Intra-class Adaptive Augmentation with Neighbor Correction for Deep Metric Learning

Zheren Fu, Zhendong Mao, Bo Hu, An-An Liu, *Senior Member, IEEE*,
and Yongdong Zhang, *Senior Member, IEEE*

Abstract—Deep metric learning aims to learn an embedding space, where semantically similar samples are close together and dissimilar ones are repelled against. To explore more hard and informative training signals for augmentation and generalization, recent methods focus on generating synthetic samples to boost metric learning losses. However, these methods just use the deterministic and class-independent generations (e.g., simple linear interpolation), which only can cover the limited part of distribution spaces around original samples. They have overlooked the wide characteristic changes of different classes and can not model abundant intra-class variations for generations. Therefore, generated samples not only lack rich semantics within the certain class, but also might be noisy signals to disturb training. In this paper, we propose a novel intra-class adaptive augmentation (IAA) framework for deep metric learning. We reasonably estimate intra-class variations for every class and generate adaptive synthetic samples to support hard samples mining and boost metric learning losses. Further, for most datasets that have a few samples within the class, we propose the neighbor correction to revise the inaccurate estimations, according to our correlation discovery where similar classes generally have similar variation distributions. Extensive experiments on five benchmarks show our method significantly improves and outperforms the state-of-the-art methods on retrieval performances by 3%-6%. Our code is available at **https://github.com/darkpromise98/IAA.**

Index Terms—Deep Metric Learning, Sample Generation, Distribution Estimation, Semantic Augmentation.

I. Introduction

Deep metric learning aims to learn an appropriate distance metric among arbitrary data and project them into an embedding space, where the embeddings of semantically similar samples are close together, while dissimilar ones are far apart. It is a crucial topic in multimedia areas and has been applied to various tasks, including image retrieval [1]–[3], person re-identification [4], [5], detection [6], zero-shot learning [7], [8], face recognition [9], [10], cross-modal matching [11], [12], and hashing [13], [14]. The main paradigm of deep metric learning is designing effective loss functions, which
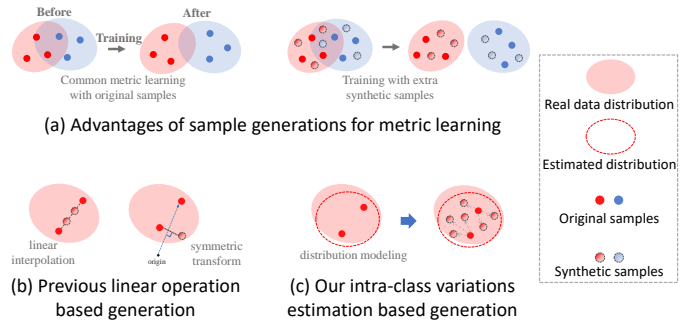
Fig. 1. **Illustration of sample generation, previous linear operation based generation methods, and our intra-class variations estimation based method for deep metric learning.** (a) Sample generation based methods can produce and explore additional hard synthetic samples to learn a more discriminative embedding space [30]. (b) Previous works [27]–[29] use the deterministic linear operation to generate simple synthetic samples, which can not represent the diverse variations for each class. (c) Our work properly considers the wide characteristic changes of different classes, and reasonably models the intra-class variation for generations. Therefore, our generated samples have abundant and rational semantic property.

consider pair-wise relationships between different samples in the embedding space, such as Contrastive loss [15], Triplet loss [16], and other elaborate losses [17], [18]. Metric learning losses get great improvements from various way, including sample mining [19], [20], ensemble models [21], [22], and regularization techniques [23], [24]. Besides, sample generation methods [25], [26] have been proposed to alleviate models focus on a small minority of selected samples, and ignore a large majority of non-selected samples. Various generated samples can provide extra informative training signals to support models learning a more discriminative embedding space, as shown in Fig. 1(a). Existing works [27]–[29] apply the algebraic computation in embedding spaces to generate samples and get powerful results. In general, the essential challenge of generation-based methods is how to synthesize semantically rich samples with high confident labels.

However, previous methods just use the deterministic and class-independent generations, e.g., simple linear interpolation and symmetric transform in Fig. 1(b), which only can cover a tiny and limited part of distribution spaces around original samples. They overlook the wide range of characteristic changes of different classes, and fail to model the diverse intra-class variations, which is significant for effective generations. As a result, generated samples not only lack semantic diversity for the certain class, but also might be noisy signals during training, which will hamper models to learn discriminative
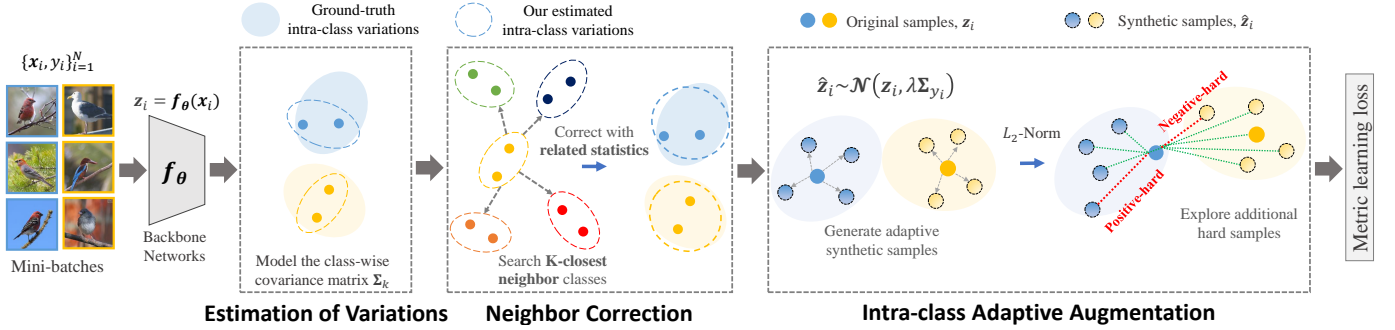
Fig. 2. **Overview of our intra-class adaptive augmentation with neighbor correction framework.** We first tentatively estimate the intra-class variations for each class in the embedding space, and use the neighbor correction to revise the inaccurate estimation for the few-samples class by integrating neighbor statistical information. Then we generate semantically diverse and meaningful synthetic samples and combine them with the sample mining to explore extra hard training signals. Finally, we boost previous metric learning losses to get more discriminative embeddings and achieve better generalization.

embeddings and generalization ability. Besides, most datasets in metric learning and real scenes DO NOT have sufficient number of samples within the class (called few-samples class), thus the accurate estimation of intra-class variations is difficult by common statistical modeling. That's why existing generation-based methods just use deterministic and linear generation, which is not constrained by the number of samples within class, but have various weaknesses simultaneously.

In this paper, we propose a novel intra-class adaptive augmentation (IAA) framework for deep metric learning, which is the first work to model rational intra-class variations for each class and generate semantically diverse and meaningful synthetic samples. Different from previous deterministic generations, we present an efficient class-wise distribution estimation, which can be used to generate a variety of adaptive samples in Fig. 1(c). Further, for the few-samples class, we present the neighbor correction to solve the well-known challenge and get more accurate estimation. Specifically, as shown in Fig. 2, we first estimate the intra-class variations statistically, which can be regarded as the specific distributions for every class in the embedding space. Then we use the neighbor correction to integrates neighbor statistical information and revise the defective estimations for the few-samples class, in the light of our correlation discovery where similar classes generally have similar variation distributions. Finally, we make use of estimated variations to generate adaptive samples and combine them with hard sample mining to boost metric learning losses. Our method can be applied to any existing methods in a plug-and-play manner without any extra network modules, available for all kinds of datasets, and achieve significant performance improvements. The contributions of this paper are four-fold:

- We propose an efficient intra-class variations estimation approach for any form of datasets, which can properly model the class-wise distributions and support to generate adaptive hard samples for deep metric learning.
- We discover the strong correlation between classes and their variations, then present the neighbor correction to revise biased estimations for the few-samples class.
- We propose the intra-class adaptive augmentation to generate semantically diverse and meaningful synthetic samples, then combine them with sample mining strategies to explore extra hard and informative representations.

- Extensive experiments on five typical benchmarks show our method greatly improves and outperforms the state-of-the-art approaches on retrieval tasks by 3%-6%.

II. RELATED WORK

A. Deep Metric Learning

Deep Metric learning aims to learn an explicit non-linear mapping from raw data to a low dimensional embedding space, where similar (positive) samples are pushed together and dissimilar (negative) samples are repelled against, by the metric of the Euclidean distance. Plenty of loss functions have been proposed to get robust embeddings, and can be distinguished into categories, pair-based and proxy-based.

Pair-based losses leverage the pairs of samples to build constraints. Contrastive loss [15] is the most typical method, which minimizes distances between positive pairs and maximizes distances between negative pairs. In parallel, Triplet loss [16] considers three-tuples samples, and requires the distance of a negative pair to be higher than a positive pair by a fixed margin. Then N-pair [31] and Lifted-structured loss [32] are proposed to exploit multiple samples and richer pair-wise relations, have shown to reach great results.

Proxy-based losses [33], [34] use the learnable proxy embeddings as the class representation and the part of network parameters. They leverage data-to-proxy relations [35], [36], which encourage each sample to be close to the proxies of the same class and far away from different classes, instead of other samples. Since they optimize the global structure of the embedding space, have less computations and converge faster if the number of classes is minor, otherwise the parameters are unaccepted and may cause out of memory [35], [37].

Besides, computaions of metric learning is infeasible in practical for large datasets, and the majority of samples are not informative, more works started to focus on the sample minings. Conclusive evidences [19] show that the design of smart sampling strategies is as important as the design of efficient loss functions. Recent improved losses [17], [38] are combined with various samples mining strategies [19], [20], which select hard and informative sample pairs and give them higher loss weighting. But samples minings usually focus on the selected minority and overlook the non-selected majority, which can lead to a overfitting model [27], [28].

