide rich supervisory signals for learning an embedding space. On standard benchmarks for metric learning, it clearly outperforms existing unsupervised learning methods and sometimes even beats supervised learning models using the same backbone network. It is also applied to semi-supervised metric learning as a way of exploiting additional unlabeled data, and achieves the state of the art by boosting performance of supervised learning substantially.

1. Introduction

Understanding similarities between data is at the heart of many machine learning tasks such as data retrieval [29, 37, 47, 48], face verification [35, 45], person re-identification [7, 57], few-shot learning [41, 46, 49], and representation learning [29, 53, 62]. Metric learning embodies the perception of similarity by learning an embedding space where the distance between a pair of data represents their inverse semantic similarity. Also, the mapping from data to such a space is typically modeled by deep neural networks.

Recent advances in metric learning rely heavily on supervised learning using large-scale datasets. Manual annotation of such datasets is however costly, and thus could limit the class diversity of training data and in consequence the generalization capability of learned models. Unsupervised metric learning has been studied to resolve this issue. Exist-

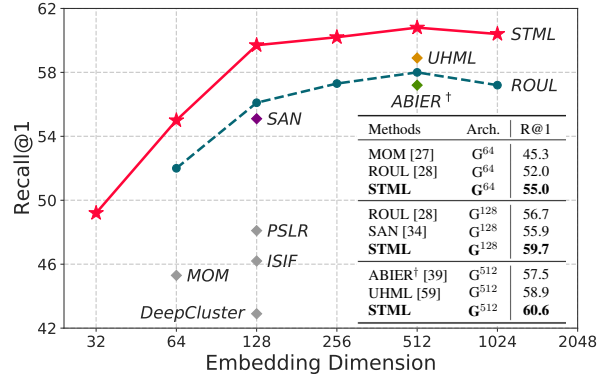


Figure 1. Accuracy in Recall@1 versus embedding dimension on the CUB-200-2011 [55] dataset using GoogleNet [50] backbone. Superscripts denote embedding dimensions and † indicates supervised learning methods. Our model with 128 embedding dimensions outperforms all previous arts using higher embedding dimensions and sometimes surpasses supervised learning methods.

ing methods in this line of research mainly synthesize class information of training data by assigning a surrogate class per training instance [13, 56, 60, 61] or discovering pseudo classes through k -means clustering [3, 4, 26, 33, 34, 58], hierarchical clustering [58], or random walk [25]. Although these methods have demonstrated impressive results without using groundtruth labels in training, they often fail to capture intra-class variation [13, 56, 60, 61] or impose substantial computational burden due to the off-the-shelf techniques [3, 4, 25, 26, 33, 34, 58].

In this paper, we propose a new method for unsupervised metric learning that addresses the aforementioned limitations of previous work and achieves the state of the art as shown in Fig. 1. The major contribution of this paper is two-fold. First, we introduce a novel, end-to-end self-taught metric learning framework (STML), whose overall pipeline is illustrated in Fig. 2. Unlike the existing works, it predicts class-equivalence relations between data within each mini-batch by *self-exploration* and leverages the predicted relations as synthetic supervision for metric learning. Our framework manages two embedding networks, namely

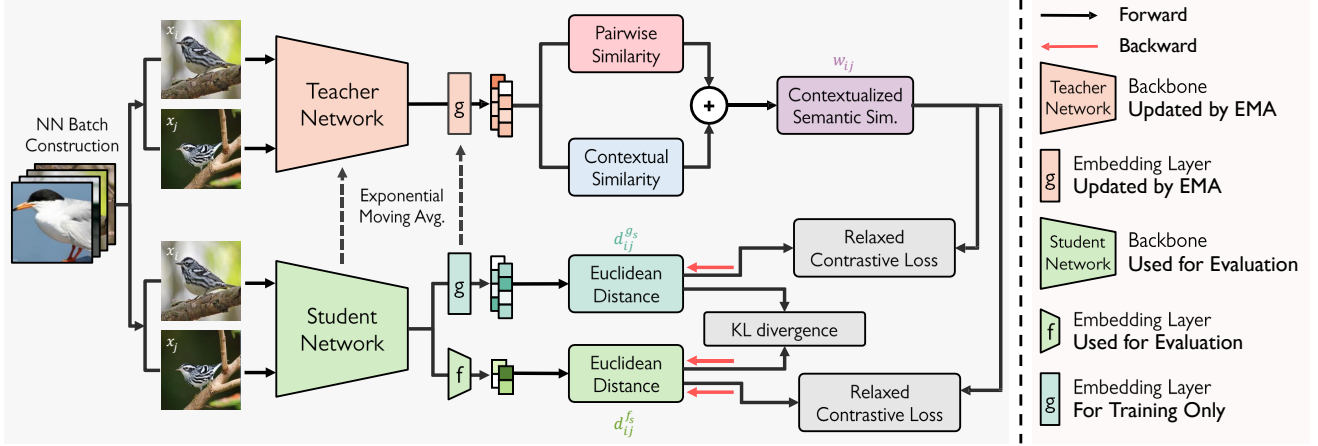


Figure 2. An overview of our STML framework. First, contextualized semantic similarity between a pair of data is estimated on the embedding space of the teacher network. The semantic similarity is then used as a pseudo label, and the student network is optimized by relaxed contrastive loss with KL divergence. Pink arrows represent backward gradient flows. Finally, the teacher network is updated by an exponential moving average of the student. The student network learns by iterating these steps a number of times, and its backbone and embedding layer in light green are considered as our final model.

teacher and student whose backbone are initialized identically. Class-equivalence between a pair of data is approximately estimated as their semantic similarity on the embedding space of the teacher network. The predicted similarity is used as a soft pseudo label for learning the student model, and the teacher model is in turn updated by a momentum-based moving average of the student. Iterating this process evolves the student model progressively.

The success of STML depends heavily on the quality of predicted semantic similarities, and our second contribution lies in the way of *estimating semantic similarities using contexts*. Specifically, given a pair of data, we compute their semantic similarity considering the overlap of their contexts (*i.e.*, neighborhoods in the embedding space) as well as their pairwise distance. We found that the contextualized semantic similarity approximates class-equivalence precisely. Further, since it has a real value indicating the degree of semantic similarity, it provides rich information beyond binary class-equivalence. Also, since it requires no external module nor memory banks [20, 31], it makes STML efficient and concise. To further enhance the quality of predicted semantic similarity, we design the teacher network to learn a higher dimensional embedding space than the student counterpart. This asymmetric design of the two networks allows the teacher to provide more effective supervision thanks to its improved expression power while the student, *i.e.*, our final model, remains compact.

To the best of our knowledge, STML is the only unsupervised metric learning method that can consider semantic relations between data in end-to-end training without introducing off-the-shelf techniques. Compared to the instance-level surrogate classes [13, 56, 60, 61], pseudo labels generated and exploited by STML are more appropriate

to capture semantic relations between data since they indicate class-equivalence relations. Also, unlike previous work based on pseudo labeling [3, 4, 25, 26, 33, 58], STML employs no external algorithm and thus allows training to be efficient, end-to-end, and less sensitive to hyper-parameters. Further, it is naturally applied to semi-supervised metric learning [14] as well as unsupervised metric learning.

