

RESEARCH

Open Access



Improving deep metric learning via self-distillation and online batch diffusion process

Zelong Zeng^{1*} , Fan Yang², Hong Liu^{3*}  and Shin'ichi Satoh¹ 

Abstract

Currently, deep metric learning (DML) methods typically rely on class labels to keep positive samples as closely clustered as possible, while distancing them from negative samples. However, this approach tends to disregard some vital information inherent in the data, such as intra-class information variation, which can hinder the generalization of the trained model. To address this issue, we propose an online batch diffusion-based self-distillation method (OBD-SD), which consists of a progressive self-distillation (PSD) technique and an online batch diffusion process (OBDP). Specifically, PSD is a simple yet effective self-distillation technique that is beneficial for the diversity of embedding representations. OBDP uses the diffusion process to reveal the intrinsic relationships among samples in the mini-batch and produce better soft distance targets. Our approach combines PSD with OBDP, which can extract richer relational information among samples by capturing the local geometric structure of manifolds in each mini-batch. OBD-SD is a highly flexible framework that can be integrated into state-of-the-art DML methods. Our experimental results on CUB200, CARS196, and Stanford Online Products datasets demonstrate that our OBD-SD consistently enhances the performance of existing DML methods, without requiring additional training time, resulting in competitive results.

Keywords: Metric learning, Knowledge Distillation, Diffusion Process, Visual Search

1 Introduction

Deep metric learning (DML) aims to learn an embedding space that uses pre-defined distance metrics (e.g., Euclidean distance) to reasonably measure the similarity between training samples and then apply the measure to unseen test data. The generalization capability of the embedding space is essential for many downstream applications of DML, such as image retrieval [1, 2], face recognition [3, 4], person re-identification [5, 6] and representation learning [7].

Most existing DML methods utilize a pre-defined distance metric, depending on the class label, to increase

the inter-class distance in the embedding space while decreasing the intra-class distance [8]. Typically, these methods prioritize the formation of a compact cluster of intra-class samples with a large distance from other clusters, resulting in a discriminative representation of the known class. However, they often overlook the intrinsic relationships among samples, such as intra-class variations. This occurs because these methods rely solely on the monotonous class-level objective, and the diversity of relationships among samples in the embedding space is overlooked. In fact, this actually hinders the generalization of the learned embedding space, since intrinsic relationships among samples are crucial in identifying unknown classes [9–13]. To increase diversity, we propose a novel, simple yet effective method called online batch diffusion-based self-distillation (OBD-SD), which is composed of

* Correspondence: zengzelong@alumni.u-tokyo.ac.jp; lynnliu.xmu@gmail.com

¹ University of Tokyo and the National Institute of Informatics, Tokyo, Japan

³ Osaka University, Osaka, Japan

Full list of author information is available at the end of the article

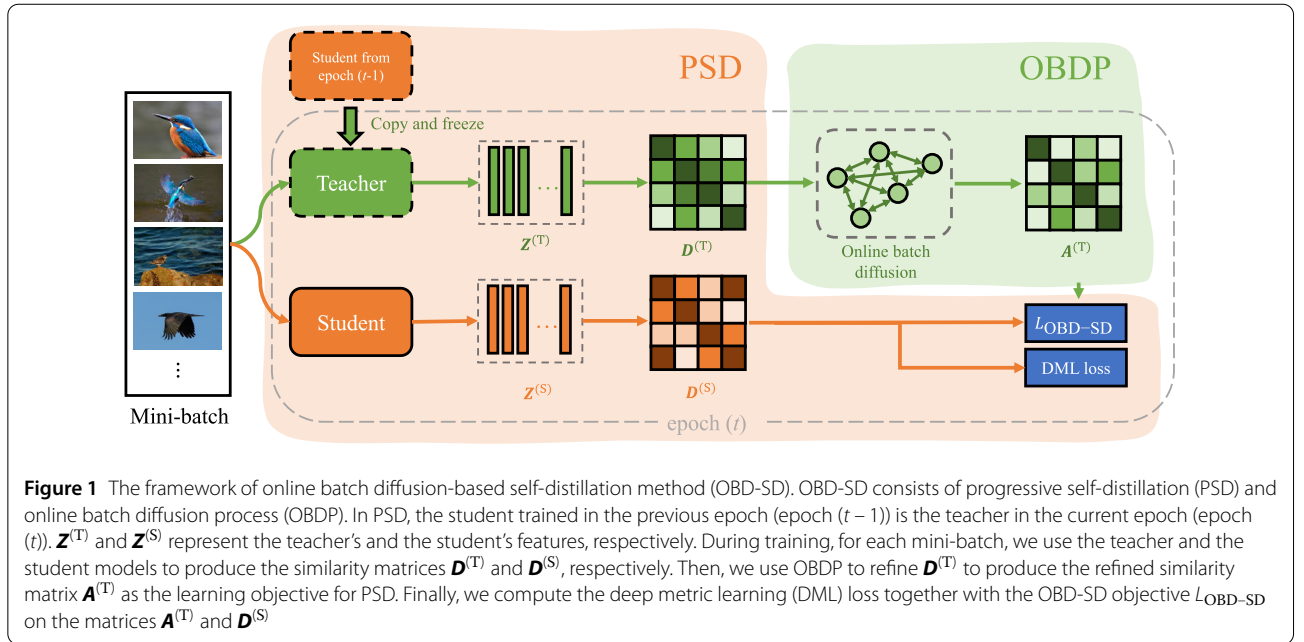


Figure 1 The framework of online batch diffusion-based self-distillation method (OBD-SD). OBD-SD consists of progressive self-distillation (PSD) and online batch diffusion process (OBDP). In PSD, the student trained in the previous epoch (epoch $t - 1$) is the teacher in the current epoch (epoch t). $Z^{(T)}$ and $Z^{(S)}$ represent the teacher's and the student's features, respectively. During training, for each mini-batch, we use the teacher and the student models to produce the similarity matrices $D^{(T)}$ and $D^{(S)}$, respectively. Then, we use OBDP to refine $D^{(T)}$ to produce the refined similarity matrix $A^{(T)}$ as the learning objective for PSD. Finally, we compute the deep metric learning (DML) loss together with the OBD-SD objective $L_{\text{OBD-SD}}$ on the matrices $A^{(T)}$ and $D^{(S)}$.

two components: progressive self-distillation (PSD) and online batch diffusion process (OBDP).

Drawing from prior knowledge distillation in classification task [14–17], we first introduce the PSD module, which can be easily integrated into common DML approaches [18–20]. Unlike the existing knowledge distillation works of DML, which require pre-training teacher models or additional auxiliary network, PSD uses the student model of the previous epoch as the teacher model, which does not require any modifications to the network architecture and has no added computational expenses. As a result, PSD can dynamically distill the distance metric from the model at the previous epoch and then adaptively assign a soft distance target for each training pair of data. Compared to the monotonous class-level objective, soft distance targets can better capture the real relationships between samples. Moreover, PSD takes into account all pairwise relationships among all samples within the mini-batch, and can thus better explore the entire embedding space. As training progresses, the teacher model becomes more robust, gradually improving the performance of the student model. Our theoretical analysis and experimental results demonstrate that soft-distance targets, that are learned from improved diversity in embedding representations, can resolve the aforementioned issue.

It is worth noting that a better distance metric should consider the local geometric structure of the manifolds, which is disregarded by existing self-distillation methods [21, 22], including our proposed PSD. To ensure that the distance targets given by the teacher model can depict a closer-to-reality correlation between samples, we innovatively employ the diffusion process technique [23–25]

to refine the distance targets during training. Therefore, we propose an online batch diffusion process (OBDP), which is designed to learn high-order information from batch manifolds. OBDP can reflect the intrinsic relationships between samples and increase the accuracy of soft-distance target identification. A notable distinction from existing diffusion process methods, which are predominantly used for re-ranking during the inference phase, is our pioneering application of the diffusion process during the training phase. In addition, in re-ranking methods, the output of the diffusion process is ranking scores rather than sample similarities. In OBDP, to tailor the diffusion process to the DML task, we apply diffusion to the affinity matrix, resulting in optimized similarities between samples. Moreover, our OBDP leverages only a mini-batch to construct the manifold structure, aligning with the online learning scheme and significantly boosting computational efficiency. Our experiments and observations show that, compared to existing methods that perform diffusion on global manifolds, our online learning approach achieves competitive or superior results with significantly improved efficiency.