B. Sample Generation

Recently, sample generation methods [25], [39], [40] are proposed to improve the loss functions and mining strategies of deep metric learning. They can provide various more informative samples and potential training signals [27], [30], which do not exist in the original data. The essential challenge of generation-based methods is how to synthesize semantically rich samples with high confident labels. Early methods [26], [30] leverage generative adversarial networks or auto-encoders to produce synthetic samples, but lead to extra-large models and unstable training. Recent works [27]–[29] generate virtual samples or classes in embedding spaces by geometric relations. They just use the simple algebraic computation, such as linear interpolation and symmetric transform, but ignore the broad characteristic changes in different classes. Further, we solve these problems by modeling the intra-class variation and generate adaptive samples without any complex networks.

C. Semantic Augmentation

Semantic augmentation is a common technique to regularize deep networks in various visual tasks, and has been widely used to improve feature generalization and reduce training overfitting. Previous works [41] discover inherent variations in feature spaces along with certain directions essentially correspond to implementing meaningful semantic changes in the original data domain. For instance, deep feature interpolation [42] uses linear interpolations of features from a pre-trained networks to edit the semantics of images. ISDA [43] estimates the class-distribution in embedding spaces, and leverages specific feature transform to achieve semantic data augmentation effectively for downstream tasks. Therefore, estimating suitable distributions of deep features and augmenting features from the estimation can improve the generalization in many areas [44], [45]. However, these methods generally get poor performances on the scarce data in minority classes [46]. We make use of the basic principle of semantic augmentation, and further propose a novel distribution correction method for the few-samples class and long-tail datasets.

III. PRELIMINARIES

This section introduces the basic mathematical formulation of deep metric learning. Let $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ denotes a train dataset in the original data space, and $\mathcal{Y} = \{y_i\}_{i=1}^N \in [1, 2, \ldots, C]$ are the corresponding class labels. Deep metric learning aims to learn a mapping $\mathcal{X} \to \mathcal{Z}$, from the original data space $\boldsymbol{x}_i \in \mathcal{X}$ to the D-dimensional embedding space $\boldsymbol{z}_i \in \mathcal{Z}$ by deep neural networks f_θ: $f_\theta(\boldsymbol{x}_i) = \boldsymbol{z}_i$, so that embeddings of similar data are close together and dissimilar ones are far apart. Distance metric between two samples $\boldsymbol{z}_i, \boldsymbol{z}_j$ in the embedding space can be defined as:

$$d_{ij} = d(\boldsymbol{z}_i, \boldsymbol{z}_j) = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2, \tag{1}$$

where $d(,)$ is the Euclidean distance between two embeddings. We also define the similarity metric by computing cosine similarity $s(\boldsymbol{z}_i, \boldsymbol{z}_j) = \boldsymbol{z}_i^T \boldsymbol{z}_j / (\|\boldsymbol{z}_i\|_2 \|\boldsymbol{z}_j\|_2)$. L_2-norm usually

is used to embeddings, so two metrics are equivalent. Various metric learning losses have been proposed in recent years.

Contrastive loss [15] is the basic metric learning loss, which only considers two-tuple sample relations. It pulls positive pairs close together, and pushes negative pairs far away until given thresholds respectively:

$$\mathcal{L}_{Cont} = \frac{1}{n} \sum_{i=1}^n \left(\sum_{y_i = y_j} [d_{ij} - \beta_p]_+ + \sum_{y_i \neq y_k} [\beta_k - d_{ik}]_+ \right), \tag{2}$$

where $[\cdot]_+$ are the hinge function and n is the mini-batch size. Every sample $\boldsymbol{z}_i$ in the mini-batch is regarded as an anchor point, it needs to match all relevant positive points $\boldsymbol{z}_j$ and negative points $\boldsymbol{z}_k$ to compute the loss.

Triplet loss [16] extends the contrastive formalism by considering three-tuple relationships. It pulls the anchor point $\boldsymbol{z}_i$ closer to the positive point $\boldsymbol{z}_j$ than to the negative point $\boldsymbol{z}_k$ $(y_i = y_j \neq y_k)$ by a fixed margin m:

$$\mathcal{L}_{Trip} = \frac{1}{n} \sum_{i=1}^n \sum_{(j,k) \in \mathcal{T}_i} [d_{ij} - d_{ik} + \alpha]_+, \tag{3}$$

where triplet sets $\mathcal{T}_i$ are constructed from the current mini-batches by various sample mining [19], [32], which usually select hard positive samples with relatively larger distances, and hard negative samples with smaller distances.

MS loss [17] is one of the latest work for metric learning. it measures both self-similarity and relative similarities of sample pairs, which can explore informative samples in a mini-batch and weight them by multiple similarities (MS):

$$\mathcal{L}_{MS} = \frac{1}{n} \sum_{i=1}^n \left\{ \frac{1}{\alpha} \log \left[1 + \sum_{j \in \mathcal{P}_i} e^{\alpha(\lambda - s_{ij})} \right] + \frac{1}{\beta} \log \left[1 + \sum_{k \in \mathcal{N}_i} e^{\beta(s_{ik} - \lambda)} \right] \right\}, \tag{4}$$

where α, β, λ are the loss hyper-parameters and s_{ij} is cosine similarity. $\mathcal{P}_i$ and $\mathcal{N}_i$ are selected hard positive and negative samples set for the anchor $\boldsymbol{z}_i$ by its hard sample mining.

$$\mathcal{N}_i = \{s_{ij} | s_{ij} > \min_{y_k = y_i} s_{ik} - \epsilon\} \tag{5}$$

$$\mathcal{P}_i = \{s_{ij} | s_{ij} < \max_{y_k \neq y_i} s_{ik} + \epsilon\} \tag{6}$$

IV. PROPOSED METHOD

In this section, we first introduce the estimation of intra-class variations and related challenge for the few-samples class. Then, we describe our correlation discovery between classes and their variations, which provides the foundation for the neighbor correction. Finally, we integrate our intra-class adaptive augmentation (IAA) into existing methods.

(a) Correlation on metric learning datasets

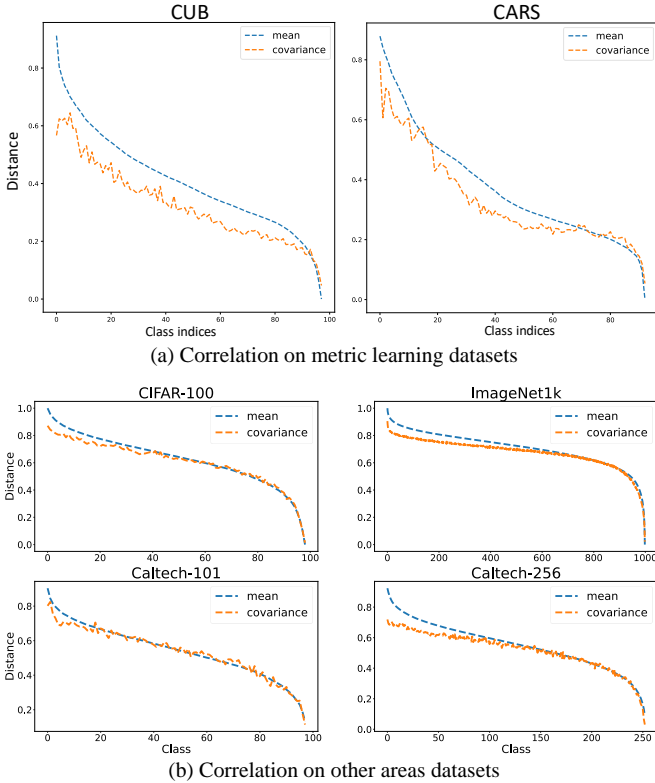(b) Correlation on other areas datasets

Fig. 3. **Illustration of the strong correlation between mean distances and covariance distances.** We use the GoogLeNet [48] pre-trained on ImageNet to extract their embeddings. (a) Correction on two typical metric learning datasets, CUB [49] and CARS [50]. (b) Correlation on four other areas datasets, CIFAR-100 [51], ImageNet-1k [52], Caltech101 [53], and Caltech256 [53]. Class indices are re-listed according to the descending orders of mean distances (blue curve), and distances are normalized to $[0, 1]$. We get the final curve by averaging the results on all classes. For simplicity, we use the diagonal elements of covariance matrix to compute distances (p=2).

A. Estimation of Intra-class Variations

To generate adaptive synthetic samples for each class, we need to estimate the intra-class variations properly. We assume all samples of each class belong to the specific Gaussian distribution in the embedding space. Therefore, we use the Maximum Likelihood principle [47] to approximately estimate the mean and covariance matrix of the class-wise distributions. If the class k ($k \in [1, ..., C]$) have the sample indices set $\mathbb{S}_k$, which contains all samples within the class, and the number of samples is n_k, we have:

$$\boldsymbol{\mu}_k = \frac{1}{n_k} \sum_{i \in \mathbb{S}_k} \boldsymbol{z}_i, \qquad (7)$$

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k} \sum_{i \in \mathbb{S}_k} \left(\boldsymbol{z}_i - \boldsymbol{\mu}_k\right)\left(\boldsymbol{z}_i - \boldsymbol{\mu}_k\right)^T, \qquad (8)$$

where the covariance matrix $\boldsymbol{\Sigma}_k$ reflects the characteristic variations within the class k, and can be easily computed by aggregating all embeddings from the whole dataset.

However, the reasonable estimation of covariance matrices is not always easy. This common statistical method requires a sufficient amount of samples in every class for the accurate estimation [43], Unfortunately, only small datasets (e.g., CUB

TABLE I

Datasets	Spearman correlation coefficient			
	$p=1$	$p=2$	$p=3$	$p=4$
CUB [49]	0.81	0.77	0.74	0.69
CARS [50]	0.91	0.85	0.78	0.74
CIFAR-100 [51]	0.94	0.90	0.85	0.80
ImageNet-1k [52]	0.82	0.71	0.64	0.62
Caltech101 [53]	0.93	0.88	0.80	0.72
Caltech256 [53]	0.94	0.91	0.85	0.78

and CARS in metric learning) could have adequate number of samples in each class. Most of the large datasets (e.g., SOP, InShop, and VehicleID in metric learning) in real scenes only have scarce samples within the class, so the appropriate estimation becomes an ill-posed problem [46].