We first evaluate STML on standard benchmarks for metric learning [32, 48, 55], where it largely outperforms existing unsupervised learning methods. Surprisingly, sometimes it even beats some of supervised learning models using the same backbone network as shown in Fig. 1. Beyond that, its efficacy is demonstrated on two benchmarks for semi-supervised metric learning [14], where it substantially outperforms previous work as well.

2. Related Work

Unsupervised metric learning has been mainly addressed in two different directions, instance discrimination [13, 56, 60, 61] and pseudo labeling [3, 4, 25, 26, 33, 34, 58]. Following the contrastive learning strategy [2, 6, 8, 20], instance discrimination methods assign a unique label per training instance and learn embedding spaces where different instances are well discriminated. Unfortunately, they have a trouble modeling variations within each latent class. On the other hand, pseudo labeling methods discover pseudo classes by applying off-the-shelf algorithms like k -means clustering [3, 4, 26, 33, 34], hierarchical clustering [58] and random walk [25] to unlabeled training data. Such methods can consider the class-equivalence relations between training data by utilizing existing supervised metric learning losses [19, 45, 54] with pseudo labels. How-

ever, the major drawback of these methods is their prohibitively high complexity imposed by the auxiliary algorithms, which obstructs end-to-end training.

Semi-supervised metric learning, recently introduced by Duan *et al.* [14], assumes that a training dataset consists of both labeled and unlabeled data. The previous work discovers and assigns pseudo labels to unlabeled training data via k -means clustering. Unfortunately, it also suffers from the aforementioned drawbacks of the unsupervised learning using pseudo labeling.

Self-supervised representation learning aims at learning high-quality intermediate features to generalize to other downstream tasks. To this end, various annotation-free pretext tasks have been proposed, *e.g.*, rotation prediction [17], solving jigsaw puzzle [38], image colorization [63], and image inpainting [40]. Pseudo labeling also has been widely used as it allows to learn models using conventional losses; existing methods in this direction discover pseudo classes by clustering [4] or treat each training instance as a surrogate class [6, 13, 20, 56]. Also, as an extension of the surrogate class, nearest neighbors have been leveraged as a positive set for self-supervised learning [15, 23, 31], but they often produce noisy labels unless a sufficiently trained embedding space is provided in advance.

3. Proposed Method

This section first presents an overview of our self-taught metric learning framework, dubbed STML. We then describe the each step of STML in detail and how to estimate the contextualized semantic similarity, which is the synthetic supervision for our framework.

3.1. Overview

STML enables efficient end-to-end training without using any off-the-shelf technique, while capturing semantic relations between data by generating and exploiting pseudo labels that approximate their class-equivalence.

Fig. 2 illustrates the overall pipeline of STML. It manages two embedding networks with the identically initialized backbone: a *teacher* and a *student*. The teacher network is used to estimate semantic similarities between data as their approximate class-equivalence relations, which are in turn used as synthetic supervision for training the student network. In particular, the teacher model leverages a high-dimensional (*e.g.*, 1024 or 2048) embedding layer g^t with the backbone encoder ϕ^t of the teacher. The use of the high dimensional space allows the teacher embedding layer to encode richer information and produce more reliable semantic similarity. Our student model has two parallel embedding layers f^s and g^s that share the backbone encoder ϕ^s . f^s is the embedding layer of our final model, and has a lower dimension (*e.g.*, 128 or 512) to learn a compact embedding space. g^s is an auxiliary layer dedicated to

Algorithm 1 Self-taught metric learning

Input: teacher model t , student model s , kernel bandwidth σ , momentum m , batch size n .

```

1: set  $\theta_t = \theta_{\phi^t} \cup \theta_{g^t}, \theta_s = \theta_{\phi^s} \cup \theta_{g^s}$ 
2: Identically initialize the backbone of  $t$  and  $s$ .
3: for number of epochs do
4:   Construct mini-batches using nearest neighbor batch construction. ▷ Step #1
5:   for number of iterations do
6:     for all samples in a mini-batch  $\{x_k\}_{k=1}^n$  do
7:        $z_k^t \leftarrow (g^t \circ \phi^t)(x_k)$ 
8:        $z_k^{f^s} \leftarrow (f^s \circ \phi^s)(x_k), z_k^{g^s} \leftarrow (g^s \circ \phi^s)(x_k)$ 
9:     end for
10:    for all  $i \in \{1, \dots, n\}$  and  $j \in \{1, \dots, n\}$  do ▷ Step #2
11:       $w_{ij}^P \leftarrow \exp(-\|z_i^t - z_j^t\|_2^2 / \sigma)$ 
12:      compute  $w_{ij}^C$  using Eq. (7, 8, 9).
13:       $w_{ij} \leftarrow \frac{1}{2}(w_{ij}^P + w_{ij}^C)$ 
14:    end for
15:    compute  $\mathcal{L}_{\text{STML}}$  using Eq. (3) and optimize  $t$ . ▷ Step #3
16:     $\theta_t \leftarrow m\theta_t + (1 - m)\theta_s$  ▷ Step #4
17:  end for
18: end for
19: return student model  $s$ 
```

continually updating the teacher embedding layer g^t , and thus having the same output dimension with that of g^t .

Algorithm 1 describes the procedure of STML in detail. First, the teacher network estimates semantic similarities that approximate class-equivalence relations between data, which are in turn used as synthetic supervision for training the student network. Then the teacher network is updated by the momentum-based moving average of the student. Our final model is the student network obtained by iterating this alternating optimization a predefined number of times. In the next section, we will elaborate each step of STML.

3.2. Self-Taught Metric Learning Framework

Step #1: Nearest neighbor batch construction. The way of constructing mini-batches for training has a massive impact on performance in metric learning [16, 44]. One of the most widely used methods for batch construction is the pk -sampling: It randomly chooses p unique classes, and then randomly samples k instances per class. Since this method leverages class labels of training data, it is however prohibited in an unsupervised learning setting. We instead propose a new strategy based on the nearest neighbor search. For each mini-batch, it randomly samples q queries and then searches for $k - 1$ nearest neighbors of each query; including both queries and their nearest neighbors, the mini-batch becomes of size qk . Our strategy is simple, fast, yet allows mini-batches to include diverse and relevant samples.

Step #2: Generating synthetic supervision. Supervised metric learning methods in general employ as supervision class-equivalence relations between data, which are how-

ever not available in our setting. STML instead approximates the class-equivalence relations by semantic similarities between data on the teacher embedding space, which are used as synthetic soft supervision for learning the student model. The major challenge is that the teacher embedding space is not mature enough to grasp the semantic relations between data at early stages of unsupervised learning. We thus propose contextualized semantic similarity, which measures the similarity between a pair of data while considering their contexts on the teacher embedding space; its details are presented in Sec. 3.3.