OBD-SD is a plug-and-play technique that can be integrated into most conventional DML methods with negligible extra computational cost. Our theoretical studies (as presented in Sect. 5) reveal that our OBD-SD can take into account the high-order relationships of samples, especially for difficult samples. That is, our OBD-SD can learn a more accurate distance metric, which benefits the overall performance. The entire framework of the proposed OBD-SD is shown in Fig. 1.

We conduct extensive experiments on three widely-used benchmarks, i.e., CUB200-2011 [26], CARS196 [27], and Stanford Online Products [28]. The results validate the merits of the proposed OBD-SD over state-of-the-art (SOTA) methods. Moreover, the experiments in Sect. 4.4 indicate that our OBD-SD indeed increases the diversity and mitigate over-clustering of learned embedding features [22], both of which encourage better generalization to unknown test classes.

2 Related works

2.1 Deep metric learning

Metric learning aims to establish an effective metric space, in which the similarity between samples can be quantified using this distance metric. Recent advances have leveraged deep learning techniques to enhance the quality of the distance metric, which is called deep metric learning (DML). Conventional methods investigate the relationships among the tuples of samples, including pairs [29], triplets [3, 18, 30], and other variants such as N -pairs [31] and lifted structured loss [28]. More recently, AP-based loss [32, 33] and proxy-based method [34–36] have demonstrated their ability to learn good embedding spaces. Moreover, various sampling schemes have been introduced to mitigate the computational complexity stemming from a substantial number of training pairs [3, 18, 37–39].

A recent and promising avenue of research involves investigating how to acquire a discriminative embedding space with good generalization capabilities for unknown classes. Many existing approaches in this category focus on learning an ensemble of embeddings by devising various learning objectives and constraints, highlighting the importance of the diversity of embedding representations in achieving good generalization [9–13, 38]. However, these methods often face issues related to scalability and high computational complexity. In contrast, our approach, OBD-SD, stands out for its efficiency and simplicity, showing good scalability and generalizability.

2.2 Knowledge distillation

Knowledge distillation (KD) attempts to compress models by transferring “dark knowledge” from larger teacher model to a small student model [40–42]. Many recent works have shown that the student model can outperform the teacher model when the student is configured with the same capacity as its teacher [14, 17]. Based on this observation, many self-distillation (Self-KD) schemes have been introduced, such as PS-KD [16], BYOT [43], and CS-KD [15].

More recently, researchers have observed that Self-KD is indeed helpful for improving generalization, with notable examples including RCL [44], BAR [21], and S2SD [22]. Among these methods, BAR adopts a conventional strategy, i.e., employing a pre-trained model as the teacher.

In contrast, S2SD extends the principle of DML by implementing knowledge distillation from multiple auxiliary heads, each with high-dimensional features. Importantly, all auxiliary heads share the same backbone as the basic networks, allowing for simultaneous training.

It is important to emphasize that our approach shares certain similarities with S2SD, but there are distinct differences between the two methods. First, S2SD primarily focuses on the task of “feature compression”, where it utilizes a high-dimensional embedding space with enhanced generalization to aid in training a low-dimensional embedding space. In contrast, our approach, OBD-SD, suggests that an embedding space of the same dimension (rather than a high-dimensional space as in S2SD) can provide valuable information for enhancing performance. This difference has been substantiated by our experiments. Second, OBD-SD incorporates the OBDP module, which refines soft distance targets from the teacher model, a module not present in previous DML methods. Furthermore, we apply OBDP to existing Self-KD methods, including S2SD. The experiments in Sect. 4.3 confirm that OBDP effectively contributes to performance improvements.

2.3 Diffusion process

The diffusion process is a well-established technique commonly used in retrieval tasks [23–25, 45, 46]. Due to the diversity of data, there is usually a distribution difference between training data and unknown test data [47]. This difference often results in data manifolds in the embedding space that are not conducive to accurately measuring relationships between test samples using traditional distance metrics [25]. To address this issue, the diffusion process can capture the local geometric structure of data manifolds based on a neighborhood graph, resulting in the extraction of more accurate distance metrics [24]. The neighborhood graph consists of various nodes and edges, where each node represents a feature from the database and each edge represents the connection between a node and its neighborhoods, weighted proportionally to the pairwise affinities between nodes. On the other hand, the diffusion process is frequently applied in label propagation for deep semi-supervised learning [48–50].

In contrast to the typical use of diffusion process methods as post-processing, OBD-SD integrates the diffusion process as an auxiliary module capable of effectively refining soft distance targets during training. Notably, OBD-SD utilizes only mini-batch data for each diffusion process, a more efficient alternative to the conventional methods of using global data.

3 Proposed method

3.1 Problem formulation

We denote $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ as the training data, and y_i is the corresponding label of $\mathbf{x}_i$. Our goal is to learn an

embedding function $f: \mathcal{X} \mapsto \Phi \subset \mathbb{R}^d$, which maps each input $\mathbf{x} \in \mathcal{X}$ to a d -dimensional embedding space Φ that allows us to measure the similarity between inputs. f can be formulated as a deep model $f(\cdot, \theta)$ with a trainable parameter θ . The output $\mathbf{v}_i = f(\mathbf{x}_i, \theta)$ is typically normalized as $\mathbf{z}_i = \mathbf{v}_i / \|\mathbf{v}_i\|$. The distance (or similarity) between two inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ in the learned metric space is defined as $D_{i,j} := d(\mathbf{z}_i, \mathbf{z}_j)$, where $d(\cdot, \cdot)$ is the pre-defined distance (or similarity) measurement. In this paper, we use cosine similarity as the default.

3.2 Progressive self-distillation

This paper aims to address the issue of neglecting intrinsic relationships. To accomplish this, we first introduce a feasible scheme to assign a more precise distance target for each pair. However, we often encounter situations in which we only have class labels for the data and lack characteristic annotations (e.g., attribute labels). This limitation makes it challenging for current DML methods to obtain precise distance targets solely from sample annotations.

To address this issue, we consequently exploit knowledge distillation strategy [14, 21] to obtain soft distance targets from a teacher model. Previous works usually use a pre-trained model [21, 44] or an auxiliary model [22] as the teacher model. Nonetheless, it is important to note that these methods exhibit inflexibility and necessitate further training.

Yuan et al. [17] observed an intriguing phenomenon that a relatively poorly-trained teacher model could still effectively enhance the performance of student model. Inspired by this observation, we introduce progressive self-distillation (PSD). PSD dynamically leverages the model trained in the previous epoch as the teacher model. Specifically, upon finishing the training of the model in epoch $t - 1$, we duplicate and freeze this model to serve as the teacher model for the subsequent epoch, referred to $f^{(T)}$. As we continue to train the student model $f^{(S)}$, in epoch t , we use $f^{(T)}$ to provide soft distance targets for knowledge distillation.

Similar to some previous works [21, 22], PSD involves matching sample relationships between matrices $\mathbf{D}^{(T)}$ and $\mathbf{D}^{(S)}$. $\mathbf{D}^{(T)} \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{B}|}$ represents the affinity matrix calculated in mini-batch $\mathcal{B}$ using the teacher model, where $\mathbf{D}_{i,j}^{(T)}$ indicates the cosine similarity between the i -th and j -th samples in the mini-batch. $\mathbf{D}^{(S)}$ is the corresponding affinity matrix calculated by using the student model. Thus, we define the PSD objective as follows:

$$L_{\text{PSD}}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \left(\sigma(\mathbf{D}_{i,:}^{(T)} / \tau)_j \ln \frac{\sigma(\mathbf{D}_{i,:}^{(T)} / \tau)_j}{\sigma(\mathbf{D}_{i,:}^{(S)} / \tau)_j} \right), \quad (1)$$

where $\sigma(\cdot)$ is the softmax operation, τ is a temperature parameter, and $\mathbf{D}_{i,:}$ is the i -th row of the matrix $\mathbf{D}$. Therefore, we can directly combine Eq. (1) with a typical DML

method, which can be written as

$$\mathcal{L} = L_{\text{DML}} + \tau^2 \lambda L_{\text{PSD}}, \quad (2)$$

where λ is the weight parameter, L_{DML} is an arbitrary metric learning loss function and τ^2 is a balanced parameter that follows the typical setting in KD loss [40].