That's why latest synthetic sample generation methods [27]–[29] just use simple linear operations to generate samples, it's not constrained by the number of samples within class, and have various limitations simultaneously. To solve this problem, we first expound on our discovery of the strong statistical (positive) correlation from estimated distributions.

B. Correlation Discovery

Theoretically, the mean and covariance matrix of a certain Guassian distribution are independent parameters, since they are the two sufficient statistics [54]. However, we have discovered their unique statistical correlation on our estimated class-wise distributions: *If the distances of different means are close, the distances of corresponding covariance matrices will be close, too.* Formally, we use the p-norm of vector or matrix difference to define the distance metrics of mean and covariance, between the class i and the class j:

$$D_m(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \left\| \boldsymbol{\mu}_i^2 - \boldsymbol{\mu}_j^2 \right\|_p, \qquad (9)$$

$$D_{cv}(\boldsymbol{\Sigma}_i, \boldsymbol{\Sigma}_j) = \left\| \boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_j \right\|_p, \qquad (10)$$

Given a certain class, we compute the mean distance and covariance distance with other classes. Then we regard two types of distances as two variable sequences, and re-list the class indices according to the descending order of mean distances. Fig. 3 shows the related results on two typical metric learning datasets (CUB and CARS), and four datasets widely used in various visual tasks. Besides, we compute the Spearman correlation coefficient [55], which quantitatively evaluates the ranking correlation of two variables in Tab. I.

It is not hard to find that the mean distances and corresponding covariance matrices distances have the strong correlation [45]. That is to say, the similar means usually have similar covariance matrices in our estimated class-wise distributions. Further, the similar classes usually have similar intra-class variations. Therefore, we can use samples of similar classes to improve the inaccurate estimation.

C. Neighbor Correction

For the few-samples class dataset, we use the k-nearest neighbor principle to assemble adequate statistical information from neighbor classes and correct the defective estimation, based on our correlation discovery in Section IV-B.

Without loss of generality, we use the diagonal covariance matrix and the 2-norm as the distance metric for simple computation. Given the class k, we select the top-K most similar classes by computing the mean distance as Eq. (9).

$$\mathbb{N}_k = \left\{ i \mid \left\| \boldsymbol{\mu}_i^2 - \boldsymbol{\mu}_k^2 \right\|_2 \in \min(\mathrm{K}), \ i \in [1, ..., C] \right\}, \quad (11)$$

where $\min(\mathrm{K})$ aims to select the minimal K elements, so $\mathbb{N}_k$ represents the indices set of K closest neighbor classes for the class k. Then we compute the weighted sum by integrating all statistical information of neighbor classes:

$$\boldsymbol{\Sigma}_{k, \text{ neighbor}} = (\sum_{i \in \mathbb{N}_k} w_i \boldsymbol{\Sigma}_i)/(\sum_{i \in \mathbb{N}_k} w_i), \quad (12)$$

w_i is the weight for each $\boldsymbol{\Sigma}_i$, and it is determined by three parts, the number of samples n_k, the relevant mean distance $D_m(\boldsymbol{\mu}_i, \boldsymbol{\mu}_k)$, and covariance distance $D_{cv}(\boldsymbol{\Sigma}_i, \boldsymbol{\Sigma}_k)$:

$$w_i = n_i \cdot \exp\left(-\frac{\left\| \boldsymbol{\mu}_i^2 - \boldsymbol{\mu}_k^2 \right\|_2^2}{2\sigma_m^2} - \frac{\left\| \boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_j \right\|_2^2}{2\sigma_{cv}^2}\right), \quad (13)$$

The number of samples n_i can represent the proportion of neighbor statistics, and two relative distances of mean and covariance are constructed by the Gaussian kernel [56], where more similar neighbor classes will get larger weight coefficients. σ_m and σ_{cv} control the spread of related distances. If $\sigma_m = \sigma_{cv} = \infty$, Eq. (13) will degrade to the simple average without considering the distances of means and covariances. Besides, we compute the global covariance matrix from all classes and use it to avoid overfitting.

$$\boldsymbol{\Sigma}_{\text{global}} = (\sum_{k=1}^{C} n_k \boldsymbol{\Sigma}_k)/(\sum_{k=1}^{C} n_k), \quad (14)$$

Finally, we introduce the neighbor correction to the initial covariance matrix for revising the defective estimation.

$$\boldsymbol{\Sigma}_k^{cr} = (1 - \alpha)\boldsymbol{\Sigma}_k + \alpha[(1 - \gamma)\boldsymbol{\Sigma}_{k, \text{ neighbor}} + \gamma \boldsymbol{\Sigma}_{\text{global}}], \quad (15)$$

where α is a dynamic coefficient relied on the number of samples n_k, and computed by the neighbor weight function:

$$\alpha = \begin{cases} (1 + \log[1 + \beta \cdot (n_k - 1)])^{-1}, & n_k \leq \tau, \\ 0, & n_k > \tau. \end{cases} \quad (16)$$

The larger number of samples within the class is, the smaller value α is, the lower effect of the neighbor correction is applied to the initial covariance matrix. Therefore If n_k is large, α will close to 0, the neighbor correlation is unnecessary. γ is a fixed coefficient to balance global and neighbor information.

D. Intra-class Adaptive Augmentation

With proper estimation and correction of intra-class variations as Eq. (8) and Eq. (15), we can generate adaptive synthetic samples. Since the class-wise covariance matrix proportionally represents the intra-class variations, and contains rich semantic diversity [43], [57]. For each original sample $\boldsymbol{z}_i$, we generate synthetic samples with semantically meaningful changes, through translating $\boldsymbol{z}_i$ along different directions sampled from $\mathcal{N}(0, \lambda \Sigma_{y_i})$ in the embedding space. Equivalently, we get:

$$\hat{\boldsymbol{z}}_i \sim \mathcal{N}(\boldsymbol{z}_i, \lambda \boldsymbol{\Sigma}_{y_i}), \quad (17)$$

where λ is a positive coefficient to control the strength of intra-class variations. We record the multiple synthetic samples generated from original $\boldsymbol{z}_i$ as $\hat{\boldsymbol{z}}_i^m, m = 1, ..., M$. Then we combine with hard sample mining to explore more informative samples. We still make original samples as the anchor, but regard all synthetic samples as additional positive and negative candidates for each anchor. We formulate the boosting versions of metric learning losses in Section III.

Contrastive + IAA makes use of all sample-pairs in mini-batches. The original positive samples set for anchor $\boldsymbol{z}_i$ is $\mathcal{P}_i$, and extra synthetic positive samples set is $\hat{\mathcal{P}}_i$. Similarly, $\mathcal{N}_i/\hat{\mathcal{N}}_i$ is the original/synthetic negative samples set.

$$\mathcal{L}_{Cont}^{IAA} = \frac{1}{n} \sum_{i=1}^{n} (\sum_{j \in \mathcal{P}_i \cup \hat{\mathcal{P}}_i} [d_{ij} - \beta_p]_+ + \sum_{k \in \mathcal{N}_i \cup \hat{\mathcal{N}}_i} [\beta_k - d_{ik}]_+), \quad (18)$$

Triplet + IAA is combined with the sample mining, we add synthetic samples as extra candidates to select the hardest negative and all available positive samples. So synthetic samples can provide more three-tuple and informative training signals. For each anchor $\boldsymbol{z}_i$, we mark $\hat{\mathcal{T}}_i$ as the triplet set from synthetic positive samples $\hat{\boldsymbol{z}}_p^m$ and negative samples $\hat{\boldsymbol{z}}_n^m$.

$$\mathcal{L}_{Trip}^{IAA} = \frac{1}{n} \sum_{i=1}^{n} (\sum_{(j,k) \in \mathcal{T}_i \cup \hat{\mathcal{T}}_i} [(d_{ij}) - \min_k (d_{ik}) + \alpha]_+), \quad (19)$$

MS + IAA also extends more informative sample-pairs by synthetic samples. For each anchor $\boldsymbol{z}_i$, we mark extra synthetic set of positive samples as $\hat{\mathcal{P}}_i$ and negative as $\hat{\mathcal{N}}_i$ according to Eq. (5) and Eq. (6), then add them to construct losses.

$$\begin{aligned}
\mathcal{L}_{MS}^{IAA} = \frac{1}{n} \sum_{i=1}^{n} \Bigg\{ &\frac{1}{\alpha} \log \left[1 + \sum_{j \in \mathcal{P}_i \cup \hat{\mathcal{P}}_i} e^{\alpha(\lambda - s_{ij})} \right] \\
&+ \frac{1}{\beta} \log \left[1 + \sum_{k \in \mathcal{N}_i \cup \hat{\mathcal{N}}_i} e^{\beta(s_{ik} - \lambda)} \right] \Bigg\},
\end{aligned} \quad (20)$$

Finnaly, the whole training process of our proposed method is outlined in Algorithm 1.

V. ANALYSIS

A. Comparison to Existing Works

Our work belongs to the generation-based metric learning methods, which produce the synthetic samples in the embedding spaces. However, both our core motivation and technical

Algorithm 1 Train a model with our framework
***
Input: images $\mathcal{X}$, class labels $\mathcal{Y}$, network model f_θ
   while Not Converged **do**
      Estimate the covariance matrix Σ_k as Eq. (8).
      if Belong the few-samples class **then**
         Apply the neighbor correction as Eq. (15).
      end if
      for $t = 0$ to T **do**
         Construct a mini-batch $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ from $\mathcal{X}, \mathcal{Y}$
         Compute embeddings $\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i)$
         Generate adaptive samples $\hat{\boldsymbol{z}}_i$ as Eq. (17)
         Boost metric learning losses as Eq. (18) or Eq. (19)
         Optimize model parameters θ.
      end for
   end while
***

implement fundamentally differ from previous methods [25], [27]–[30], [39]. Existing methods just use the deterministic and class-independent generations, (e.g., simple linear interpolation and symmetric transform) which only can cover a tiny and limited part of distribution spaces around original samples. In other words, they overlook the wide range of characteristic changes of different classes, and fail to model the diverse intra-class variations, which is significant for effective generations. In contrast, our method not only can generate samples with rich semantic diversity for the certain class, but also alleviate the noisy signals during training. Therefore, our models can learn more discriminative embeddings and generalization ability, compared with previous methods.