Step #3: Learning the student model. We simultaneously optimize the two branches of the student model using the contextualized semantic similarities as synthetic supervision. Since its value is soft, however, the contextualized semantic similarity cannot be incorporated with the ordinary metric learning losses that take discrete labels as supervision. We thus employ the relaxed contrastive loss [28], which is designed to utilize soft relational labels for metric learning. Let w_{ij} be the contextualized semantic similarity between x_i and x_j , and $z_i^{f_s}$ denote the student embedding vector of x_i generated by f_s . Defining $d_{ij}^{f_s} := \|z_i^{f_s} - z_j^{f_s}\|_2 / (\frac{1}{n} \sum_{k=1}^n \|z_i^{f_s} - z_k^{f_s}\|_2)$ as the relative distance between f_i^s and f_j^s , the loss is then given by

$$\begin{aligned} \mathcal{L}_{\text{RC}}(\mathcal{Z}^{f_s}) &= \frac{1}{n} \sum_{i=1}^n \sum_{j \neq i}^n w_{ij} (d_{ij}^{f_s})^2 \\ &+ \frac{1}{n} \sum_{i=1}^n \sum_{j \neq i}^n (1 - w_{ij}) [\delta - d_{ij}^{f_s}]_+^2, \end{aligned} \quad (1)$$

where \mathcal{Z}^{f_s} is all student embedding vectors generated by f_s , n is the number of samples in the batch, and δ is a margin. The contextualized semantic similarity serves as the weights for the attracting and repelling terms of the loss, and thus determines the magnitude of the force that pulls or pushes a pair of data on the student embedding space. Note that g_s is trained in the same manner on f_s . We additionally apply self-distillation to further utilize complex higher dimensional information for lower dimensional space. Following [43], we use Kullback-Leibler divergence as objective of self-distillation:

$$\mathcal{L}_{\text{KL}}(\mathcal{Z}^{f_s}, \mathcal{Z}^{g_s}) = \frac{1}{n} \sum_{i=1}^n \sum_{j \neq i}^n \psi(-d_{ij}^{g_s}) \log \frac{\psi(-d_{ij}^{g_s})}{\psi(-d_{ij}^{f_s})}, \quad (2)$$

where $\psi(\cdot)$ is softmax operation. Note that the gradient-flow of g_s is truncated. In summary, STML trains the two branches f_s and g_s of the student model by minimizing the overall loss defined as

$$\begin{aligned} \mathcal{L}_{\text{STML}}(\mathcal{Z}^{f_s}, \mathcal{Z}^{g_s}) &= \frac{1}{2} [\mathcal{L}_{\text{RC}}(\mathcal{Z}^{f_s}) + \mathcal{L}_{\text{RC}}(\mathcal{Z}^{g_s})] \\ &+ \mathcal{L}_{\text{KL}}(\mathcal{Z}^{f_s}, \mathcal{Z}^{g_s}). \end{aligned} \quad (3)$$

Step #4: Momentum update of the teacher model. The teacher model should be updated as the student model is trained so that the synthetic supervision computed from the teacher embedding space is enhanced progressively. The simplest update strategy is to replace the teacher model with the student counterpart at every iteration. However, this could deteriorate the consistency of the synthetic supervision over iterations, which leads to unstable training of the student model. We thus update the teacher model by the momentum-based moving average of the student, except for the embedding layer f^s since the teacher model has only a high-dimensional one. Let θ_t be the parameters of the teacher model and θ_t be the parameters of ϕ_t and g_t . Then θ_t is updated by

$$\theta_t \leftarrow m\theta_t + (1 - m)\theta_s, \quad (4)$$

where $m \in [0, 1]$ is the coefficient that controls the momentum update rate, slowing down the update of the teacher model as its value increases.

3.3. Contextualized Semantic Similarity

The quality of predicted semantic similarities is of vital importance in STML since they are the only supervision for training. However, it is not straightforward to compute reliable semantic similarities especially at early stages of unsupervised learning where the teacher network is significantly incomplete. To overcome this challenge, we propose to use *context*, i.e., neighborhood on a data manifold.

Our key idea is not to rely on the relations considering two images of interest only, but to figure out their indirect semantic relations via their contexts. Their multiple indirect relations can contribute to capturing common patterns frequently observed within each class and thus understanding the underlying class-equivalence relations. To this end, we design the *contextualized semantic similarity*. Given a pair of samples, their contextualized semantic similarity is defined as a combination of their *pairwise similarity* and *contextual similarity* on the teacher embedding space.

Pairwise similarity. The pairwise similarity of two samples x_i and x_j , denoted by w_{ij}^P , is given by

$$w_{ij}^P = \exp \left(- \frac{\|z_i^t - z_j^t\|_2^2}{\sigma} \right), \quad (5)$$

where z_i^t is the teacher embedding vector of x_i and σ is the Gaussian kernel bandwidth. This kind of similarity has been known to encode rich information about relations of x_i and x_j when the teacher embedding space is pre-trained in a fully supervised manner [28], but is often broken in unsupervised metric learning. We thus supplement it with the contextual similarity, which is detailed below.

Contextual similarity. The underlying assumption of the contextual similarity is that the more two samples semantically similar, the larger their contexts overlap. One may

consider nearest neighbors of a data point on the teacher embedding space as its context, but they often include irrelevant data when the embedding space is not sufficiently trained yet. We thus instead consider k -reciprocal nearest neighbors as context since they are known to discover highly correlated candidates [42, 64]. The set of k -reciprocal nearest neighbors of a data point x_i is given by

$$R_k(i) = \{j | (j \in N_k(i)) \wedge (i \in N_k(j))\}, \quad (6)$$

where $N_k(i)$ is the k -nearest neighbors of x_i , including itself (*i.e.*, $i \in N_k(i)$), in the current mini-batch. The initial form of contextual similarity measures the degree of overlap between $R_k(i)$ and $R_k(j)$ while considering whether x_i and x_j are in k -reciprocal nearest neighbor relation. Since the size of the neighbor set of x_i and x_j would be different, we design an asymmetric Jaccard similarity as

$$\tilde{w}_{ij}^C = \begin{cases} \frac{|R_k(i) \cap R_k(j)|}{|R_k(i)|}, & \text{if } j \in R_k(i), \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

In addition, we adopt the idea of query expansion [1, 10, 11] to further improve its reliability. Specifically, \tilde{w}_{ij}^C is reformulated as the average of the contextual similarities between x_j and nearest neighbors of x_i as follows:

$$\hat{w}_{ij}^C = \frac{1}{|N_{\frac{k}{2}}(i)|} \sum_{h \in N_{\frac{k}{2}}(i)} \tilde{w}_{hj}^C. \quad (8)$$

The expanded version in Eq. (8) is not symmetric anymore. To ensure symmetry, the final form of the contextual similarity is defined as the average of \hat{w}_{ij}^C and its transpose, which is given by

$$w_{ij}^C = w_{ji}^C = \frac{1}{2}(\hat{w}_{ij}^C + \hat{w}_{ji}^C). \quad (9)$$

Contextualized semantic similarity. Finally, the contextualized semantic similarity between x_i and x_j is defined as the average of their pairwise similarity in Eq. (5) and contextual similarity in Eq. (9):

$$w_{ij} = \frac{1}{2}(w_{ij}^P + w_{ij}^C). \quad (10)$$

Note that both of w_{ij}^P and w_{ij}^C lie within $[0, 1]$ thus their scales are balanced. The contextualized semantic similarity effectively represents semantic relatedness between data by considering both of their pairwise relation and context. We found that the contextualized semantic similarity is highly correlated with groundtruth class-equivalence, which is empirically verified in Sec. 4.1.4.