However, in the early training phase, the teacher model may not acquire sufficient knowledge from the training data, but may gradually become more reliable as training progresses. Considering this, we utilize a weight updating scheme [16] to dynamically control the weight of the PSD objective. This dynamic weight formulation is defined as $(t/T)\lambda$, where t is the current epoch and T is the total number of training epochs. As a result, we rewrite the objective of PSD-based DML as follows:

$$\mathcal{L} = L_{\text{DML}} + \tau^2 \frac{t}{T} \lambda L_{\text{PSD}}. \quad (3)$$

Our experiments demonstrate that this linear dynamic updating scheme yields superior performance compared to that of a static weight and other dynamic strategies, such as “warm up” and “step-wise”.

3.3 Online batch diffusion process

In Eq. (1), the soft distance targets $\mathbf{D}^{(T)}$ obtained from the teacher model are calculated simply using pairwise distances, which cannot fully reflect the intrinsic relationships among the samples. Many previous works [23–25] have demonstrated that the diversity of data often results in manifolds in the embedding space. Based on this observation, we propose an online batch diffusion process (OBDP) to refine soft distance targets from the teacher model. OBDP captures the local geometric structure of manifolds based on a graph constructed using a mini-batch, providing more information about the intrinsic relationships among samples in the embedding space, leading to improved generalizability. It is worth noting that our OBDP is proposed for the training phase, unlike the existing diffusion process methods used within the inference phase [23–25].

3.3.1 Preliminaries of diffusion process

We first provide a brief overview of the standard diffusion process in image retrieval, primarily following the steps outlined in Refs. [23, 25].

For simplicity, we only elaborate on the case where a certain sample in the gallery is treated as the query, corresponding to the offline phase in Ref. [25]. Given a dataset $\mathcal{U} = \{u_1, \dots, u_n\}$, we sequentially select the i -th sample u_i as the query and use it as the starting position to perform a random walk. Mathematically, we set a one-hot initial state vector $\mathbf{m}^0 = [m_1^0, \dots, m_n^0]^T$, where $m_i^0 = 1$ and the rest are zeros. The random walk is then conducted in line with the following steps.

- 1) Use a pre-trained model $f(\cdot)$ to extract feature vectors $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ from the dataset $\mathcal{U}$, where $\mathbf{z}_i = f(\mathbf{u}_i)$. Then, we calculate the affinity matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, where each component $W_{ij} = d(\mathbf{z}_i, \mathbf{z}_j)$. Note that $W_{ii} = 0$ is guaranteed to circumvent self-loops in the graph represented by $\mathbf{W}$.
- 2) Symmetrically normalize $\mathbf{W}$ by $\mathbf{S} = \mathbf{V}^{-1/2} \mathbf{W} \mathbf{V}^{-1/2} \in \mathbb{R}^{n \times n}$, in which $\mathbf{V} \in \mathbb{R}^{n \times n}$ is the diagonal matrix with $V_{ii} = \sum_{j=1}^n W_{ij}$. We refer to $\mathbf{S}$ as the transition matrix.
- 3) Perform a random walk to update $\mathbf{m}^{t+1} = \omega \mathbf{S} \mathbf{m}^t + (1 - \omega) \mathbf{m}^0$ until convergence, where $\omega \in (0, 1)$ is the transition probability of a random walk.

Finally, the entire iteration process can be written as a closed-form formulation:

$$\mathbf{m}^\infty = (1 - \omega)(\mathbf{I} - \omega \mathbf{S})^{-1} \mathbf{m}^0 \in \mathbb{R}^n, \quad (4)$$

where $\mathbf{I}$ is the diagonal identity matrix, $\mathbf{m}^\infty$ is the converged state of the random walk that respects the ranking score on manifolds between the query and gallery images, and can therefore serve as the criterion for ranking in image retrieval.

3.3.2 Online diffusion on batch manifolds

The standard diffusion process is commonly used in re-ranking task for retrieval and does not align with our requirements. First, in re-ranking tasks, the output of the standard diffusion process are ranking scores, whereas we aim to determine the similarity between samples. Second, Eq. (4) entails a time complexity of $O(N^3)$ (with N being the number of samples), and it needs to be computed only once offline for the re-ranking task. If we aim to use the diffusion process during training, we should update and maintain Eq. (4) on-the-fly. This becomes impractical, especially when dealing with large training sets due to the computational demands involved.

To solve these problems, we propose an OBDP. With OBDP, we apply the diffusion process to the affinity matrix and utilize only the samples within a given mini-batch to create a neighborhood graph for the diffusion process in each iteration. In other words, we are diffusing similarity information specifically within the batch manifolds. Specifically, for each mini-batch $\mathcal{B}$, our approach involves the following steps. First, we use a model to extract the features $\mathbf{Z} \in \mathbb{R}^{|\mathcal{B}| \times d}$ and then calculate the similarity matrix $\mathbf{D} \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{B}|}$. Second, we calculate the transition matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{B}|}$. Then, we set the similarity matrix $\mathbf{D}$ as the initial state for the random walk. The i -th row of $\mathbf{D}$ indicates the similarity vector of the i -th sample in the mini-batch relative to the other samples. Finally, we use a random walk to refine the initial states $\mathbf{D}$:

$$\mathbf{D}^{t+1} = \omega \mathbf{S} \mathbf{D}^t + (1 - \omega) \mathbf{D}^0, \quad (5)$$

where $\mathbf{D}^t$ is the state in the t -th iteration. During each iteration of the random walk, the similarity vectors of each sample are propagated across the manifolds. When two samples exhibit similarity on the manifolds, their performance in similarity vectors also becomes akin. The closed-form solution of Eq. (5) is as follows:

$$\mathbf{A} = (1 - \omega)(\mathbf{I} - \omega \mathbf{S})^{-1} \mathbf{D} \in \mathbb{R}^{|\mathcal{B}| \times |\mathcal{B}|}, \quad (6)$$

where $\mathbf{A}$ is the refined batch-similarity matrix.

The overall time complexity of OBDP is reduced to $O(|\mathcal{B}|^2 N)$ for each epoch. Therefore, OBDP can be efficiently computed online and more efficiently than the standard diffusion process, because $|\mathcal{B}|$ is usually much smaller than the size of the training set N .

3.4 Online batch diffusion-based self-distillation

To integrate OBDP with PSD, we first use the teacher model to extract the features $\mathbf{Z}^{(T)} = f^{(T)}(\mathcal{B})$ and then apply OBDP to calculate the refined similarity matrix $\mathbf{A}^{(T)}$ as follows:

$$\mathbf{A}^{(T)} = (1 - \omega)(\mathbf{I} - \omega \mathbf{S}^{(T)})^{-1} \mathbf{D}^{(T)}. \quad (7)$$

Next, we replace $\mathbf{D}^{(T)}$ in Eq. (1) with $\mathbf{A}^{(T)}$, and the final objective of the proposed OBD-SD is

$$L_{\text{OBD-SD}}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \left(\sigma(\mathbf{A}_{i,:}^{(T)}/\tau)_j \ln \frac{\sigma(\mathbf{A}_{i,:}^{(T)}/\tau)_j}{\sigma(\mathbf{D}_{i,:}^{(S)}/\tau)_j} \right). \quad (8)$$

Similar to Eq. (3), the final objective function for DML can be described as

$$\mathcal{L} = L_{\text{DML}} + \tau^2 \frac{t}{T} \lambda L_{\text{OBD-SD}}, \quad (9)$$

where we can use a traditional stochastic gradient descent (SGD) algorithm to update the parameters.

4 Experiments

4.1 Experimental details

We follow the existing experiment settings established in Refs. [20, 22], which are characterized by a zero-shot scenario. In these settings, the training set does not overlap with the test set.

Datasets. We conduct experiments on three benchmark dataset: CUB200-2011 [26], CARS196 [27] and Stanford Online Products (SOP) [28]. CUB200-2011 contains 11,788 images of 200 classes of birds. We use the first/last 100 classes (5864/5924 images respectively) as the training/test subsets. CARS196 contains 16,185 images of 196 car categories. We use the first/last 98 classes (8054/8131 images) as the training/test subsets. SOP contains 120,053

online product images in 22,634 categories. We use the first 11,318 classes of 59,551 images as the training subset and the remaining 11,316 classes of 60,502 images as the test subset.

Evaluation metrics. We use Recall@ K (R@ K) [51] and normalized mutual information (NMI) [52] as the basic evaluation metrics. The metric R@ K is used to evaluate image retrieval performance, while NMI is employed to assess image clustering performance.