Besides, our method is close to semantic augmentation, which performs semantic changes for enhancing generalization and reduce overfitting. Previous works [42]–[45] implement specific feature transform or estimate feature distributions in embedding spaces to achieve semantic augmentation. But these methods do not consider the scarce data in minority classes, and can not get reasonable results on the few-samples class. In comparison, we propose a novel distribution correction method for the few-samples class and long-tail datasets to solve this problem.

B. Ground of correlation

Intuitively, our correlation discovery is based on this ground: the deep feature learning of visual tasks aims to ensure that the distance of image features can reflect the similarity of image semantics, i.e., image features are the numerical representation of image semantics. So the changes of image semantics can be reflected on the variance of image features. In real scene, these semantical changes in similar classes are similar, e.g., the posture changes between different cats are similar, compared to other animals. Therefore, features of similar classes have similar variance distribution.

Besides, we find on the few-shot learning [45], [58] and long-tail classification [59], [60] fields, the similar phenomenon is observed, that is the similar classes usually have similar mean and variance of features. These works can provide sufficient guidances and supports for our correlation discovery, which is rigorous enough for metric learning.

Datasets	Objects	Samples	Classes	Number of samples in class			
				max	min	avg	std
CUB	birds	5,864	100	60	41	**58.6**	3.5
CARS	cars	8,054	98	97	59	**82.2**	7.2
SOP	furnitures	59,551	11,318	12	2	**5.3**	3.0
InShop	clothes	25,882	3,997	162	1	**6.5**	6.4
VehicleID	vehicles	110,178	13,134	142	2	**8.4**	9.5

More profound analysis and experiments is on Section VI-D. We provide more correlation results on different datasets (Section VI-D1 and VI-D2), and show how the correlation affects the embedding learning and final performances (Section VI-E6 and VI-F4). These experiments can prove the authority and indispensability of correction discovery.

C. Model Robustness

According to Section IV-C, α controls the weight of neighbor correction, and is an important parameter for our method. Fortunately, α need not to experiment many times to get the best values, since it is a dynamic coefficient relied on the number of samples n_k, and computed by our designed neighbor weight function as Eq. (16): The larger number of samples within the class is, the smaller value α is, the lower effect of the neighbor correction is applied to the initial covariance matrix. When n_k in is large, α will close to 0, the neighbor correlation becomes invalid. Eq. (16) only depends on an insensitive hyper-parameter. (see Section VI-F1 for more explanations and analysis)

Our method produces the adpative virtual samples for different metric learning losses, Since it provides more informative samples and extra supervised signals during training, it can combined with the hard sample minings to boost the performances. (see Section VI-F2 for more experimental analysis)

D. Apply to Proxy-based Loss

In section IV-D, we have applied our method on the pair-based metric learning loss [16], [17], [19], which take pairs of samples to constitute pairwise distances. We only can apply our method on the proxy-based loss [33], [34], [36]. Since proxy-based loss does not need large batches and hard mining for boosting, unlike our used pair-based loss relies heavily on them, so the improvements are not so well (More experimental results and analysis is on Section VI-E4).

VI. EXPERIMENTS

A. Datasets

We evaluate our method on five typical deep metric learning benchmark datasets for image retrieval task. The training set and testing set of all datasets do not overlap in classes.

(1) CUB-200-2011 (CUB) [49] contains 200 classes of birds with 11,788 images. We split the first 100 classes with 5,864

TABLE III
RETRIEVAL PERFORMANCES OF OUR METHOD ON FIVE DATASETS AND THREE BASELINE LOSSES. WE LIST THE PREVIOUS SAMPLE GENERATION METHODS, AND ALL OF THEM (INCLUDING OURS) USE THE GOOGLENET BACKBONE WITH 512 EMBEDDING SIZES FOR A FAIR COMPARISON. *ReImp* INDICATES OUR RE-IMPLEMENTATION WITH THE STANDARD SETTINGS [37]. WE USE THE MEDIUM SIZE OF TEST SET FOR PKU VEHICLEID.

Method	CUB-200-2011			Cars-196			Stanford Online Products			In-shop Clothes Retrieval			PKU VehicleID	
	R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100	R@1	R@10	R@20	R@1	R@5
Triplet+DAML [30]	37.6	49.3	61.3	60.6	72.5	82.5	58.1	75.0	88.0	-	-	-	-	-
Triplet+HDML [26]	43.6	55.8	67.7	61.0	72.6	80.7	58.5	75.5	88.3	-	-	-	-	-
Triplet+DVML [39]	43.7	56.0	67.8	64.3	73.7	79.2	66.5	82.3	91.8	-	-	-	-	-
Triplet+Symm [27]	55.0	67.3	77.5	69.7	78.7	86.1	68.5	82.4	91.3	-	-	-	-	-
Triplet+EE [28]	51.7	63.5	74.5	71.6	80.7	87.5	77.2	89.6	95.5	-	-	-	-	-
Triplet (*ReImp*)	54.4	65.8	76.0	69.0	78.1	85.4	73.5	87.8	94.9	86.1	95.7	97.0	83.5	93.3
Triplet + IAA	**58.9**	**70.6**	**80.5**	**72.4**	**81.5**	**88.2**	**76.9**	**89.9**	**96.0**	**88.6**	**97.4**	**98.3**	**87.7**	**94.7**
Contrastive (*ReImp*)	57.3	68.6	79.5	74.1	82.8	89.0	75.0	88.3	95.2	86.5	96.0	97.3	87.9	94.9
Contrastive + IAA	**61.7**	**73.3**	**82.2**	**80.8**	**87.9**	**92.8**	**77.8**	**90.2**	**96.1**	**88.8**	**97.3**	**98.2**	**91.6**	**95.9**
MS+EE [27]	57.4	68.7	79.5	76.1	84.2	89.8	78.1	90.3	95.8	-	-	-	-	-
MS+PS [29]	55.9	68.1	78.1	71.8	80.9	87.6	-	-	-	-	-	-	-	-
MS+L2A [61]	60.6	72.5	82.2	81.2	88.4	93.0	-	-	-	-	-	-	-	-
MS (*ReImp*)	56.2	68.3	79.1	77.0	84.3	89.9	75.9	89.3	95.6	87.0	97.0	98.1	89.5	94.9
MS + IAA	**62.1**	**73.8**	**82.7**	**84.2**	**90.4**	**94.3**	**78.6**	**91.0**	**96.5**	**89.3**	**97.8**	**98.6**	**92.2**	**96.2**

TABLE IV
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON FOUR DATASETS. BACKBONES ARE DENOTED BY ABBREVIATIONS: BN–INCEPTIONBN, R–RESNET50. THE SUPERSCRIPTS REPRESENT EMBEDDING SIZES. THE BEST AND SECOND BEST RESULTS ARE MARKED IN BOLD AND UNDERLINE.

Method	Setting	CUB-200-2011			Cars-196			Stanford Online Products			In-shop Clothes Retrieval		
		R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100	R@1	R@10	R@20
Circle [18]	BN^{512}	66.7	77.4	86.2	83.4	89.8	94.1	78.3	90.5	96.1	-	-	-
XBM [37]	BN^{512}	65.8	75.9	84.0	82.0	88.7	93.1	79.5	90.8	96.1	89.9	97.6	98.4
PA [35]	BN^{512}	68.4	79.2	**86.8**	86.1	91.7	95.0	79.1	90.8	96.2	91.5	98.1	98.8
CGML [24]	BN^{512}	66.3	77.2	86.3	85.6	91.1	94.2	79.1	90.8	96.1	90.5	98.2	98.6
PS [29]	BN^{512}	66.0	76.6	85.0	84.7	90.7	94.6	79.8	90.9	96.4	90.9	97.9	98.5
FastAP [62]	R^{512}	-	-	-	-	-	-	76.4	89.0	95.1	90.9	97.7	98.5
RaMBO [22]	R^{512}	63.5	74.8	84.1	-	-	-	78.6	90.5	96.0	88.1	97.0	97.9
DiVA [21]	R^{512}	**69.2**	79.3	-	87.7	92.9	-	79.6	91.2	-	-	-	-
SR [23]	R^{512}	68.2	78.1	86.5	87.7	92.5	95.4	78.6	90.6	96.2	-	-	-
MS + IAA	BN^{512}	66.5	76.8	84.4	85.3	91.3	94.8	80.5	92.0	96.8	92.0	98.5	99.1
MS + IAA	R^{512}	68.5	78.4	86.4	**88.6**	**93.1**	**95.8**	**82.8**	**93.2**	**97.1**	**93.3**	**98.6**	**99.2**

TABLE V
RETRIEVAL PERFORMANCE OF OUR METHODS COMPARED WITH THE STATE-OF-THE-ART METHODS ON VEHICLEID. THE TEST SETS ARE SPLITED INTO THREE SIZES (SMALL, MEDIUM, AND LARGE).

Method	Setting	Small		Medium		Large	
		R@1	R@5	R@1	R@5	R@1	R@5
BIER [63]	G^{512}	82.6	90.6	79.3	88.3	76.0	86.4
ABIER [64]	G^{512}	86.3	92.7	83.3	88.7	81.9	88.7
MS [17]	BN^{512}	91.0	96.1	89.4	94.8	86.7	93.8
XBM [37]	BN^{512}	94.6	96.9	93.4	96.0	93.0	96.1
MIC [65]	R^{128}	86.9	93.4	-	-	82.0	91.0
Divide [66]	R^{128}	87.7	92.9	85.7	90.4	82.9	90.2
FastAP [62]	R^{512}	91.9	96.8	90.6	95.9	87.5	95.1
MS + IAA	BN^{512}	95.4	97.8	93.8	96.7	92.7	96.6
MS + IAA	R^{512}	**95.7**	**97.9**	**94.3**	**97.0**	**93.6**	**96.9**

images for training, and remain the rest of the 100 classes with 5,924 images for testing. The images are very similar, some breeds can only be distinguished by minor details.

(2) Cars-196 (CARS) [50] contains 196 classes of car models with 16,185 images. We use the standard split with the first 98 classes for training (8,054 images) and the rest of the 98 classes for testing (8,131 images).