4. Experiments

This section validates the effectiveness of STML on two metric learning tasks with lack of supervision, *i.e.*, unsupervised metric learning and semi-supervised metric learning.

4.1. Unsupervised Metric Learning

We consider the student embedding model trained by STML as our final model, which is evaluated and compared with state-of-the-art unsupervised metric learning methods and representative self-supervised learning methods.

4.1.1 Experimental Setup

Datasets and evaluation. The models are evaluated and compared on three benchmark datasets, *i.e.*, CUB-200-2011 (CUB) [55], Cars-196 (Cars) [32], and Stanford Online Product (SOP) [48]; for their train-test splits, we directly follow the standard protocol presented in [61]. Performance on these datasets is evaluated by Recall@ k , the fraction of queries that have at least one relevant sample in their k -nearest neighbors on a learned embedding space.

Embedding networks. For comparisons to previous work, we employ GoogleNet [50] and Inception-BN [24] as our teacher and student embedding model architecture. The dimension of embedding layers g_t and g_s are set as the output size of the last pooling, and that of f_s follows each experimental setting. We append an l_2 normalization layer on top of the teacher so that the Euclidean distances in Eq. (5) are bounded for computing semantic similarities stably.

Implementation details. The student model of STML is optimized by AdamP with Nesterov momentum [22] for 90 epochs on a single Titan RTX; the learning rate is initialized to 10^{-4} and scaled down by the cosine decay function [36]. Training images are randomly cropped to 227×227 and flipped horizontally at random while test images are resized to 256×256 then center cropped. For all experiments, σ in Eq. (1) is set to 3. The margin δ , momentum coefficient m and the number of nearest neighbors k are 1.0, 0.999 and 10 on the CUB and Cars datasets, and 0.9, 0.9 and 4 on the SOP dataset. Note that k in the nearest neighbor batch construction and that in Eq. (6) are identical.

Batch construction. Nearest neighbors used to construct mini-batches in STML are computed on the student embedding space and update at every epoch. On the CUB and Cars datasets, each mini-batch consists of 120 images including 24 queries and their 4 nearest neighbors. Since the SOP dataset has a tiny number of samples per class, we construct a mini-batch with 60 queries and their top-1 neighbors on the dataset. Also, all images are randomly augmented twice following the multi-view augmentation strategy in [28].

4.1.2 Quantitative Results

For fair comparisons to previous work, performance evaluation is done in the following settings: GoogleNet with 64/128/512 embedding dimensions, and Inception with BatchNorm with 512 embedding dimensions. All backbone networks in this experiment are pre-trained on ImageNet. The results on the three datasets are summarized in Table 1.

Methods	Arch.	CUB			Cars			SOP		
		R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100
<i>Supervised</i>										
ABIER [39]	G ⁵¹²	57.5	69.7	78.3	82.0	89.0	93.2	74.2	86.9	94.0
ABE [30]	G ⁵¹²	63.0	74.5	83.3	85.2	90.5	94.0	76.3	88.4	94.8
MS [54]	BN ⁵¹²	65.7	77.0	86.3	84.1	90.4	94.0	78.2	90.5	96.0
PA [27]	BN ⁵¹²	69.1	78.9	86.1	86.4	91.9	95.0	79.2	90.7	96.2
<i>Unsupervised (Using ImageNet pre-trained embedding networks)</i>										
MOM [25]	G ⁶⁴	45.3	57.8	68.6	-	-	-	-	-	-
ROUL [26]	G ⁶⁴	52.0	64.0	74.5	-	-	-	-	-	-
STML	G ⁶⁴	55.0	67.5	78.6	42.0	54.4	65.9	64.9	79.6	89.3
MOM [25]	G ¹²⁸	-	-	-	35.5	48.2	60.6	43.3	57.2	73.2
Exemplar [13]	G ¹²⁸	38.2	50.3	62.8	36.5	48.1	59.2	45.0	60.3	75.2
NCE [56]	G ¹²⁸	39.2	51.4	63.7	37.5	48.7	59.8	46.6	62.3	76.8
DeepCluster [4]	G ¹²⁸	42.9	54.1	65.6	32.6	43.8	57.0	34.6	52.6	66.8
ISIF [61]	G ¹²⁸	46.2	59.0	70.1	41.3	52.3	63.6	48.9	64.0	78.0
PSLR [60]	G ¹²⁸	48.1	60.1	71.8	43.7	54.8	66.1	51.1	66.5	79.8
ROUL [26]	G ¹²⁸	56.7	68.4	78.3	45.0	56.9	68.4	53.4	68.8	81.7
SAN [34]	G ¹²⁸	55.9	68.0	78.6	44.2	55.5	66.8	58.7	73.1	84.6
STML	G ¹²⁸	59.7	71.2	81.0	49.0	60.4	71.3	65.8	80.1	89.9
UDML-SS [3]	G ⁵¹²	54.7	66.9	77.4	45.1	56.1	66.5	63.5	78.0	88.6
TAC-CCL [33]	G ⁵¹²	57.5	68.8	78.8	46.1	56.9	67.5	63.9	77.6	87.8
UHML [58]	G ⁵¹²	58.9	70.6	80.4	47.7	58.9	70.3	65.1	78.2	88.3
STML	G ⁵¹²	60.6	71.7	81.5	50.5	61.8	71.7	65.3	79.8	89.8
UDML-SS [3]	BN ⁵¹²	63.7	75.0	83.8	-	-	-	-	-	-
STML	BN ⁵¹²	68.0	78.8	86.4	66.2	74.5	81.9	69.7	82.7	91.2

Table 1. Performance of the unsupervised and supervised metric learning methods on the three datasets. Their network architectures are denoted by abbreviations, G–GoogleNet [50], BN–Inception with BatchNorm [24], where superscripts denote their embedding dimensions.

Methods	SOP		
	R@1	R@10	R@100
NCE [56]	34.4	49.0	65.2
ISIF [61]	39.7	54.9	71.0
PSLR [60]	42.3	57.7	72.5
ROUL [26]	45.4	60.5	74.8
SAN [34]	46.3	61.9	77.0
STML	60.7	74.8	85.2

Table 2. Performance on the SOP dataset using ResNet18 without pre-trained weights.

On all the datasets, our model achieves state-of-the-art performance in every setting. It improves the previous best score by a large margin, from 3.8% to 7.1% at Recall@1 with the 128 embedding dimensions. We note that our model clearly surpasses all existing models whose embedding dimension is 4 times higher on all the datasets, sometimes at 8 times higher. Moreover, on the CUB dataset, it even outperforms some of the supervised learning methods, e.g., ABIER [39] and MS [54], given the same embedding networks. These results suggest that our framework allows the model to learn compact yet effective embedding spaces by providing high-quality synthetic supervision. Last but

not least, STML enables more efficient learning than existing pseudo labeling methods due to no use of off-the-shelf algorithms, especially on large-scale SOP dataset.