Implementation details. We use ResNet50 (R50/ d) [53] and Inception (IBN/ d) [54] as the backbones, where d is the dimension in the embedding space. The architecture of Inception used here is the Inception BN with frozen batch-normalization layers. If not specifically emphasized, we use R50/128 as the default in the experiments. We use the Adam optimizer to update the parameters of the neural network. We set a fixed learning rate of 1.0×10^{-5} and a weight decay of 4.0×10^{-4} . Please note that, for the sake of a fair comparison, we do not implement any learning rate scheduling unless explicitly stated otherwise. We set the distillation weight/transition probability, i.e., λ/ω , as 1000/0.30 on CUB200-2011, 75/0.99 on CARS196, and 100/0.50 on Stanford Online Products. The temperature score is set to 1 by default. We set the number of epochs to 150 on both CUB200-2011 and CARS196, and 100 on SOP. All the experiments are conducted on a single Nvidia V100 GPU, and the results are computed as averages over multiple runs with different seeds.

4.2 Effectiveness of OBD-SD

First, we integrate our approach with three recent representative methods, namely multisimilarity (MS) loss [19], margin loss with distance-based sampling [18], and regularized margin loss [20]. The experimental results are reported in Table 1, where we observe a consistent enhancement in performance with OBD-SD. For instance, when we set the MS loss as L_{DML} and evaluate the R@1 metric, our proposed OBD-SD shows a remarkable performance improvement of 4.46% on CUB200 and 2.54% on CARS196. Table 1 illustrates that combining MS loss with our OBD-SD consistently achieves the best performance across all three benchmarks. Therefore, we adopt this composition method as our default approach in the following experiments.

Second, we compare our OBD-SD with other state-of-the-art methods [9–11, 13, 36, 38, 39, 55–59]. The results are reported in Table 2. The compared methods are divided into two groups according to their embedding dimensions, (128-dim and 512-dim).

In the first group (R50/128), we observe that the proposed OBD-SD significantly outperforms most SOTA methods such as RankMI [55] and MS+PLG [56]. Compared with that of the MS+PLG, the improvement in the OBD-SD is not significant. We argue that this is because

MS+PLG leverages additional pre-trained natural language processing models, whereas our method can achieve better results without any other additional auxiliary information.

In the second group (IBN/512 and R50/512), the improvement of our proposed OBD-SD is also significant. In contrast to most SOTA methods, OBD-SD is a plug-and-play module that can be integrated into most existing methods. For example, we apply our method to MS+DAS [59], which further improves the performance, resulting in better results on average.

4.3 Comparison with recent self-KD

We first compare OBD-SD with two recent Self-KD-based DML methods, BAR [21] and S2SD [22]. For these compared methods, we reproduce the experiments and carefully select the optimal parameters to achieve the highest performance. The results are shown in Table 3. PSD achieves competitive results on par with BAR using the traditional KD strategy, but BAR requires almost twice the training time. We further compare OBD-SD with S2SD, and the corresponding results in Table 3 show that OBD-SD requires much less extra training time, 17.4% and 7.4% on the two datasets, and achieves competitive performance.

Since OBDP is also a plug-and-play module, we combine it with a recent Self-KD-based DML, i.e., BAR, and S2SD. Specifically, similar to our OBD-SD, we use OBDP to refine the soft labels generated by the teachers (in BAR and S2SD), and then use the refined soft labels for knowledge distillation. The results in Table 3 consistently demonstrate that OBDP contributes to performance improvements across different Self-KD methods. Here, we carefully reproduce the experiments of MS+S2SD+MSD [22] by using the official code provided by the authors.

4.4 Embedding space metrics

We study the impact of OBD-SD on the learned embedding space. We observe the differences between the embedding space learned with and without OBD-SD by using different structural measurements, including embedding space density [20] and spectral decay [20]. The spectral decay quantifies the number of principal directions with variance in the learned embedding space. Note that the lower spectral decay indicates high feature variety and is linked to stronger generalization. In our experiments, we exclude the two largest spectral values for a more robust estimate.

The results in Table 4 show that both PSD and OBDP improve the embedding space density and reduce the spectral decay. This validation confirms that our approach leads to better generalizability. In other words, OBD-SD has the capability to mitigate over-clustering in the embedding space and enhance the diversity of the embedding representation. This means that OBD-SD encourages the model to

Table 1 Comparison with strong metric learning baselines. We conduct experiments on three benchmarks, namely CUB200-2011 [26], CARS196 [27] and SOP [28]. Bold indicates the better result between using the online batch diffusion-based self-distillation method (OBD-SD) and not using OBD-SD with the same metric learning loss function. β is a variable that determines the boundary between positive and negative pairs, which is a key hyperparameter in margin-based loss function as in Ref. [20]

Methods	CUB200-2011		CARS196		SOP	
	R@1	NMI	R@1	NMI	R@1	NMI
Margin ($\beta = 1.2$) [18]	63.15	68.05	80.02	67.50	78.20	90.32
Margin ($\beta = 1.2$) + OBD-SD (ours)	65.78	69.67	81.49	68.18	78.81	90.51
R-Margin ($\beta = 0.6$) [20]	64.54	68.31	82.39	68.57	77.45	90.36
R-Margin ($\beta = 0.6$) + OBD-SD (ours)	66.54	69.24	84.49	70.24	77.75	90.50
MS [19]	63.10	68.33	81.56	69.38	77.75	90.01
MS + OBD-SD (ours)	67.76	70.81	84.10	70.93	78.83	90.22

Table 2 Comparison with state-of-the-art metric learning methods across different architecture (Arch) and dimensionality (Dim). For example, R50/512 means the ResNet50 [53] backbone with a 512-dimensional embedding space. “+D” indicates that one uses combinational pooling in the backbone as performed in ProxyAnchor. Italics indicate the best and second best results in the given Arch/Dim setting, respectively. Bold and italic indicate the best overall results

Methods	Venue	Arch/Dim	CUB200-2011			CARS196			SOP		
			R@1	R@2	NMI	R@1	R@2	NMI	R@1	R@10	NMI
Div&Conq [9]	CVPR’19	R50/128	65.9	76.6	69.6	84.6	90.7	70.3	75.9	88.4	90.2
MIC [38]	ICCV’19	R50/128	66.1	76.8	69.7	82.6	89.1	68.4	77.2	89.4	90.0
PADS [39]	CVPR’20	R50/128	67.3	78.0	69.9	83.5	89.7	68.8	76.5	89.0	89.9
RankMI [55]	CVPR’20	R50/128	66.7	77.2	71.3	83.3	89.8	69.4	74.3	87.9	90.2
MS+PLG [56]	CVPR’22	R50/128	67.8	78.2	70.1	86.0	91.4	72.4	77.9	89.9	90.2
MS+OBD-SD (ours)	-	R50/128	67.8	78.1	70.8	84.9	91.0	72.1	79.6	91.2	90.4
	-	R50/128+D	68.0	78.3	71.1	86.2	91.6	73.1	79.6	91.2	90.3
ProxyAnchor (PA) [36]	CVPR’20	IBN/512+D	68.4	79.2	-	86.1	91.6	-	79.1	90.8	-
ProxyGML [57]	NeurIPS’20	IBN/512	66.7	77.6	69.8	85.5	91.8	72.4	78.0	90.6	90.4
PA+DRML [12]	ICCV’21	IBN/512	68.7	78.6	69.3	86.9	92.1	72.1	71.5	85.2	88.1
PA+MemVir [58]	ICCV’21	IBN/512	69.0	79.2	-	86.7	92.0	-	79.7	91.0	-
EPSHN [11]	WACV’20	R50/512	64.9	75.3	-	82.7	89.3	-	78.3	90.7	-
DiVA [10]	ECCV’20	R50/512	69.2	79.3	71.4	87.6	92.9	72.2	79.6	91.2	90.6
DCML-MDW [13]	CVPR’21	R50/512	68.4	77.9	71.8	85.2	91.8	73.9	79.8	90.8	90.8
MS+DAS [59]	ECCV’22	R50/512+D	69.0	79.2	71.7	87.8	92.8	73.0	80.2	91.5	90.2
HIER [60]	CVPR’23	R50/512+D	70.1	79.4	-	88.2	93.0	-	80.2	91.5	-
MS+HSE [61]	ICCV’23	R50/512+D	67.6	78.0	-	82.0	88.9	-	78.7	90.4	-
MS+OBD-SD (ours)	-	IBN/512	65.8	76.8	68.8	85.4	91.4	70.4	77.6	90.0	89.6
	-	IBN/512+D	65.8	76.9	68.7	85.8	92.1	71.6	77.9	90.1	89.5
	-	R50/512	69.1	79.5	71.1	86.5	92.3	72.0	80.7	92.0	90.4
	-	R50/512+D	69.4	79.7	71.4	88.7	93.6	74.5	81.0	92.1	90.1
MS+DAS+OBD-SD (ours)	-	R50/512+D	71.3	80.7	73.3	88.8	93.7	75.2	80.7	91.9	90.2