(3) Stanford Online Products (SOP) [32] contains 22,634 classes with 120,053 product images downloaded from eBay. We use the standard split as 11,318 classes for training (59,551 images) and other 11,316 classes for testing (60,502 images).

(4) In-shop Clothes Retrieval (InShop) [67] contains 7,982 classes of clothing items with 52,712 images. According to official guidelines, 3,997 classes are for training (25,882 images) and 3,985 classes are for testing (28,760 images). The test set are partitioned to query set and gallery set, where query set contains 14,218 images of 3,985 classes and gallery set contains 12,612 images of 3,985 classes.

(5) PKU VehicleID (VehicleID) [68] contains 221,736 images of 26,267 vehicles categories captured by surveillance cameras. we use 110,178 images of 13,134 classes for training and 111,585 images of the other classes for testing. We evaluate on the predefined small, medium and large test sets which contain 800, 1,600 and 2,400 classes respectively.

The first two small datasets (CUB, CARS) have *plenty of* samples in each class. The last three large datasets (SOP, InShop, and VehicleID) have *a few* samples. Hence we apply our neighbor correction on the last three datasets. More statistical information is shown in Tab. II.

TABLE VI

PERFORMANCES ON INFORMATIVE METRICS [72]. ALL MODELS USE THE GOOGLENET AS THE BACKBONE AND 512 AS EMBEDDING SIZES.

Method	CUB		CARS		SOP		InShop	
	MAP	RP	MAP	RP	MAP	RP	MAP	RP
Triplet	20.6	31.4	19.4	30.2	45.8	49.0	60.2	63.2
Triplet+IAA	**21.5**	**32.5**	**22.6**	**33.4**	**48.8**	**52.0**	**63.1**	**66.0**
Contrastive	20.8	31.9	20.3	30.5	47.7	50.8	59.9	62.8
Contra+IAA	**22.8**	**33.7**	**24.6**	**34.9**	**49.7**	**52.8**	**61.9**	**64.7**
MS	20.5	31.6	22.0	32.7	48.0	51.2	60.8	63.8
MS+IAA	**22.6**	**33.7**	**27.6**	**37.7**	**50.5**	**53.7**	**63.3**	**66.3**

TABLE VII

COMPARISON WITH THE STATE-OF-THE-ART DEEP METRIC LEARNING METHODS. BACKBONES ARE DENOTED BY ABBREVIATIONS: BN–INCEPTIONBN, R–RESNET50. THE SUPERSCRIPTS REPRESENT EMBEDDING SIZES.

Method	Setting	CUB		CARS		SOP	
		MAP	RP	MAP	RP	MAP	RP
Triplet [16]	BN^{512}	23.7	34.6	23.0	33.7	43.4	46.5
PNCA [36]	BN^{512}	24.2	35.1	25.4	35.6	47.2	50.1
Margin [19]	BN^{512}	23.1	34.0	24.2	34.8	41.8	44.9
FastAP [62]	BN^{512}	23.5	34.2	23.1	33.6	43.6	46.6
MS [17]	BN^{512}	24.7	35.4	28.1	38.1	45.8	48.8
SNR [73]	BN^{512}	25.8	36.6	25.0	35.2	44.5	47.4
SR [23]	R^{128}	-	35.1	-	37.5	-	44.5
DIML [74]	R^{128}	23.3	34.0	23.3	33.6	42.2	45.6
MS + IAA	BN^{512}	25.5	36.5	29.2	39.3	54.6	57.6
MS + IAA	R^{512}	**26.1**	**37.1**	**32.4**	**42.4**	**57.9**	**60.7**

B. Implementation Details

1) Embedding networks: We implement our method on the single GPU of NVIDIA RTX 2080Ti or 3090. As previous methods, we use three common backbone networks: GoogLeNet [48], Inception with batch normalization [69], and ResNet50 [70] pretrained on ImageNet [52]. We replace its last layer with a randomly initialized fully-connected layer for metric learning. The output embeddings are L_2 normalized for computing distance, and the embedding size is 512.

2) Optimization: The input images are first resized to 256 $\times$ 256, then cropped to 224 $\times$ 224. We use random crop and random horizontal flips as common data augment for training, and only use the single center crop for test. We use Adam [71] with 4×10^{-4} weight decay as the optimizer. The initial learning rate is 10^{-4} and scaled up 10 times on the output layers for faster convergence. We set the training epoch as 100 and scale the learning rate by 0.2 times during the 40-th and 60-th epoch. The mini-batches are constructed with the balanced sampler. The batch size is 128 or 256.

3) Hyper-parameters: We set $M = 3$, $\beta = \gamma = 0.1$, $K = 25$, $\sigma_m = \sigma_{cv} = 1$, $\tau = 40$ as the constant, $\lambda \in [0.6, 0.8]$ based on datasets. The estimation of covariance matrix is updated every 4 epochs, and uses diagonal elements for simplicity.

4) Evaluation protocol: We follow the same standard evaluation protocol as previous works [35], [72], so we evaluate the performances by three retrieval metric. **(1) Recall@K (R@K)**: The fraction of queries that have at least one positive sample in their K-nearest neighbors. **(2) R-precision (RP)**: For each query, let R be the number of positive samples, RP is defined as the fraction of the number of positive samples in the R-nearest neighbors. **(3) MAP@R (MAP)**: Similar to mean
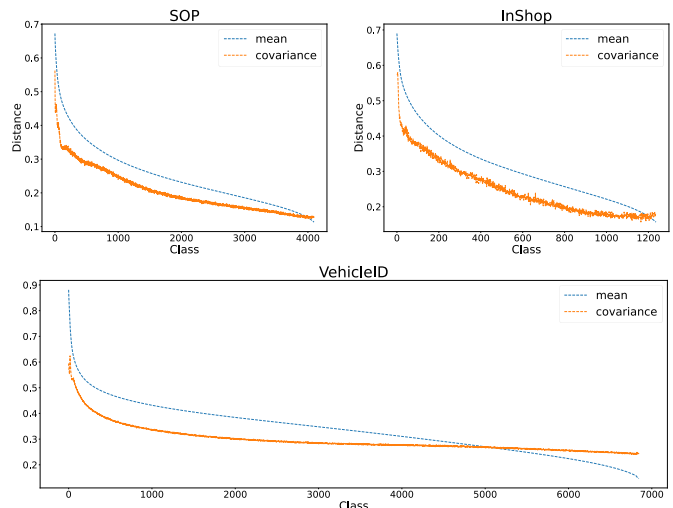

Fig. 4. **Correlation in other metric learning datasets, SOP, InShop, and VehicleID**. We select the classes whose amount is greater than 5 to compute mean and covariance (last 4233, 1313, and 7022 classes, respectively).
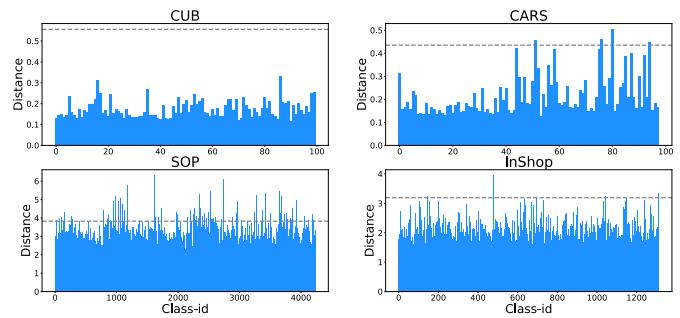

Fig. 5. **Distances between the global covariance matrix and every class covariance.** The gray dashed is the value of distance from the global covariance matrix to origin. For stability, we select classes whose amount is greater than 5 to compute covariance in SOP and InShop.

average precision, but limit the number of nearest neighbors to R. The formal definition is on Supplementary Material.

C. Quantitative Results

We compare our method with existing sample generation based metric learning methods and all typical state-of-the-art metric learning methods, respectively.

1) Comparison with generation-based methods: We implement baselines and our IAA with GoogLeNet backbone and 512 embedding sizes by following all previous methods of synthetic sample generation. We also list previous methods with the same backbone and embedding sizes for a fair comparison. As presented in Tab. III and VI, our method can bring significant performance improvements, such as 3%-6% R@1 gains, 1%-3% MAP and RP gains on three baselines and five datasets. Compared with existing generation methods (e.g., DAML, HDML, DVML, Symm, and EE), our method gets obvious performance advantages. It shows that our method can help to learn discriminative embeddings by leveraging various hard synthetic samples for deep metric learning.

TABLE VIII
SPEARMAN CORRELATION RESULTS BETWEEN MEAN AND COVARIANCE DISTANCES BY DIFFERENT DISTANCE METRICS.. WE USE THE GOOGLENET BACKBONE TRAINED ON EACH DATASET TO EXTRACT EMBEDDINGS, AND COMPUTE DISTANCES BY DIAGONAL ELEMENTS OF COVARIANCE MATRICE.

Distance metric		p	CUB	CARS	SOP	InShop	VehicleID
mean	covariance						
$\left\|\boldsymbol{\mu}_i^2 - \boldsymbol{\mu}_j^2\right\|_p$	$\left\|\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_j\right\|_p$	1	0.71	0.70	0.65	**0.71**	**0.59**
		2	**0.72**	**0.72**	**0.68**	0.69	0.55
		3	0.69	0.71	0.56	0.66	0.53
$\left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|_p$	$\left\|\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_j\right\|_p$	1	0.66	0.64	0.60	0.64	0.52
		2	0.67	0.62	0.63	0.66	0.54
		3	0.65	0.60	0.61	0.65	0.51
$\left\|\boldsymbol{\mu}_i^2 - \boldsymbol{\mu}_j^2\right\|_p$	$\left\|\boldsymbol{\Sigma}_i^{\frac{1}{2}} - \boldsymbol{\Sigma}_j^{\frac{1}{2}}\right\|_p$	1	0.66	0.59	0.53	0.56	0.50
		2	0.68	0.64	0.55	0.58	0.48
		3	0.67	0.65	0.54	0.59	0.45
$\left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right\|_p$	$\left\|\boldsymbol{\Sigma}_i^{\frac{1}{2}} - \boldsymbol{\Sigma}_j^{\frac{1}{2}}\right\|_p$	1	0.64	0.58	0.49	0.51	0.49
		2	0.65	0.57	0.53	0.55	0.51
		3	0.64	0.56	0.50	0.52	0.43

2) Comparison with state-of-the-art methods: We compare our approach with the state-of-the-art deep metric learning methods. We write related backbone networks and embedding sizes (generally, larger is better), which affect final performances greatly. As shown in Tab. IV, V, and VII, our method undoubtedly outperforms the state-of-the-art methods on all benchmarks by obvious margins. We achieve higher performances than loss-designed methods (e.g., PA and Circle) and boost-like methods (e.g., DiVA and RaMBO). Besides, compared with works published within one year (e.g., CGML, SR, and PS), our method still gets preferable performances. All results demonstrate our method have strong effectiveness and generalization for metric learning.