4.1.3 Unsupervised Metric Learning from Scratch

We demonstrate that STML is also effective for metric learning from scratch without pre-training on ImageNet. Following [61], we employ randomly initialized ResNet18 [21] with 128 embedding dimensions as the backbone network, and evaluate our model trained on the SOP dataset. As shown in Table 2, our model substantially outperforms existing methods including those for self-supervised representation learning in this setting. This result suggests that STML can generate reliable pseudo labels even on a randomly initialized embedding space, and our model trained by STML well generalizes to unseen class.

4.1.4 In-depth Analysis

Correlation between similarities and class-equivalence.

To justify the contextualized semantic similarity, we compare our three similarity metrics (i.e., pairwise similarity w_{ij}^P , contextual similarity w_{ij}^C , contextualized semantic sim-

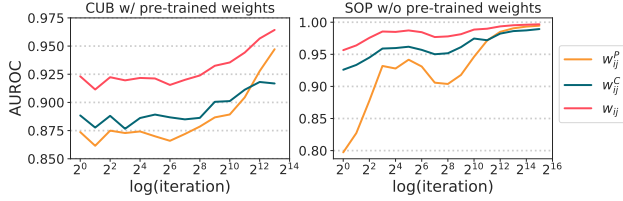


Figure 3. Comparison between the contextualized semantic similarity and its two components in terms of class-equivalence prediction accuracy on the CUB dataset with pre-trained weights (*left*) and SOP dataset without pre-trained weights (*right*).

	CUB	Cars
Methods	R@1	R@1
STML	60.6	50.5
w/o contextual pairwise similarity	44.2	40.2
w/o pairwise similarity	56.0	45.0
w/o relaxed contrastive loss	53.4	38.7
w/o momentum update	56.8	46.1
w/o NN batch construction	54.6	44.8
w/o KL divergence	59.2	49.4

Table 3. Ablation study of the components in STML on the CUB and Cars datasets.

ilarity w_{ij}) in terms of how well they are correlated with class-equivalence, *i.e.*, the groundtruth for supervised metric learning. Specifically, we predict class-equivalence of every pair of data by thresholding its semantic similarities, and count the accuracy of such predictions at various threshold settings. Fig. 3 presents AUROC scores of our contextualized semantic similarity and its two components during training. The results suggest that the accuracy of the pairwise similarity relies heavily on the quality of the teacher embedding space, while the contextual similarity helps infer class-equivalence correctly even at early stages of training without pre-trained weights. The contextualized semantic similarity achieves the best consistently, which also indicates that the pairwise and contextual similarities are complementary to each other.

Ablation study. The impact of each component in STML is reported in Table 3. The results in the table are obtained by using GoogleNet with 512 embedding dimensions. The analysis reveals that the contextual similarity contributes most to the performance while the pairwise similarity has the minimum impact; Fig. 3 also supports this observation. Even if both similarities are used, the performance is largely degraded when using the conventional contrastive loss [19] instead of the relaxed version since the conventional one quantizes our semantic similarity into a binary value, leading to significant information loss. The momentum-based update of the teacher model and the nearest neighbor batch construction strategy also have significant impacts; the performance degrades substantially when the teacher model is

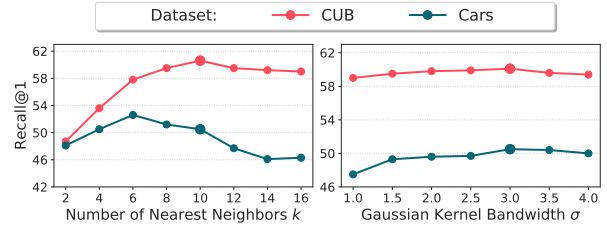


Figure 4. Recall@1 versus the hyper-parameters k and σ on the CUB and Cars datasets. Larger markers indicate our default experimental setting.

		CUB	Cars	SOP
Methods	Arch.	R@1	R@1	R@1
MoCo [20]	G ¹²⁸	48.3	37.2	53.3
BYOL [18]	G ¹²⁸	47.7	31.5	48.7
MSF [31]	G ¹²⁸	46.0	30.5	41.6
STML	G ¹²⁸	59.7	49.0	65.8
MoCo [20]	G ⁵¹²	51.0	39.0	53.4
BYOL [18]	G ⁵¹²	50.9	38.5	51.2
MSF [31]	G ⁵¹²	49.6	32.5	47.8
STML	G ⁵¹²	60.6	50.5	65.3

Table 4. Comparison between STML and recent self-supervised representation learning methods using momentum encoder.

updated directly by the student counterpart or the batch construction strategy is replaced with a random sampling. The KL divergence loss has relatively little influence compared to other components.

Effect of hyper-parameters. We investigate how much sensitive STML is to the hyper-parameters k and σ . Specifically, we measure performance of STML employing GoogleNet with 512 embedding dimensions on the CUB and Cars datasets while varying values of k and σ . The results shown in Fig. 4 suggest that the performance of STML is consistently high when $k < 14$, and σ has virtually no effect on the performance if $\sigma > 1$ on both datasets. Moreover, whenever $k < 14$ our model outperforms all existing unsupervised metric learning methods. Note that our final results reported in Table 1 are not the best scores in Fig. 4 since we did not tune the hyper-parameters on the test splits.

Comparison with recent SSL methods. STML has components used in recent self-supervised representation learning (SSL) methods such as the momentum encoder. To validate that the improvements of our framework do not originate from its structure, such methods are compared with STML under the same setting. Table 4 shows that MoCo and BYOL clearly underperform our method in all settings and achieve similar performance to unsupervised metric learning methods based on instance-level surrogate classes. Although they are structurally similar to STML, the reason for the degradation of performance is that they cannot capture intra-class variation unlike our method, and are

Methods	Arch.	Init.	CUB			Cars		
			MAP@R	RP	R@1	MAP@R	RP	R@1
<i>Supervised</i>								
Contrastive [9]	R50 ⁵¹²	ImageNet	25.0	35.8	65.3	26.0	36.4	81.2
MS [54]	R50 ⁵¹²	ImageNet	26.4	37.5	66.3	28.3	38.3	85.2
Proxy-Anchor [27]	R50 ⁵¹²	ImageNet	28.3	39.1	69.9	30.5	39.9	87.7
Contrastive [9]	R50 ⁵¹²	SwAV [5]	29.3	39.8	71.2	31.7	41.2	88.1
MS [54]	R50 ⁵¹²	SwAV [5]	29.2	40.2	70.8	33.4	42.7	89.3
Proxy-Anchor [27]	R50 ⁵¹²	SwAV [5]	31.7	42.0	74.6	35.1	44.1	90.4
<i>Semi-supervised (Labeled dataset + Additional unlabeled dataset)</i>								
SLADE (MS) [14]	R50 ⁵¹²	ImageNet	30.9	41.9	69.6	32.1	41.5	87.4
STML (MS)	R50 ⁵¹²	ImageNet	37.8	47.9	75.5	38.4	46.1	93.0
SLADE (MS) [14]	R50 ⁵¹²	SwAV [5]	33.9	44.4	74.1	38.0	46.9	91.5
STML (MS)	R50 ⁵¹²	SwAV [5]	35.2	45.3	76.2	38.2	46.5	93.2

Table 5. Performance of the supervised and semi-supervised methods on the two datasets. Network architectures of the methods are all ResNet50 (R50) [21] and superscripts denote their embedding dimensions. Also, the column “Init.” indicates whether the models are pre-trained on ImageNet or by SwAV. Our model and SLADE are all fine-tuned with the MS loss.

limited in representing semantic similarities between data.