Table 3 Comparison with self-distillation based methods. “PSD” is our proposed progressive self-distillation. “OBPD” is our proposed online batch diffusion process

Methods	CUB200-2011			CARS196		
	R@1	NMI	GPU-time	R@1	NMI	GPU-time
MS [19]	63.1	68.3	-	81.6	69.4	-
MS+BAR [21]	64.1	68.4	+105.3%	82.6	69.6	+101.9%
MS+PSD (ours)	63.5	69.1	+5.3%	82.3	70.0	+1.9%
MS+BAR+OBPD (ours)	66.7	70.2	-	84.5	71.4	-
MS+S2SD-MSD [22]	65.2	70.6	+50.4%	84.0	71.3	+35.4%
MS+OBD-SD (ours)	67.8	70.8	+17.4%	84.1	70.9	+7.4%
MS+S2SD-MSD+OBPD (ours)	65.8	70.9	-	84.6	71.9	-

find more intrinsic relation information. These outcomes align with our initial motivation for designing OBD-SD.

In Fig. 2, we visualize the feature distributions on CUB200. Using Fig. 2(a) as a reference, we can observe that there is no significant change in the distribution density of the sample points in Fig. 2(b), indicating that the manifestation of PSD improvement is not prominent. However, a comparison between Fig. 2(a) and Fig. 2(c) reveals that each cluster in Fig. 2(c) maintains pronounced inter-class cluster gaps (inter-class differences), while exhibiting a sparser distribution within clusters (intra-class differences). This observation indicates that the embedding space in Fig. 2(c) has greater intra-class divergence. These observations are consistent with the results analyzed in Table 4.

4.5 Ablation study

The effect of each element on the OBD-SD. We first evaluate the effectiveness of PSD and OBDP. The experimental results are reported in Table 5. We can observe that both PSD and OBDP are beneficial for improving performance, without significantly increasing training overhead. We further conduct experiments to compare the diffusion process within batch manifolds and the diffusion process across global manifolds. During training, for the global diffusion process, we update the whole transition matrix among all training samples once after each training epoch, a method known as offline diffusion. Meanwhile, the diffusion process within batch manifolds remains the default setting, following our OBDP approach. We use MS loss as the default training objective. The results are reported in Table 6. We find that the diffusion process within the batch manifold can achieve almost the same result as that within the global manifold, but the training is more efficient.

The effect of ω . Table 7 shows the results when ω changes from 0.10 to 0.99. We find that the best results are achieved when ω is approximately 0.30 on CUB200-2011 and 0.99 on CARS196. Since the performance is very similar when $\omega \in [0.50, 0.70]$, we set $\omega = 0.50$ on the SOP.

Different sampling strategies. Data sampling is an important technique for DML methods. Therefore, we employ four popular data sampling strategies: 1) random sampling, 2) semi-hard sampling [3], 3) soft-hard mining [38], and 4) distance-based tuple mining [18] with margin loss [18]. We report the results in Table 8. We find that OBD-SD can consistently improve performance through different sampling strategies.

Dynamic weight updating. We investigate the effect of different dynamic weight updating schemes in the proposed PSD. In addition to the linear function, we also consider “warm-up” and “step-wise” as dynamic updating strategies. The “warm-up” strategy first uses the L_{DML} in the first T epochs to train the model, and then adds the PSD objective L_{PSD} with a linear weight until the end of

Table 4 Results of embedding space density $\pi_{\text{ratio}}(\Phi)$ and spectral decay $\rho(\Phi)$. Here, Φ is the learned embedding space. “↑” indicates higher is better, and “↓” indicates lower is better

Methods	CUB200-2011		CARS196	
	$\pi_{\text{ratio}}(\Phi)\uparrow$	$\rho(\Phi)\downarrow$	$\pi_{\text{ratio}}(\Phi)\uparrow$	$\rho(\Phi)\downarrow$
MS	0.4427	0.1544	0.3962	0.1622
MS+PSD	0.4504	0.1444	0.4256	0.1496
MS+OBD-SD	0.6291	0.0292	0.7083	0.0264

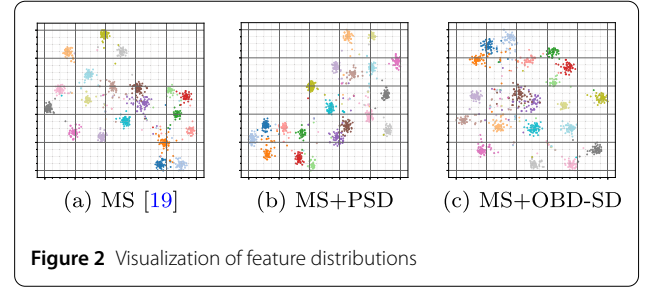


Table 5 Ablation study for each element of OBD-SD

Methods	CUB200-2011		CARS196	
	R@1	NMI	R@1	NMI
MS	63.1	68.3	81.6	69.4
+ PSD w/o dynamic weight	62.9	68.6	81.9	69.4
+ PSD	63.5	69.1	82.3	70.0
+ OBD-SD (PSD+OBDP)	67.8	70.8	84.1	70.9

Table 6 Batch manifold vs. global manifold

Methods	CUB200-2011			CARS196		
	R@1	NMI	GPU-time	R@1	NMI	GPU-time
In global manifold	67.3	70.6	+248.8%	84.4	71.0	+345.0%
In batch manifold (ours)	67.8	70.8	+17.4%	84.1	70.9	+7.4%

training. The “step-wise” strategy also uses L_{DML} in the first T epochs and then adds the PSD objective L_{PSD} with a static weight until the end of the training. Table 9 shows the experimental results. Compared with the linear function, “warm-up” and “step-wise” do not improve performance and introduce an extra hyper-parameter (i.e., T). Therefore, we use a linear function as the default dynamic weight updating scheme.

The effect of λ . Figure 3 displays the results in terms of R@1. The Y-axis describes the performance gain with respect to the baseline and the X-axis indicates the value of λ . On CUB200-2011, the performance is very similar when $\lambda \in [500, 2000]$, so we set $\lambda = 1000$. Similarly, we set $\lambda = 75$ on CARS196, and we set $\lambda = 100$ on SOP.

4.6 Visualization of image retrieval

We present the results of image retrieval on the test sets of the three datasets. Specifically, we display the top 5 gallery

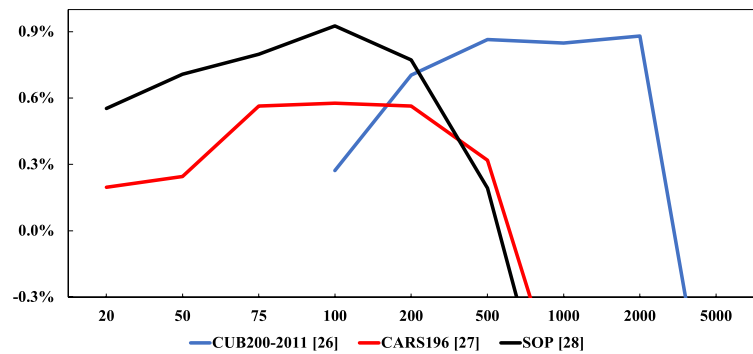


Figure 3 The effect of the distillation weight λ . We conduct experiments on three benchmark dataset, CUB200-2011 [26], CARS196 [27] and SOP [28]. The Y-axis describes the performance gain with respect to the baseline and the X-axis indicates the value of λ .