D. Correlation Analysis

We focus on the authenticity of our correlation discovery, and conduct deep experiments to verify its existence.

1) Correlation on more datasets: We first conduct the correlation experiments on SOP, InShop, and VehicleID datasets, where the number of class samples is well not distributed. As shown in Fig. 4, since SOP and Inshop are few-samples and large datasets, the variance estimation is very inaccurate and there are many classes, so the correlation is not as good as in CUB and CARS, but the correlation does exist. All similar results show our correlation discovery are indeed consistent despite fields and datasets.

2) Distance metric: We show the correlation results by other distance metrics. First, we define the distance of different mean by using the *matrix p-norm*, with or without square operation. Similarly, we define the distance of different covariance matrices by using the same the *matrix p-norm*, with or without sqrt operation. As shown in Tab. VIII, different combinations of distance metrics lead to different degrees of correlation. We find The first metric is best for measuring the correlation between classes and their variations. So we select this distance metric to construct the neighbor correction.

3) Variety of covariance: To reveal the diversity of estimated covariance matrix, we compute the distance between the global covariance matrix and covariance matrice of each class in Fig. 5. We can find most of classes have quite different from global covariance, it proves the covariance matrices are diverse as intra-class variations, and not lead to a trivial solution.
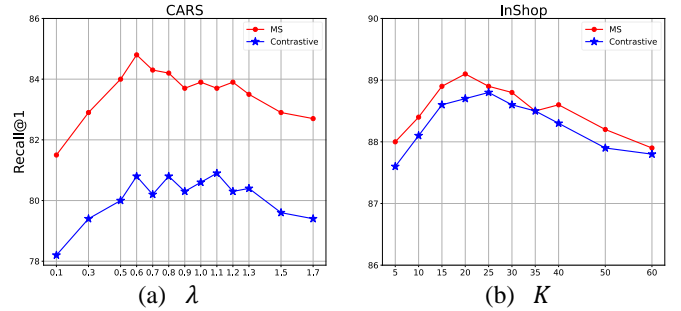

Fig. 6. **Effect of two important hyper-parameters.** (a) Intra-class variations strength λ. (b) The number of neighbor classes K.
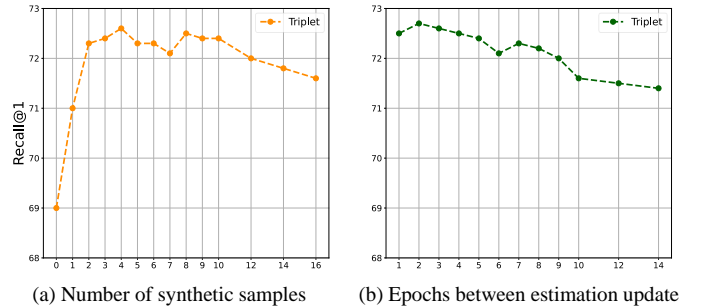

Fig. 7. **Effect of (a) the number of synthetic samples M, (b) the update frequency for estimation** in CARS dataset.

E. Ablation Study

We provide the ablation study to verify the effectiveness of our method and evaluate the contributions of different parts. We use the GoogLeNet backbone with 512 embedding sizes and report Recall@1 performance as default.

1) Hyper-parameters: We show the effect of four important hyper-parameters as shown in Fig. 6. When one is variable, the other is fixed as default for controlled experiments. We find the performances are stable when λ is changed in a proper range, but can't be too small and large, since it might generate non-informative and invalid synthetic samples. When the number of neighbor classes K reduces, the helpful neighbor statistical information will decrease for estimation correction, then related performances drop heavily. It proves the importance of our neighbor correction, but selecting too many classes also will introduce noisy signals to hinder correction.

2) Number of synthetic samples: We explore the importance of the number of synthetic samples in Fig. 7(a). Generating $2-4$ samples from each original sample can get superior results, and the performances do not keep improving by the number of synthetic samples increasing. It shows our method can provide enough hard and informative samples by the small amount of generations, and excessive synthetic samples will distract models from original hard samples.

3) Update frequency of estimation: Updating the estimated covariance frequently can guarantee the timeliness of distributions during training, but extra time cost is unacceptable by collecting all embeddings from the datasets. So we perform on the update frequency in Fig. 7(b). We find performances have

TABLE IX
EFFECT OF VARIOUS DEGENERATIONS OF THE ESTIMATED COVARIANCE MATRIX Σ_k. *Identity matrix* MEANS REPLACING Σ_k BY THE IDENTITY MATRIX. *Global matrix* MEANS USING THE GLOBAL COVARIANCE MATRIX Σ_{GLOBAL} TO REPLACE Σ_k, AND *Diagonal matrix* MEANS USING THE DIAGONAL ELEMENTS OF Σ_k. *Correction* MEANS APPLYING THE NEIGHBOR CORRECTION FOR ESTIMATION AND ONLY USING THE DIAGONAL MATRIX FOR COMPUTING SIMPLICITY.

Setting	w/o Correction		w/ Correction	
	CUB	CARS	SOP	InShop
Baseline (Triplet)	54.4	69.0	73.9	86.3
Identify matrix	50.8	66.1	71.7	84.5
Global matrix	54.9	69.8	73.1	85.8
Diagonal matrix	58.7	72.3	-	-
$\alpha = 0$ (w/o correction)	-	-	74.2	86.9
$\alpha = 1$ (w/o Σ_k)	-	-	76.6	88.2
$\gamma = 1$ (w/o Σ_{global})	-	-	76.5	88.4
$\sigma_m = \sigma_{cv} = \infty$	-	-	76.4	88.3
IAA	**58.9**	**72.4**	**76.9**	**88.6**

TABLE X
RECALL@1 PERFORMANCES ON OTHER METRIC LEARNING LOSS FUNCTIONS. ALL OF THEM USE THE GOOGLENET AS BACKBONE AND 512 AS EMBEDDING SIZE. '✓' INDICATES OUR METHOD ARE APPLIED.

More metric learning losses		CUB		CARS	
			✓		✓
Pair-based	Triplet+Distance [19]	57.7	**59.8**	78.2	**80.8**
	Margin [19]	57.5	**59.3**	75.4	**77.6**
	Circle [18]	60.5	**60.9**	80.8	**81.0**
Proxy-based	Norm-softmax [33]	60.1	**61.3**	80.4	**81.8**
	Proxy-NCA [36]	60.5	**61.0**	81.0	**82.1**
	SoftTriple [34]	61.0	**61.5**	82.6	**82.8**

Fig. 8. **Recall@1 comparison on various embedding dimensions.**

Fig. 9. **The effect of our neighbor weight function** (a) The α values according to different β and the number of samples. (b) The performances of the parameter β of the neighbor weight function.

no significant drop within a suitable range of epoch interval, since the drifting of embeddings is slow in late training [37].

4) Loss Function: Our method can be widely applied to all metric learning losses. As shown in Tab. X, our method also brings performance boost on all benchmarks for other baseline losses, but the promotion is not so well compared to the results in Tab. III. For pair-based losses, we think because the distance-weighted sampling [19] will omit most of informative synthetic samples. For proxy-based losses, we think because they need not large batch sizes and hard mining, unlike pair-based loss relies heavily on them.

5) Embedding dimensions: In metric learning and various retrieval tasks, the trade-off between speed and accuracy is an important issue, where the embedding dimension is the key factor. We test our method with different embedding dimensions, as shown in Fig. 8. Our method significantly improves the performance of baselines in all embedding dimensions. It indicates that our method constructs a highly efficient embedding space for all the dimensionality.

6) Covariance matrix: To better verify the effectiveness of estimated covariance matrices, we evaluate them with different degeneration in Tab. IX. When we use the identity matrix or single global matrix to replace class-wise Σ_k, the results are terrible, it proves the significance of intra-class variance estimation. Only using the diagonal matrix can get get the best performance, but the full matrix is the best. Without our neighbor correction, the defective estimations lead to
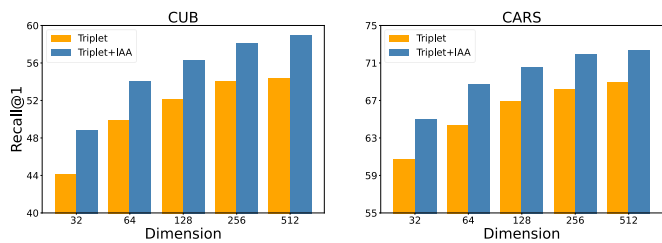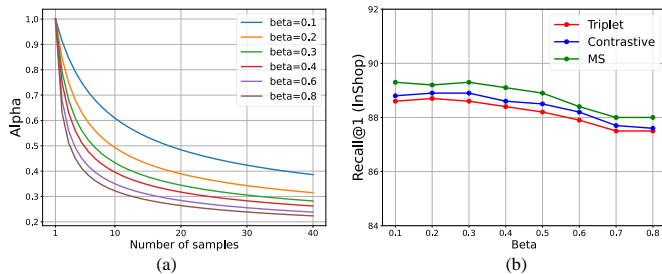
trivial results. And the global information is also helpful in preventing overfitting. Finally, the relative distance weight for neighbor classes improves the performance slightly.

F. Robustness Analysis

To investigate the availability and robustness of our method, we provide more detailed experiments and reasonable analysis. The experimental settings are the same as Section VI-E.