4.2. Semi-supervised Metric Learning

Semi-supervised metric learning aims to improve performance of supervised metric learning by exploiting additional unlabeled data [14]. STML can be employed for this task naturally since it provides a way of using the unlabeled data for training by predicting their class-equivalence relations as synthetic supervision. In this section, we validate the efficacy of STML for semi-supervised metric learning and compare it with SLADE [14], the first and only existing method for the task, on the CUB and Cars datasets.

4.2.1 Experimental Setup

Datasets and evaluation. We directly adopt the evaluation protocol of SLADE. Specifically, we employ two combinations of labeled and unlabeled datasets, {CUB [55], NABirds [52]} and {Cars [32], CompCars [59]}, and three performance metrics, MAP@R, RP, and Recall@1.

Embedding networks. For comparisons to SLADE, we employ ResNet50 [21] with 512 embedding dimensions as the backbone network of the teacher and student models. Before applying STML, these models are pre-trained either on ImageNet [12] or by SwAV [5], then fine-tuned with the MS loss [54] on a labeled dataset (*i.e.*, CUB or Cars).

Implementation details. In this task, STML adopts an embedding model trained on a labeled dataset as the initial teacher and student models and trains them using both labeled and unlabeled data. Note that it computes and utilizes contextualized semantic similarities as supervision for labeled data as well as unlabeled data. For all experiments, m in Eq. (4) and σ in Eq. (10) are set to 0.9999 and 2, respectively, and the rest settings are the same as those for unsupervised metric learning.

4.2.2 Results

STML is compared with SLADE and state-of-the-art supervised metric learning methods on the two datasets. The results summarized in Table 5 show that STML improves performance of its supervised learning counterpart (*i.e.*, MS) significantly regardless of the model initialization schemes. In particular, it enhances the performance by more than 5% in Recall@1 on both datasets for ImageNet pre-trained setting. Also, STML clearly outperforms SLADE in all the settings except one, where their gap is only 0.4% in RP.

5. Conclusion

We have proposed STML, a novel end-to-end unsupervised metric learning framework that estimates and exploits semantic relations between samples as pseudo labels. The pseudo labels in STML are estimated by investigating contexts of data as well as their pairwise distances on an embedding space. STML has achieved the state of the art on the three benchmark datasets for unsupervised metric learning, and demonstrated impressive performance even without pre-trained weights. Moreover, it has been applied to semi-supervised learning, and outperformed state of the art on the two benchmark datasets. Unfortunately, like other approaches, STML has a limitation in that it uses ImageNet pre-trained weights in some settings. In the future, we will explore extensions of our method where class labels are never involved in the training.

Acknowledgement. This work was supported by the NRF grant, the IITP grant, and R&D program for Advanced Integrated-intelligence for IDentification, funded by Ministry of Science and ICT, Korea (NRF-2021R1A2C3012728, NRF-2018R1A5-A1060031, NRF-2018M3E3A1057306, No.2019-0-01906 Artificial Intelligence Graduate School Program-POSTECH).

References

- [1] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. [5](#)
- [2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2019. [2](#)
- [3] Xuefei Cao, Bor-Chun Chen, and Ser-Nam Lim. Unsupervised deep metric learning via auxiliary rotation loss. *arXiv preprint arXiv:1911.07072*, 2019. [1](#), [2](#), [6](#)
- [4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. [1](#), [2](#), [3](#), [6](#)
- [5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. [8](#)
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. International Conference on Machine Learning (ICML)*, 2020. [2](#), [3](#)
- [7] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [1](#)
- [8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. [2](#)
- [9] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. [8](#)
- [10] Ondřej Chum, Andrej Mikulík, Michal Perdoch, and Jiří Matas. Total recall ii: Query expansion revisited. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. [5](#)
- [11] Ondřej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2007. [5](#)
- [12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. [8](#)
- [13] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015. [1](#), [2](#), [3](#), [6](#)
- [14] Jiali Duan, Yen-Liang Lin, Son Tran, Larry Davis, and C-C Jay Kuo. Slade: A self-training framework for distance metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#), [3](#), [8](#)
- [15] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [3](#)
- [16] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott. Deep metric learning with hierarchical triplet loss. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. [3](#)
- [17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. [3](#)
- [18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Dohersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. [7](#)
- [19] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. [2](#), [7](#)
- [20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [2](#), [3](#), [7](#)
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [6](#), [8](#)
- [22] Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. AdamP: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. [5](#)
- [23] Jiabo Huang, Qi Dong, Shaogang Gong, and Xiatian Zhu. Unsupervised deep learning by neighbourhood discovery. In *Proc. International Conference on Machine Learning (ICML)*, 2019. [3](#)
- [24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning (ICML)*, 2015. [5](#), [6](#)
- [25] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Mining on manifolds: Metric learning without labels. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [1](#), [2](#), [6](#)
- [26] Shichao Kan, Yigang Cen, Yang Li, Vladimir Mladenovic, and Zhihai He. Relative order analysis and optimization for unsupervised deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [1](#), [2](#), [6](#)
- [27] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [6](#), [8](#)
- [28] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Embedding transfer with label relaxation for improved metric learning. In *Proc. IEEE Conference on Computer Vision*

- and Pattern Recognition (CVPR), 2021. 4, 5
- [29] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [30] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 6
- [31] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 7
- [32] 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, 8, 12
- [33] Yang Li, Shichao Kan, and Zhihai He. Unsupervised deep metric learning with transformed attention consistency and contrastive clustering loss. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 6
- [34] Yang Li, Shichao Kan, Jianhe Yuan, Wenming Cao, and Zhihai He. Spatial assembly networks for image representation learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 6
- [35] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphreface: Deep hypersphere embedding for face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1
- [36] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5
- [37] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. 1
- [38] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. European Conference on Computer Vision (ECCV)*, 2016. 3
- [39] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with bier: Boosting independent embeddings robustly. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 6
- [40] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3
- [41] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1
- [42] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 5
- [43] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. Simultaneous similarity-based self-distillation for deep metric learning. In *Proc. International Conference on Machine Learning (ICML)*, 2021. 4
- [44] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *Proc. International Conference on Machine Learning (ICML)*, 2020. 3
- [45] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2
- [46] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2017. 1
- [47] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2016. 1
- [48] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 5, 12
- [49] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [50] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 5, 6, 12
- [51] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 2014. 12
- [52] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 8
- [53] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 1
- [54] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6, 8
- [55] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 1, 2, 5, 8, 12
- [56] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3, 6
- [57] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

- [58] Jiexi Yan, Lei Luo, Cheng Deng, and Heng Huang. Unsupervised hyperbolic metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [1](#), [2](#), [6](#)
- [59] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [8](#)
- [60] Mang Ye and Jianbing Shen. Probabilistic structural latent representation for unsupervised embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#), [2](#), [6](#)
- [61] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [1](#), [2](#), [5](#), [6](#)
- [62] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [1](#)
- [63] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proc. European Conference on Computer Vision (ECCV)*, 2016. [3](#)
- [64] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [5](#)

A. Appendix

This appendix provides further analyses, and additional experimental results, all of which are left out from the main paper due to the space limit. In Section A.1, we first discuss properties of the contextualized semantic similarity in detail. Section A.2 presents *t*-SNE visualizations of the learned embedding space and qualitative examples of image retrieval on the three benchmark datasets.