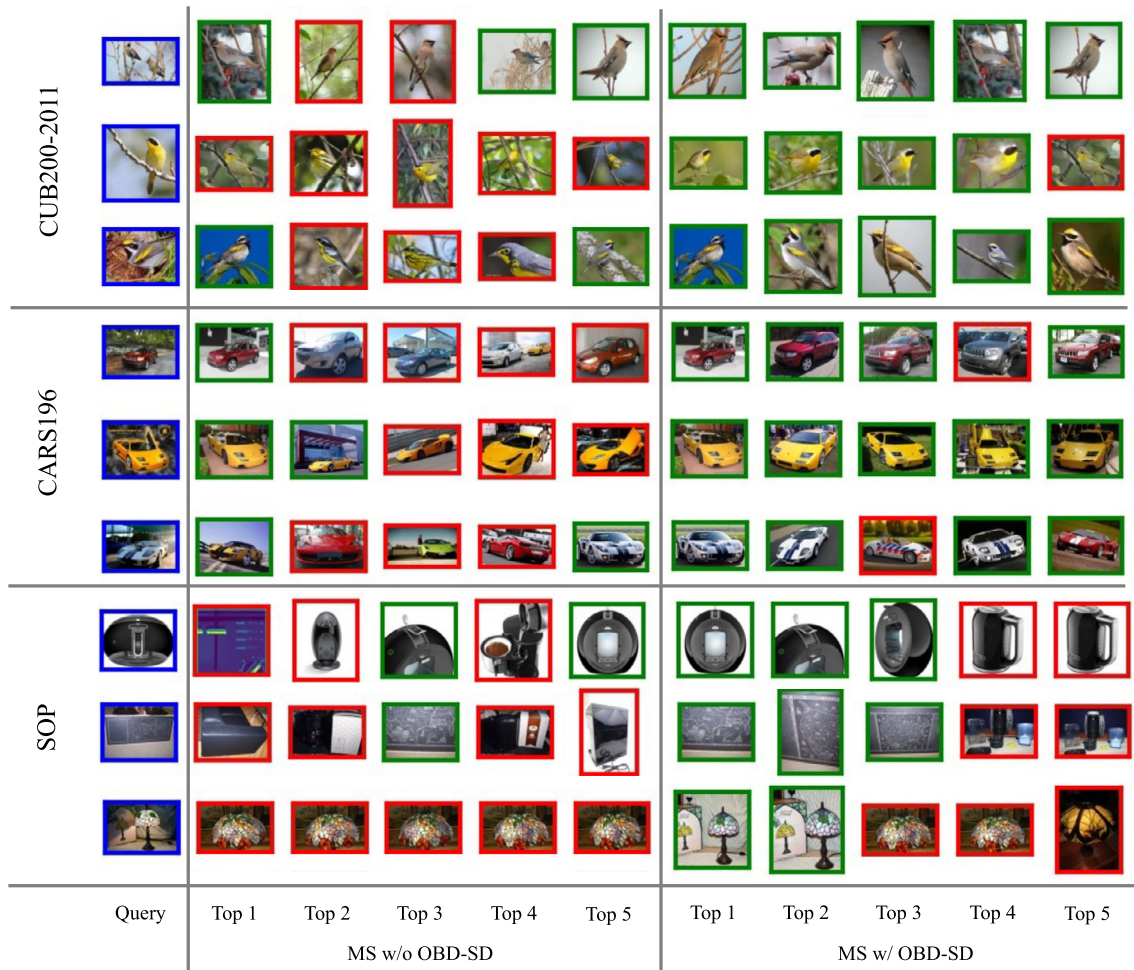


Figure 4 Visualization of the top 5 retrieved images w/o and w/ OBD-SD on three datasets. Green and red borders indicate correct and incorrect search results, respectively

images retrieved for a given query image. The visualization is presented in Fig. 4. Our results reveal that OBD-SD

Table 7 The effect of transition probability ω on the evaluation metric R@1 across three benchmarks

Parameter	CUB200-2011	CARS19	SOP
$\omega = 0.10$	65.19	82.83	78.73
$\omega = 0.30$	67.76	82.82	78.77
$\omega = 0.50$	67.56	82.88	78.88
$\omega = 0.70$	67.19	83.32	78.88
$\omega = 0.90$	66.90	83.66	78.84
$\omega = 0.99$	66.94	84.10	78.84

Table 8 Ablation studies on different sampling strategies. The suffixes “(R)”, “(H)”, “(S)” and “(D)” denote the random, semi-hard, soft-hard and distance-based sampling strategies, respectively

Methods	CUB200-2011		CARS196	
	R@1	NMI	R@1	NMI
Margin (R)	55.5	61.7	66.1	59.5
+ OBD-SD	62.3	67.1	73.3	63.1
Margin (H) [3]	61.2	66.5	76.1	64.4
+ OBD-SD	63.6	67.8	78.3	65.6
Margin (S) [38]	61.3	66.5	77.7	66.1
+ OBD-SD	63.3	68.1	78.6	66.4
Margin (D) [18]	63.2	68.1	80.0	67.5
+ OBD-SD	65.8	69.7	81.5	68.2

Table 9 Ablation studies on different dynamic weight updating functions. “T” is the number of epochs for the warm-up and step-wise strategies

Methods	T	CUB200-2011		CARS196	
		R@1	NMI	R@1	NMI
Warm-up	10	63.5	69.1	81.9	69.3
	20	63.0	68.4	82.1	69.8
	30	62.6	67.8	81.7	69.5
Step-wise	10	63.4	69.1	82.0	68.2
	20	63.0	69.0	81.5	69.5
	30	62.8	68.1	81.8	69.5
Linear	-	63.5	69.1	82.3	70.0

significantly enhances the ability to represent fine-grained information in the embedding space. For example, in the third row of the CUB200-2011, the model trained with OBD-SD accurately distinguishes subtle differences in the patterns of the species from other similar birds. Similarly, in the second row of the CARS196, OBD-SD enables the trained model to differentiate between multiple headlight designs. This is attributed to the fact that OBD-SD leverages richer and more precise information related to the intrinsic relationships of samples during training.

Moreover, we also present some failure cases involving highly challenging samples. We display the top 3 gallery images retrieved for a given query image. The results are shown in Fig. 5. For example, in the third row, the query image displays only a partial view of the sofa, while the ground truth image depicts the entire sofa, even with dif-

ferent colors. In the last row, the task involves retrieving frontal images of the same car model from the gallery using a query image taken from the rear of the car. However, although OBD-SD fails to correctly retrieve some challenging samples, the retrieval results are visually closer to both the ground truth and query than those of the baseline method. For instance, in the third row, OBD-SD is capable of retrieving similar sofas from the gallery, whereas the baseline method fails.

5 Theoretical analysis

5.1 Analysis of PSD

First, we analyze the gradients of the PSD. The gradient of L_{PSD} w.r.t. the embedded feature $\mathbf{v}_i$ is

$$\frac{\partial L_{\text{PSD}}}{\partial \mathbf{v}_i} = \sum_j^{|\mathcal{B}|} ((\mathbf{z}_j - (\mathbf{z}_i \cdot \mathbf{z}_j)\mathbf{z}_i)(P_{ij} - S_{ij})), \quad (10)$$

where P_{ij} and S_{ij} are the similarities between $\mathbf{x}_i$ and $\mathbf{x}_j$ obtained from the student and teacher models, respectively. Then, we can find that Eq. (10) contains two different parts: “attention term” $\mathbf{z}_j - (\mathbf{z}_i \cdot \mathbf{z}_j)\mathbf{z}_i$ and “difference term” $P_{ij} - S_{ij}$.

Attention term. Take positive samples as an example, for an easy positive sample $\mathbf{z}_i \cdot \mathbf{z}_p \approx 1$, we can achieve the following result:

$$\|\mathbf{z}_p - (\mathbf{z}_i \cdot \mathbf{z}_p)\mathbf{z}_i\| = \sqrt{1 - (\mathbf{z}_i \cdot \mathbf{z}_p)^2} \approx 0. \quad (11)$$

Then, for a difficult positive sample, where $0 < \mathbf{z}_i \cdot \mathbf{z}_p \ll 1$, we get the following result:

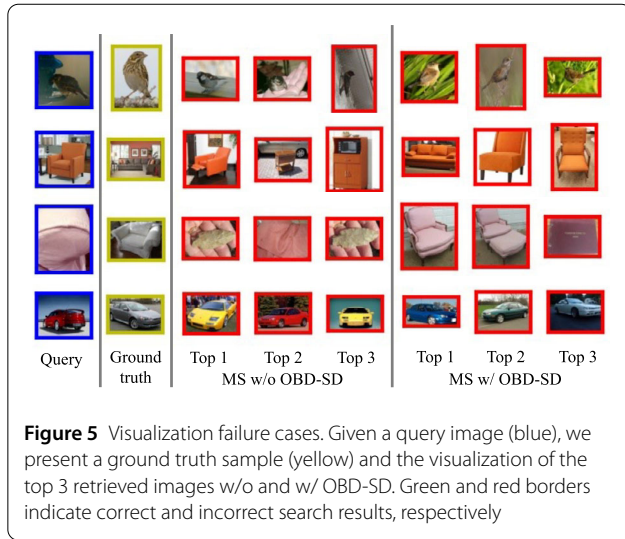
$$\|\mathbf{z}_p - (\mathbf{z}_i \cdot \mathbf{z}_p)\mathbf{z}_i\| = \sqrt{1 - (\mathbf{z}_i \cdot \mathbf{z}_p)^2} \approx 1. \quad (12)$$

Obviously, for easy samples, the gradient of L_{PSD} along $\mathbf{v}_i$ is approximated to be 0. On the other hand, when dealing with difficult samples, this gradient equals $\|\mathbf{z}_p - (\mathbf{z}_i \cdot \mathbf{z}_p)\mathbf{z}_i\| |P_{ip} - S_{ip}| \approx |P_{ip} - S_{ip}|$.