1) Weight of correction: Although α controls the weight of neighbor correction, it is a dynamic coefficient relied on the number of samples n_k, and computed by our designed neighbor weight function as Eq. (16). As shown in Fig. 9, β is the parameter of the neighbor weight function, the smaller it is, the larger the weight is, under the same number of samples. It proves we need enough neighbor information to correct the defective estimation for optimal results.

2) Hard mining for synthetic samples: We investigate the hard mining ability by synthetic samples in Fig. 10. We find our method can provides more hard samples for different metric learning losses, and the selection radio of synthetic samples is always high during hard mining process. Therefore, our generation method can continuously contribute various informative synthetic samples and supervisory signals.

3) Time and memory cost: Our method also need additional operations to generate and make use of synthetic samples, but these computational and memory consumption are almost negligible, since our method has no complex network structure, as shown in Fig. 11. Our method just needs *2%* more run-time cost and *1%* more GPU-memory cost.

4) Availability of neighbor correction: To further verify the availability of neighbor correction, we apply the neighbor correction on the CUB and CARS They have adequate number of samples in each class, we just randomly select N samples from every class to estimate the covariance matrices, then use
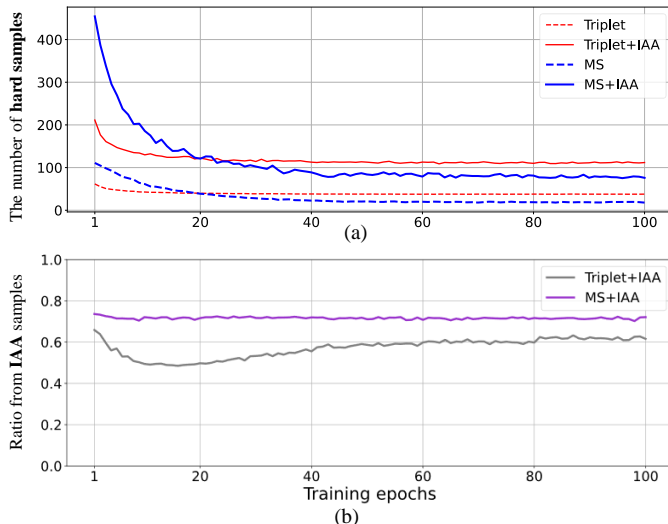
Fig. 10. **The effect of hard mining from IAA** (a) The number of hard negative samples selected by metric learning losses during training. (b) Ratio of synthetic sample selected by related hard samples mining during training. The number of generated samples from each original sample is $M = 3$.
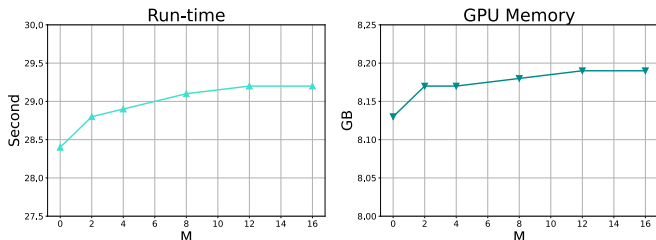

Fig. 11. **Run-time and GPU-memory cost for sample generation and utilization.** We train models with Triplet loss and GoogLeNet backbone for 100 iterations. M is the number of generated samples.

or not the neighbor correction. Tab. XI shows the defective estimation of intra-class variations leads to performance drops for the few-samples class. Our neighbor correction certainly can alleviate the problem and bring performance promotions by revising the estimation with neighbor statistical information. However, the approximate method is weaker than the direct estimation from sufficient samples.

5) Generation strategy: When we generate synthetic samples from the original sample z_i, we sampling from the distribution $\mathcal{N}(z_i, \lambda\Sigma_{y_i})$ (dynamic mean) to get diversely semantic intra-class variations and generate synthetic samples. Besides, we also can sampling from the standard distribution $\mathcal{N}(\mu_{y_i}, \lambda\Sigma_{y_i})$ (fixed mean). Tab. XII shows directly sampling from the distribution gets terrible results, even worse than baseline. Because synthetic samples from this strategy lack of one-to-one correspondence to original samples, and gradients can't be back propagated from these synthetic samples.

6) Similarity distribution: To understand how our method affects the distribution of the similarity, Fig. 12 shows the histograms of similarity of the positive and negative pairs with or without our method on test set. It's easy to find that the similarity distribution get improved by our method since negative pairs and positive pairs are separated further.

Settings	CUB	✓	CARS	✓
Baseline (Triplet)	54.4		69.0	
$N = 3$	55.5	56.7	70.3	71.8
$N = 5$	55.8	57.2	70.6	71.8
$N = 7$	55.9	57.1	70.9	71.9
Use all samples	**58.9**	58.7	**72.4**	72.2

Settings	CUB			CARS		
	Trip	Cont	MS	Trip	Cont	MS
Baseline	54.4	57.3	56.2	69.0	74.1	77.0
fixed	54.7	55.9	57.8	70.1	74.8	79.2
dynamic	**58.9**	**61.7**	**62.1**	**72.4**	**80.8**	**84.2**

G. Visualization

1) Retrieval results: We present the qualitative retrieval results of our model on four datasets in Fig. 13. Our method can help models to learn more discriminative and robust embeddings, even if intra-class appearance variation is significantly large. For examples, there are misleading poses (Fig. 13c), viewpoints (Fig. 13b), background (Fig. 13a) , or colors from various semantically similar samples in the same class, but our model achieves great retrieval results qualitatively.

2) Embedding space: To better qualitatively evaluate the embedding space, we also illustrate the T-SNE [75] visualizations of the embedding space on two datasets. As shown in Fig. 14, the synthetic samples are generated in-between original samples and overlap most areas of class-clusters, which guarantee the authenticity of synthetic samples.

3) Generation in original domains: To demonstrate our method generate semantically meaningful samples, we use the utilize the pretrained Big-GAN [76] to map the synthetic samples back to the pixel space [43] to explicitly show the augmentations of intra-class variations. The visualization results in Fig. 15 show synthetic samples have various of changes of semantic strengths and diversity, e.g., viewpoint, background, and color, which are reliable and robust.

VII. CONCLUSION

In this paper, we present a novel intra-class adaptive augmentation framework for deep metric learning. We reasonably estimate intra-class variations and generate adaptive virtual samples. Especially, we discover the similarity correlation between classes and their variations, and propose the neighbor correction to revise inaccurate estimations for the few-samples class. Our idea of integrating neighbor information provides a new insight for solving defective distribution estimations of insufficient samples, which is a fundamental dilemma for statistical learning. Moreover, we believe the similarity
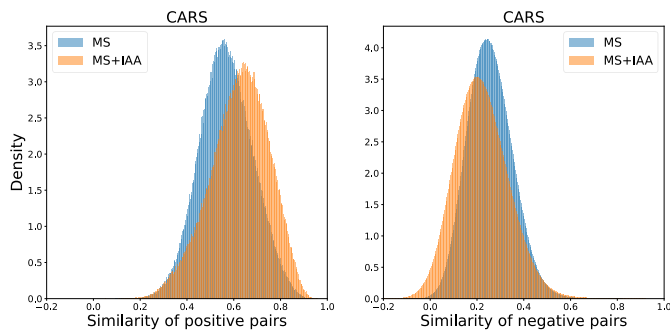
Fig. 12. **Histogram of cosine similarities of positive pairs and negative pairs with or without our method.** We take all sample-pairs in the dataset.
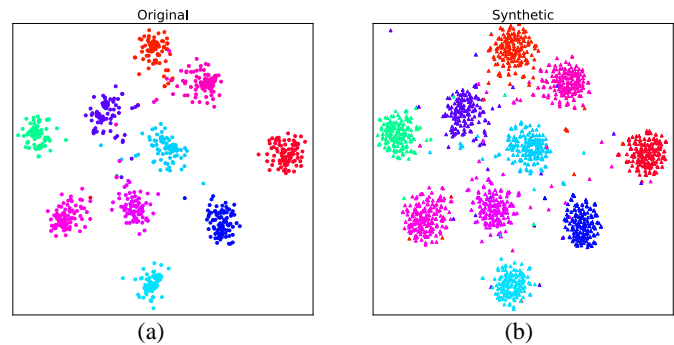

Fig. 14. **T-SNE visualization of the embedding space from 10 randomly selected classes in CARS.** Different colors represent different classes. (a) Original samples, (b) Synthetic samples. All of them are all from train set.
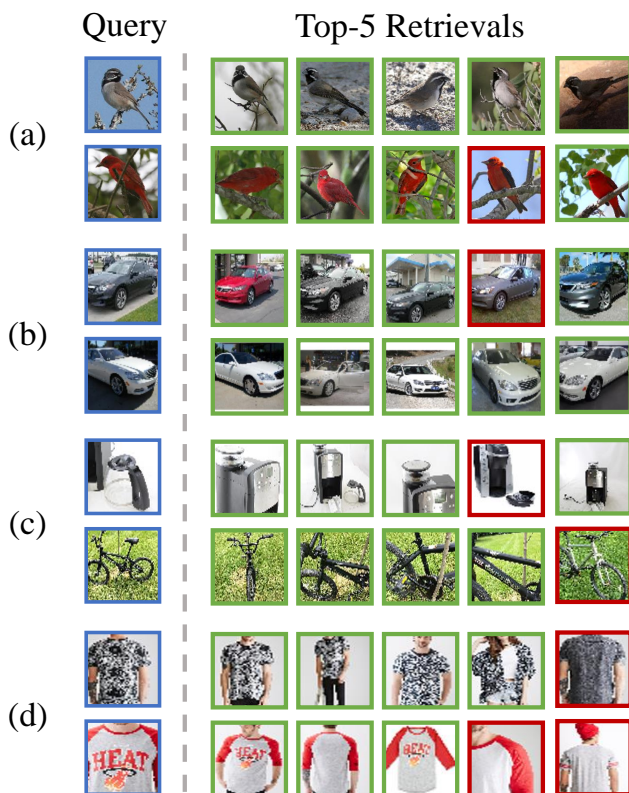

Fig. 13. **Qualitative retrieval results with our method on MS loss.** (a), (b), (c), (d) is for four datasets (CUB, CARS, SOP, and InShop) respectively. For each query image (leftmost with blue edge), the top-5 recall results are presented with left-to-right ranking by relative distances. Correct recalls are highlighted with green, while incorrect recalls are red.