A.1. In-depth Analysis on Contextualized Semantic Similarity

STML employs the contextualized semantic similarity as pseudo label that can capture semantic relations between data. To better understand its property, we illustrate in Fig. 5 how the contextualized semantic similarity is determined in four different cases of relations between samples. As shown in Fig. 5(b), when a pair of samples not only share many k -reciprocal nearest neighbors but also have a small Euclidean distance, they obviously have high contextualized semantic similarity; they are visually similar and placed on the same manifold, thus highly likely to belong to the same class. In contrast, as shown in Fig. 5(c), two samples that are far apart and share no neighbor have a low contextualized semantic similarity as they are semantically unrelated; we found that relations of most pairs of unlabeled data fall into this case. In Fig. 5(a) and 5(d), where only one of the pairwise and contextual similarities is high, two samples have a higher contextualized semantic similarity than that in the case of Fig. 5(c), but lower than that in the case of Fig. 5(b). Note that although the class-equivalence between samples is uncertain in the cases of Fig. 5(a) and 5(d), the pairwise and contextual similarities could be overly high in each of these cases, leading to noisy synthetic supervision. On the other hand, the contextualized semantic similarity is modest thus provides reliable pseudo labels in these cases.

The contextualized semantic similarity allows the relaxed contrastive loss to exploit reliable supervision. To demonstrate this, we empirically analyze the information provided by the contextualized semantic similarity. Fig. 6 presents the top-8 samples in a mini-batch of each query in terms of their contextualized semantic similarity on the three benchmark datasets [32, 48, 55]. The results are obtained by the source embedding model where the number of nearest neighbor k is 5. As shown in the figure, the contextualized semantic similarity is highly correlated with the groundtruth class-equivalence on all datasets. In particular, birds of the same class are assigned higher contextualized semantic similarities although all birds are floating on water with similar poses in the 2nd row of Fig. 6. Moreover, as can be seen in the 1st and 6th rows of Fig. 6, the contextualized semantic similarity successfully captures class-equivalence even under view-point variations.

A.2. Qualitative Results

To demonstrate the efficacy of the proposed method empirically, we present Barnes-Hut *t*-SNE visualizations [51] on the CUB-200-2011 [55], Cars-196 [32] and SOP [48] datasets in Fig. 7, 8, and 9, respectively. The visualizations of the embedding space learned by STML are compared with those of the embedding space before training (*i.e.*, ImageNet pre-trained) using GoogleNet [50] with 128 embedding dimensions. As shown in Fig. 7, the pre-trained model aggregates birds with similar colors to the query regardless of class. However, after being trained by STML, the model only clusters birds of the same class. Fig. 8 shows that the embedding space of our model places the samples of the same class close to each other despite different colors and viewpoint changes, unlike the embedding space of the pre-trained model where unrelated samples are aggregated. In Fig. 9, the pre-trained model clusters samples of similar categories, whereas our model successfully gathers samples belonging to some parts of the query image.

In addition, more qualitative retrieval results of our model at every 30 epochs of training on the three datasets are presented in Fig. 10, 11, and 12. As shown in the figures, at the beginning of training, the model tends to retrieve only samples that are similar in appearance to the query, without understanding their semantic relations. As training progresses, the model trained by STML distinguishes samples of the same class more accurately among visually similar samples.

All the results in these figures are obtained from unseen class samples without any manual annotation. The results suggest that contextualized semantic similarity estimates the class-equivalence relations between samples quite accurately, and thus STML allows the final embedding model to generalize well even to unseen classes through reliable pseudo labels considering semantic relations between data.

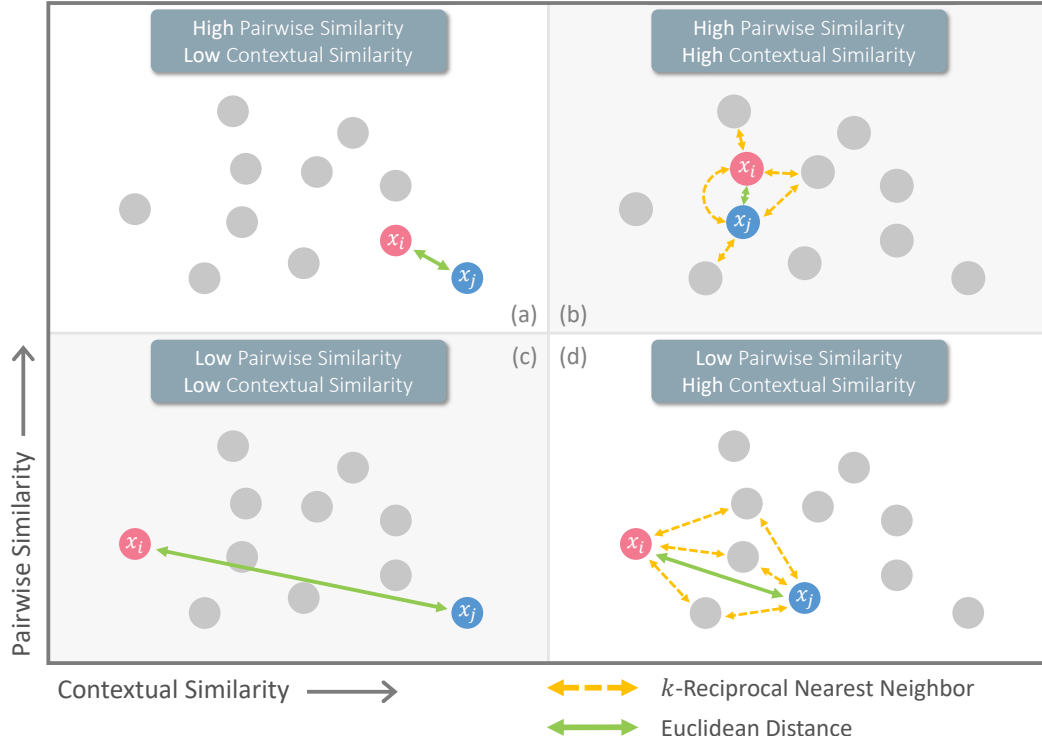


Figure 5. Difference in contextualized semantic similarity according to the relation between two samples.