This indicates that easy samples make a relatively small gradient while difficult samples have a significant impact on the gradient. This demonstrates that PSD encourages the model to focus more on difficult samples during training. We believe that difficult samples usually contain more intrinsic information, such as intra-class variations, which conventional methods tend to suppress.

Difference term. The second part of Eq. (10) illustrates that PSD forces the student’s output, P_{ip} , to resemble the teacher’s output, S_{ip} . This means that PSD does not rely solely on class labels to increase inter-class distances while decreasing intra-class distances.

Consequently, the student model can learn more information in the process of distillation, especially for difficult samples, thus alleviating the problem of ignoring the intrinsic relationships of samples.



5.2 Analysis for OBDP

Inspired by Refs. [24, 62], Eq. (6) in our method can be written as the following optimization problem:

$$\min_{\mathbf{A}} \frac{1}{2} \sum_{i,j,k=1}^{|\mathcal{B}|} W_{j,k} \left(\frac{A_{i,j}}{\sqrt{V_{i,i} V_{j,j}}} - \frac{A_{i,k}}{\sqrt{V_{i,i} V_{k,k}}} \right)^2 + \frac{1-\omega}{\omega} \sum_{i,j=1}^{|\mathcal{B}|} (A_{i,j} - D_{i,j})^2, \quad (13)$$

where $\mathbf{A}$ is the refined similarity (the output of OBDP), $\mathbf{D}$ is the initial similarity matrix, and $\mathbf{W}$ is the corresponding affinity matrix.

Therefore, Eq. (13) consists of two terms: the first term emphasizes that OBDP considers high-order relationship between samples; the second term suggests that the initial states should be retained to a certain degree, which emphasizes that a good distance metric should not differ substantially from the initial states.

Specifically, the refined similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is equivalent to aggregating information about the relationship between $\mathbf{x}_i$ and all the other samples in the mini-batch. In other words, if $\mathbf{x}_j$ is located near $\mathbf{x}_k$ in the embedding space, the refined similarities $A_{i,j}$ and $D_{i,k}$ should also exhibit a close similarity score. This indicates that OBDP has the capability to capture the local geometric structure of manifolds and subsequently uncover the intrinsic relationships among samples. In contrast, the initial similarity matrix $D_{i,j}$ only considers pairwise relationships while ignoring the intrinsic relationships between samples.

6 Conclusion

In this paper, we propose online batch diffusion-based self-distillation, named OBD-SD. Our approach consists of two key components: the progressive self-distillation (PSD)

module and the online batch diffusion process (OBDP). PSD aims to encourage the model to acquire more relational information among samples, while OBDP is designed to assist PSD in revealing the intrinsic relationships among samples within batch manifolds. Our extensive experiments, involving various loss functions across three benchmark datasets demonstrate the effectiveness and efficiency of OBD-SD. In future work, we plan to explore the application of the diffusion process to other tasks such as self-supervised learning.

Abbreviations

DML, deep metric learning; IBN, Inception with frozen batch-normalization; MS, multisimilarity; Margin, margin loss with distance-based sampling; OBDP, online batch diffusion process; OBD-SD, online batch diffusion-based self-distillation; PSD, progressive self-distillation; R-Margin, regularized margin loss; Self-KD, self-knowledge distillation; SOTA, state-of-the-art.

Acknowledgements

This work was done when Zelong Zeng was a Ph.D. student at the University of Tokyo.

Author contributions

Conceptualization: Zelong Zeng, Yang Fan, Hong Liu, and Shin'ichi Satoh; Methodology: Zelong Zeng and Yang Fan; Writing (original draft preparation): Zelong Zeng; Writing (review and editing): Zelong Zeng, Yang Fan, Hong Liu, and Shin'ichi Satoh; Validation and visualization: Zelong Zeng; Formal analysis and investigation: Zelong Zeng, Yang Fan, and Hong Liu; Computation resources: National Institutes of Informatics; Supervision: Hong Liu and Shin'ichi Satoh. All authors read and approved the final manuscript.

Funding

This work was supported in part by Grant-in-Aid for JSPS Fellows Grant (No. 21F20377) and was part of the results of Value Exchange Engineering, a joint research project between Mercari, Inc. and the RIIE of the University of Tokyo.

Data availability

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study. All the datasets utilized in this paper are open source.

The code is available at: https://github.com/ZelongZeng/OBD-SD_Pytorch.

Declarations

Competing interests

Hong Liu is an Associate Editor at Visual Intelligence and was not involved in the editorial review of this article or the decision to publish it. The authors declare that they have no other competing interests.

Author details

¹University of Tokyo and the National Institute of Informatics, Tokyo, Japan.

²Shanghai Jiao Tong University, Shanghai, China. ³Osaka University, Osaka, Japan.

Received: 24 October 2023 Revised: 10 June 2024 Accepted: 11 June 2024
Published online: 02 July 2024

References

- Gordo, A., Almazán, J., Revaud, J., & Larlus, D. (2016). Deep image retrieval: learning global representations for image search. In B. Leibe, J. Matas, N. Sebe, et al. (Eds.), *Proceedings of the 14th European conference on computer vision* (pp. 241–257). Cham: Springer.
- Gordo, A., Almazán, J., Revaud, J., & Larlus, D. (2017). End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2), 237–254.
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: a unified embedding for face recognition and clustering. In *Proceedings of the IEEE*