Fig. 15. **Visualization of synthetic samples in the pixel space.** The first columns are real images in CARS, and rest are related synthetic images by our generation method. We use the Big-GAN [76] to map back pixel domain.

correlation is a general discovery not only existing in metric learning, and our neighbor correction can be applied to various artificial intelligence tasks.

REFERENCES

[1] J. Yang, J. Luo, J. Yu, and T. S. Huang, "Photo stream alignment and summarization for collaborative photo collection and sharing," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1642–1651, 2012.

[2] X. Yao, D. She, H. Zhang, J. Yang, M.-M. Cheng, and L. Wang, "Adaptive deep metric learning for affective image retrieval and classification," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[3] D. H. Kim and B. C. Song, "Deep metric learning with manifold class variability analysis," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.
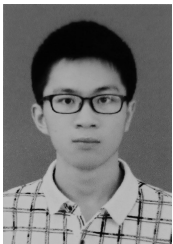
[4] L. Wu, Y. Wang, J. Gao, and X. Li, "Where-and-when to look: Deep siamese attention networks for video-based person re-identification," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1412–1424, 2019.

[5] C. Zhao, X. Lv, Z. Zhang, W. Zuo, J. Wu, and D. Miao, "Deep fusion feature representation learning with hard mining center-triplet loss for person re-identification," *IEEE Transactions on Multimedia*, vol. 22, no. 12, pp. 3180–3195, 2020.

[6] B. Xu, J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Interact as you intend: Intention-driven human-object interaction detection," *IEEE Transactions on Multimedia*, vol. 22, no. 6, pp. 1423–1432, 2020.

[7] P. Cui, S. Liu, and W. Zhu, "General knowledge embedded image representation learning," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 198–207, 2018.

[8] Y. Yang, X. Zhang, M. Yang, and C. Deng, "Adaptive bias-aware feature generation for generalized zero-shot learning," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[9] S. Xia, M. Shao, J. Luo, and Y. Fu, "Understanding kin relationships in a photo," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1046–1056, 2012.

[10] W. Huang, S. Zhang, P. Zhang, Y. Zha, Y. Fang, and Y. Zhang, "Identity-aware facial expression recognition via deep metric learning based on synthesized images," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[11] A. Zheng, M. Hu, B. Jiang, Y. Huang, Y. Yan, and B. Luo, "Adversarial-metric learning for audio-visual cross-modal matching," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[12] F. Chen, J. Shao, Y. Zhang, X. Xu, and H. T. Shen, "Interclass-relativity-adaptive metric learning for cross-modal matching and beyond," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[13] H. Cui, L. Zhu, J. Li, Y. Yang, and L. Nie, "Scalable deep hashing for large-scale social image retrieval," *IEEE Transactions on Image Processing*, vol. 29, pp. 1271–1284, 2020.

[14] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Deep collaborative multi-view hashing for large-scale image search," *IEEE Transactions on Image Processing*, vol. 29, pp. 4643–4655, 2020.

[15] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[17] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5022–5030.

[18] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6398–6407.

[19] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2840–2848.

[20] Y. Suh, B. Han, W. Kim, and K. M. Lee, "Stochastic class-based hard example mining for deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7251–7259.

[21] T. Milbich, K. Roth, H. Bharadhwaj, S. Sinha, Y. Bengio, B. Ommer, and J. P. Cohen, "Diva: Diverse visual feature aggregation for deep metric learning," in *European Conference on Computer Vision*. Springer, 2020, pp. 590–607.

[22] M. Rolínek, V. Musil, A. Paulus, M. Vlastelica, C. Michaelis, and G. Martius, "Optimizing rank-based metrics with blackbox differentiation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7620–7630.

[23] Z. Fu, Y. Li, Z. Mao, Q. Wang, and Y. Zhang, "Deep metric learning with self-supervised ranking," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

[24] B. Chen, P. Li, Z. Yan, B. Wang, and L. Zhang, "Deep metric learning with graph consistency," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, vol. 35, 2021.

[25] Y. Zhao, Z. Jin, G.-j. Qi, H. Lu, and X.-s. Hua, "An adversarial approach to hard triplet generation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 501–517.

[26] W. Zheng, Z. Chen, J. Lu, and J. Zhou, "Hardness-aware deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 72–81.

[27] G. Gu and B. Ko, "Symmetrical synthesis for deep metric learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 10 853–10 860.

[28] B. Ko and G. Gu, "Embedding expansion: Augmentation in embedding space for deep metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7255–7264.

[29] G. Gu, B. Ko, and H.-G. Kim, "Proxy synthesis: Learning with synthetic classes for deep metric learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[30] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, "Deep adversarial metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2780–2789.

[31] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in neural information processing systems*, 2016, pp. 1857–1865.

[32] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4004–4012.

[33] A. Zhai and H. Wu, "Classification is a strong baseline for deep metric learning," in *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*. BMVA Press, 2019, p. 91.

[34] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, "Softtriple loss: Deep metric learning without triplet sampling," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6450–6458.

[35] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3238–3247.

[36] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 360–368.

[37] X. Wang, H. Zhang, W. Huang, and M. R. Scott, "Cross-batch memory for embedding learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6388–6397.

[38] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, and N. M. Robertson, "Ranked list loss for deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5207–5216.

[39] X. Lin, Y. Duan, Q. Dong, J. Lu, and J. Zhou, "Deep variational metric learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 689–704.

[40] Z. Fu, Z. Mao, C. C. Yan, A. an Liu, H. Xie, and Y. Zhang, "Self-supervised synthesis ranking for deep metric learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 4736–4750, 2022.

[41] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *International conference on machine learning*. PMLR, 2013, pp. 552–560.

[42] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7064–7073.

[43] Y. Wang, X. Pan, S. Song, H. Zhang, G. Huang, and C. Wu, "Implicit semantic data augmentation for deep networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 12 635–12 644.

[44] Q. Cai, Y. Wang, Y. Pan, T. Yao, and T. Mei, "Joint contrastive learning with infinite possibilities," in *Advances in Neural Information Processing Systems*, 2020.

[45] S. Yang, L. Liu, and M. Xu, "Free lunch for few-shot learning: Distribution calibration," in *International Conference on Learning Representations*, 2021.

[46] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "Metasaug: Meta semantic augmentation for long-tailed visual recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[47] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected papers of hirotugu akaike*. Springer, 1998, pp. 199–213.

[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[49] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[50] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.

[51] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[53] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 178–178.

[54] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.

[55] J. L. Myers and A. D. Well, "Research design and statistical analysis," *Japanese Journal of National Medical Services*, vol. 55, 2013.

[56] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *International Conference on Computer Vision*, 2002.

[57] Y. Em, F. Gag, Y. Lou, S. Wang, T. Huang, and L.-Y. Duan, "Incorporating intra-class variance to fine-grained visual recognition," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 1452–1457.

[58] S. Yang, S. Wu, T. Liu, and M. Xu, "Bridging the gap between few-shot and many-shot learning via distribution calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[59] J. Liu, Y. Sun, C. Han, Z. Dou, and W. Li, "Deep representation learning on long-tailed data: A learnable embedding augmentation perspective," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2967–2976, 2020.

[60] Y. Hu, V. Gripon, and S. Pateux, "Leveraging the feature distribution in transfer-based few-shot learning," in *ICANN*, 2021.

[61] K. Lee, S. Kim, S. Hong, and S. Kwak, "Learning to generate novel classes for deep metric learning," *ArXiv*, vol. abs/2201.01008, 2022.

[62] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1861–1870.

[63] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "Bier-boosting independent embeddings robustly," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5189–5198.

[64] ——, "Deep metric learning with bier: Boosting independent embeddings robustly," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[65] K. Roth, B. Brattoli, and B. Ommer, "Mic: Mining interclass characteristics for improved metric learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8000–8009.

[66] A. Sanakoyeu, V. Tschernezki, U. Buchler, and B. Ommer, "Divide and conquer the embedding space for metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 471–480.

[67] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1096–1104.

[68] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2167–2175.

[69] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[72] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 681–699.

[73] T. Yuan, W. Deng, J. Tang, Y. Tang, and B. Chen, "Signal-to-noise ratio: A robust distance metric for deep metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4815–4824.

[74] W. Zhao, Y. Rao, Z. Wang, J. Lu, and J. Zhou, "Towards interpretable deep metric learning with structural matching," *arXiv preprint arXiv:2108.05889*, 2021.

[75] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[76] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019.

Bo Hu received the B.Sc. degree in Computer Science from the University of Science and Technology of China, Hefei, China in 2007, and the Ph.D. degree in Electrical and Computer Engineering from the University of Alberta, Edmonton, AB, Canada in 2013. Currently, he is an associate professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include computational social science, recommender systems, data mining and information retrieval.

An-An Liu received the Ph.D. degree from Tianjin University in 2010. He is currently a Full Professor with the School of Electronic Engineering, Tianjin University, China. His research interests include computer vision and machine learning.

Zheren Fu received the B.Sc. degree from University of Science and Technology of China, Hefei, China, in 2020. He is currently working toward the Ph.D. degree with the University of Science and Technology of China, Hefei, China. His research interests mainly cover deep metric learning, cross-modal retrieval, and image-text matching.

Yongdong Zhang (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. He has authored more than 100 refereed journal and conference papers. His research interests include multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology. He was the recipient of the Best Paper Awards in PCM 2013, ICIMCS 2013, and ICME 2010, and the Best Paper Candidate in ICME 2011. He serves as an Editorial Board Member for the Multimedia Systems Journal and the IEEE TRANSACTIONS ON MULTIMEDIA.

Zhendong Mao received the Ph.D. degree in computer application technology from the Institute of Computing Technology, Chinese Academy of Sciences, in 2014. He is currently a professor with the School of Cyberspace Science and Technology, University of Science and Technology of China, Hefei, China. He was an assistant professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, from 2014 to 2018. His research interests include computer vision, natural language processing, and cross-modal understanding.