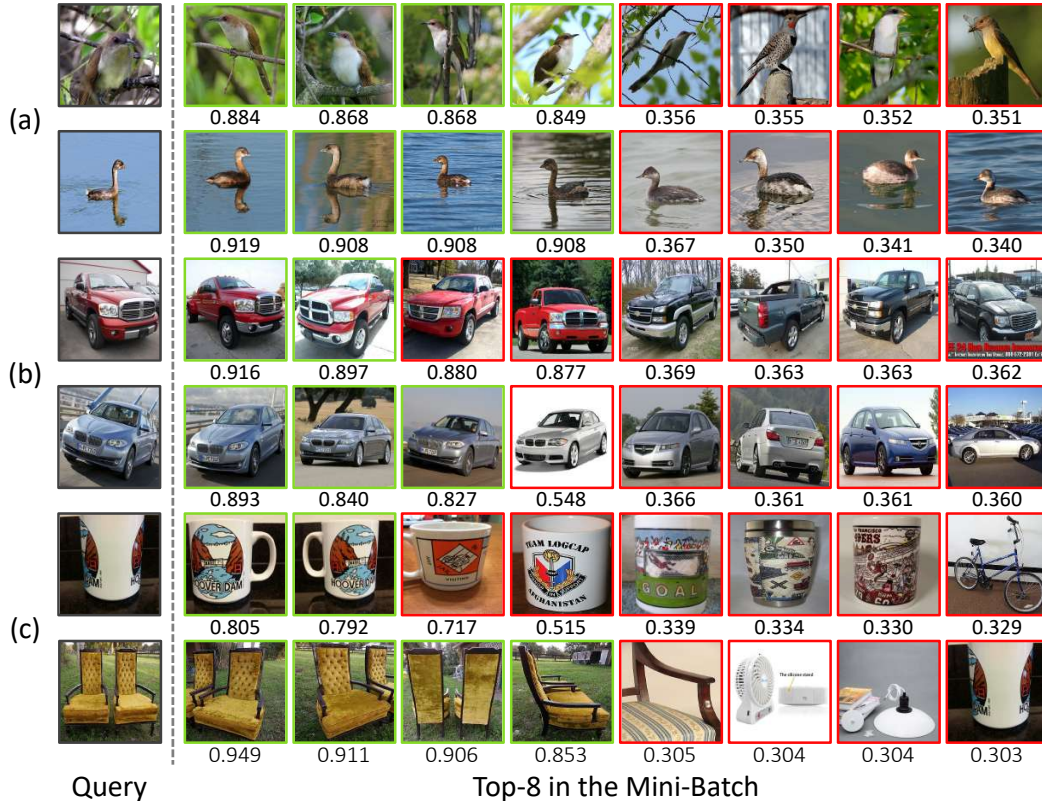


Figure 6. Image pairs sorted by their contextualized semantic similarity on (a) CUB-200-2011, (b) Cars-196, and (c) SOP datasets. Images with green boundary are of the same class as query and those with red boundary are of a different class from query.

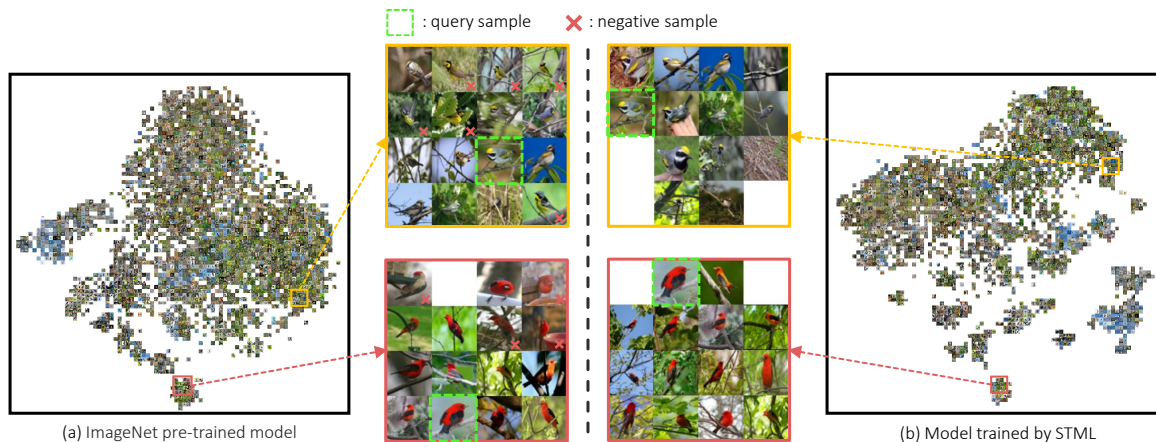


Figure 7. t -SNE visualization of (a) ImageNet pre-trained model and (b) model trained by STML on the test split of CUB-200-2011 dataset. A green dotted box indicates query samples, and a red cross mark indicates samples with different class labels from the query.



Figure 8. t -SNE visualization of (a) ImageNet pre-trained model and (b) model trained by STML on the test split of Cars-196 dataset. A green dotted box indicates query samples, and a red cross mark indicates samples with different class labels from the query.



Figure 9. t -SNE visualization of (a) ImageNet pre-trained model and (b) model trained by STML on the test split of SOP dataset. A green dotted box indicates query samples, and a red cross mark indicates samples with different class labels from the query.

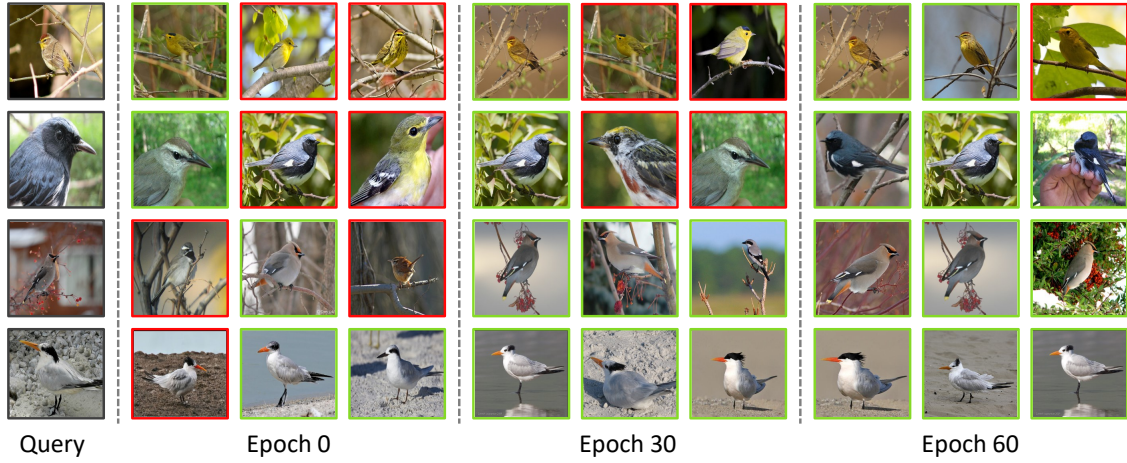


Figure 10. Top-3 retrievals of our model at every 30 epochs on the CUB-200-2011 datasets. Images with green boundary are correct and those with red boundary are incorrect.



Figure 11. Top-3 retrievals of our model at every 30 epochs on the Cars-196 datasets. Images with green boundary are correct and those with red boundary are incorrect.



Figure 12. Top-3 retrievals of our model at every 30 epochs on the SOP datasets. Images with green boundary are correct and those with red boundary are incorrect.