- conference on computer vision and pattern recognition (pp. 815–823). Piscataway: IEEE.
4. Deng, J., Guo, J., Xue, N., & Arcface, S. Z. (2019). Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4690–4699). Piscataway: IEEE.
 5. Zheng, W., Ruimin, H., Chao, L., Yi, Y., Junjun, J., Mang, Y., et al. (2015). Zero-shot person re-identification via cross-view consistency. *IEEE Transactions on Multimedia*, 18(2), 260–272.
 6. Hermans, A., Beyer, L., & Leibe, B. (2017). In defense of the triplet loss for person re-identification. arXiv preprint. [arXiv:1703.07737](https://arxiv.org/abs/1703.07737).
 7. Wang, X., Zhang, H., Huang, W., & Scott, M. R. (2020). Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6388–6397). Piscataway: IEEE.
 8. Suárez, J. L., García, S., & Herrera, F. (2021). A tutorial on distance metric learning: mathematical foundations, algorithms, experimental analysis, prospects and challenges. *Neurocomputing*, 425, 300–322.
 9. Sanakoyeu, A., Tschernetzki, V., Buchler, U., & Ommer, B. (2019). Divide and conquer the embedding space for metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 471–480). Piscataway: IEEE.
 10. Milbich, T., Roth, K., Bharadhwaj, H., Sinha, S., Bengio, Y., Ommer, B., et al. (2020). DiVA: diverse visual feature aggregation for deep metric learning. In A. Vedaldi, H. Bischof, T. Brox, et al. (Eds.), *Proceedings of the 16th European conference on computer vision* (pp. 590–607). Cham: Springer.
 11. Xuan, H., Stylianou, A., & Pless, R. (2020). Improved embeddings with easy positive triplet mining. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 2474–2482). Piscataway: IEEE.
 12. Zheng, W., Zhang, B., Lu, J., & Zhou, J. (2021). Deep relational metric learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 12065–12074). Piscataway: IEEE.
 13. Zheng, W., Wang, C., Lu, J., & Zhou, J. (2021). Deep compositional metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9320–9329). Piscataway: IEEE.
 14. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., & Anandkumar, A. (2018). Born again neural networks. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (pp. 1602–1611). Stroudsburg: International Machine Learning Society.
 15. Yun, S., Park, J., Lee, K., & Shin, J. (2020). Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13876–13885). Piscataway: IEEE.
 16. Kim, K., Ji, B., Yoon, D., & Hwang, S. (2021). Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6567–6576). Piscataway: IEEE.
 17. Yuan, L., Tay, F.E., Li, G., Wang, T., & Feng, J. (2020). Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3903–3911). Piscataway: IEEE.
 18. Wu, C.-Y., Manmatha, R., Smola, A. J., & Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision* (pp. 2840–2848). Piscataway: IEEE.
 19. Wang, X., Han, X., Huang, W., Dong, D., & Scott, M. R. (2019). Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5022–5030). Piscataway: IEEE.
 20. Roth, K., Milbich, T., Sinha, S., Gupta, P., Ommer, B., & Cohen, J. P. (2020). Revisiting training strategies and generalization performance in deep metric learning. In H. Daumé & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (pp. 8242–8252). Stroudsburg: International Machine Learning Society.
 21. Qin, Z., Yan, L., Tay, Y., Zhuang, H., Wang, X., Bendersky, M., et al. (2021). Born again neural rankers. arXiv preprint. [arXiv:2109.15285](https://arxiv.org/abs/2109.15285).
 22. Roth, K., Milbich, T., Ommer, B., Cohen, J. P., & Ghassemi, M. (2021). Simultaneous similarity-based self-distillation for deep metric learning. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (pp. 9095–9106). Stroudsburg: International Machine Learning Society.
 23. Zhou, D., Weston, J., Gretton, A., Bousquet, O., & Schölkopf, B. (2003). Ranking on data manifolds. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Proceedings of the 16th international conference on neural information processing systems* (pp. 169–176). Cambridge: MIT Press.
 24. Bai, S., Bai, X., Tian, Q., & Latecki, L. J. (2017). Regularized diffusion process for visual retrieval. In S. Singh & S. Markovitch (Eds.), *Proceedings of the AAAI conference on artificial intelligence* (pp. 3967–3973). Palo Alto: AAAI Press.
 25. Yang, F., Hinami, R., Matsui, Y., Ly, S., & Satoh, S. (2019). Efficient image retrieval via decoupling diffusion into online and offline processing. In P. van Hentenryck & Z. Zhou (Eds.), *Proceedings of the 33rd AAAI conference on artificial intelligence* (pp. 9087–9094). Palo Alto: AAAI Press.
 26. Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). *The caltech-ucsd birds-200-2011 dataset*. Technical report, California Institute of Technology.
 27. Krause, J., Stark, M., Deng, J., & Li, F.-F. (2013). 3D object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 554–561). Piscataway: IEEE.
 28. Oh Song, H., Xiang, Y., Jegelka, S., & Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4004–4012). Piscataway: IEEE.
 29. Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1735–1742). Piscataway: IEEE.
 30. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., et al. (2014). Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1386–1393). Piscataway: IEEE.
 31. Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama & U. von Luxburg, et al. (Eds.), *Proceedings of the 30th international conference on neural information processing systems* (pp. 1857–1865). Curran Associates: Red Hook.
 32. Brown, A., Xie, W., Kalogeiton, V., & Zisserman, A. (2020). Smooth-ap: smoothing the path towards large-scale image retrieval. In A. Vedaldi, H. Bischof, T. Brox, et al. (Eds.), *Proceedings of the 16th European conference on computer vision* (pp. 677–694). Cham: Springer.
 33. Rolinek, M., Musil, V., Paulus, A., Vlastelica, M., Michaelis, C., & Martius, G. (2020). Optimizing rank-based metrics with blackbox differentiation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7620–7630). Piscataway: IEEE.
 34. Movshovitz-Attias, Y., Toshev, A., Leung, T. K. Ioffe, S., & Singh, S. (2017). No fuss distance metric learning using proxies. In *Proceedings of the IEEE international conference on computer vision* (pp. 360–368). Piscataway: IEEE.
 35. Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., & Jin, R. (2019). Softtriple loss: deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6450–6458). Piscataway: IEEE.
 36. Kim, S., Kim, D., Cho, M., & Kwak, S. (2020). Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3238–3247). Piscataway: IEEE.
 37. Ge, W., Huang, W., Dong, D., & Scott, M. R. (2018). Deep metric learning with hierarchical triplet loss. In *Proceedings of the 15th European conference on computer vision* (pp. 269–285). Cham: Springer.
 38. Roth, K., Brattoli, B., & Ommer, B. (2019). Mic: mining interclass characteristics for improved metric learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8000–8009). Piscataway: IEEE.
 39. Roth, K., Milbich, T., & Pads, B. O. (2020). Policy-adapted sampling for visual similarity learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6568–6577). Piscataway: IEEE.
 40. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint. [arXiv:1503.02531](https://arxiv.org/abs/1503.02531).
 41. Zagoruyko, S., & Komodakis, N. (2017). Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *Proceedings of the 9th international conference on learning representations* (pp. 1–13). Retrieved April 9, 2024, from https://openreview.net/pdf?id=Sks9_ajex.
 42. Tung, F., & Mori, G. (2019). Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1365–1374). Piscataway: IEEE.
 43. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., & Ma, K. (2019). Be your own teacher: improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3713–3722). Piscataway: IEEE.
 44. Kim, S., Kim, D., Cho, M., & Kwak, S. (2021). Embedding transfer with label relaxation for improved metric learning. In *Proceedings of the IEEE/CVF*

- conference on computer vision and pattern recognition (pp. 3967–3976). Piscataway: IEEE.
45. Donoser, M., & Bischof, H. (2013). Diffusion processes for retrieval revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1320–1327). Piscataway: IEEE.
 46. Iscen, A., Tolias, G., Avrithis, Y., Furon, T., & Chum, O. (2017). Efficient diffusion on region manifolds: recovering small objects with compact CNN representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2077–2086). Piscataway: IEEE.
 47. Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: from theory to algorithms*. Cambridge: Cambridge University Press.
 48. Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2003). Learning with local and global consistency. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Proceedings of the 16th international conference on neural information processing systems* (pp. 321–328). Cambridge: MIT Press.
 49. Iscen, A., Tolias, G., Avrithis, Y., & Chum, O. (2019). Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5070–5079). Piscataway: IEEE.
 50. Huang, Q., He, H., Singh, A., Lim, S.-N., & Benson, A. R. (2021). Combining label propagation and simple models out-performs graph neural networks. In *Proceedings of the 9th international conference on learning representations* (pp. 1–21). Retrieved April 9, 2024, from <https://openreview.net/pdf?id=8E1-f3VhX1o>.
 51. Jegou, H., Douze, M., & Schmid, C. (2010). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1), 117–128.
 52. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.
 53. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 770–778). Piscataway: IEEE.
 54. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2818–2826). Piscataway: IEEE.
 55. Kemertas, M., Pishdad, L., Derpanis, K. G., & Rankmi, A. F. (2020). A mutual information maximizing ranking loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14362–14371). Piscataway: IEEE.
 56. Roth, K., Vinyals, O., & Akata, Z. (2022). Integrating language guidance into vision-based deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16177–16189). Piscataway: IEEE.
 57. Zhu, Y., Yang, M., Deng, C., & Liu, W. (2020). Fewer is more: a deep graph metric learning perspective using fewer proxies. In H. Larochelle, M. Ranzato & R. Hadsell, et al. (Eds.), *Proceedings of the 34th international conference on neural information processing systems* (pp. 17792–17803). Curran Associates: Red Hook.
 58. Ko, B., Gu, G., & Kim, H.-G. (2021). Learning with memory-based virtual classes for deep metric learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 11792–11801). Piscataway: IEEE.
 59. Liu, L., Huang, S., Zhuang, Z., Yang, R., Tan, M., & Wang, Y. (2022). DAS: densely-anchored sampling for deep metric learning. In S. Avidan, G. J. Brostow, M. Cissé, et al. (Eds.), *Proceedings of the 17th European conference on computer vision* (pp. 399–417). Cham: Springer.
 60. Kim, S., Jeong, B., & Kwak, S. (2023). HIER: metric learning beyond class labels via hierarchical regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 19903–19912). Piscataway: IEEE.
 61. Yang, B., Sun, H., Li, F. W. B., Chen, Z., Cai, J., & Hse, C. S. (2023). Hybrid species embedding for deep metric learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 11047–11057). Piscataway: IEEE.
 62. Bai, S., Zhou, Z., Wang, J., Bai, X., Latecki, L. J., & Tian, Q. (2017). Ensemble diffusion for retrieval. In *Proceedings of the IEEE international conference on computer vision* (pp. 774–783). Piscataway: IEEE.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)