Learning to Generate Novel Classes for Deep Metric Learning

Kyungmoon Lee[1]
kyungmoon@postech.ac.kr
Sungyeon Kim[1]
sungyeon.kim@postech.ac.kr
Seunghoon Hong[2]
seunghoon.hong@kaist.ac.kr
Suha Kwak[1]
suha.kwak@postech.ac.kr

[1] POSTECH
Pohang, South Korea
[2] KAIST
Daejeon, South Korea

Abstract

Deep metric learning aims to learn an embedding space where the distance between data reflects their class equivalence, even when their classes are unseen during training. However, the limited number of classes available in training precludes generalization of the learned embedding space. Motivated by this, we introduce a new data augmentation approach that synthesizes novel classes and their embedding vectors. Our approach can provide rich semantic information to an embedding model and improve its generalization by augmenting training data with novel classes unavailable in the original data. We implement this idea by learning and exploiting a conditional generative model, which, given a class label and a noise, produces a random embedding vector of the class. Our proposed generator allows the loss to use richer class relations by augmenting realistic and diverse classes, resulting in better generalization to unseen samples. Experimental results on public benchmark datasets demonstrate that our method clearly enhances the performance of proxy-based losses.

1 Introduction

Deep metric learning is the task of learning an embedding space where data of the same class are placed closely so that the distance between data reflects their class equivalence. It has been a driving force behind recent advances in numerous computer vision and machine learning tasks including image retrieval [15, 31, 32], face identification [6, 30], person re-identification [4], and representation learning [3, 15]. The main reason for adopting deep metric learning in these tasks is its generalization capability; the learned embedding space is expected to be well generalized to unseen classes so that it can be used to predict the class equivalence of a pair of data even when their classes are unavailable during training.

Various ways to improve the performance of deep metric learning have been studied so far, such as advanced loss functions [16, 26, 31, 32, 36], ensemble methods [17, 27], regularization techniques [14, 25], sample mining [12, 37], and sample generation [18, 21].

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

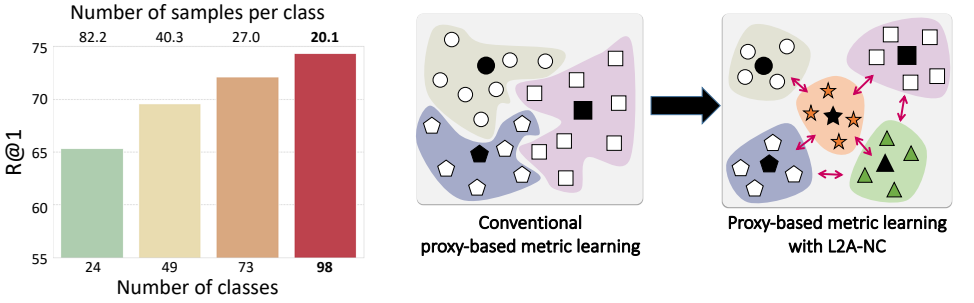

Figure 1: Our motivation and conceptual diagram. **Left**: Accuracy in Recall@1 versus the number of training classes with Proxy-Anchor loss on the Cars-196 dataset. The total number of samples used for training is fixed (Please see the first footnote.). **Right**: Comparison between proxy-based metric learning with and without L2A-NC. Black, empty, and colored nodes denote proxies, real embedding vectors, and synthetic embedding vectors of augmented classes, respectively. Also, different shapes indicate different classes.

Although the effectiveness of these methods has been demonstrated, we believe their generalization capability could be further improved in the sense that they are inherently limited only to classes available in a training set.

In this paper, we argue that the number of training classes is relatively more important than that of training samples in deep metric learning. We demonstrate the importance of the number of training classes by validating generalization ability while varying the number of training classes but fixing the total number of samples used in training. Specifically, for the test split of Cars-196 (*i.e.* 8,131 images of the latter 98 classes), we measure the performances of models whose size of training classes ranges from 25% (24 classes) of total training classes to 100% (98 classes); reversely, the number of samples per training class decreases from 82.2 (for 25%) to 20.1 (for 100%) so that all models are trained with the same training sample size [1]. As shown in Figure 1 (left), a larger number of classes lead to better performance although the number of samples per class becomes smaller. It is natural since more diverse classes would offer richer semantic relations between training classes. According to this observation, we remark the existing sample generation methods for deep metric learning are limited to leverage impoverished relations between given training classes.

Meanwhile, a recently proposed method [16] aims to synthesize new classes through linear interpolation between data representations of real classes so that inter-class relations with synthetic classes can yield better supervisory signals beyond relations between real classes. However, since this approach heavily relies on the data representations of real classes, the generated classes cannot cover a broad range of data characteristics, resulting in limitations in improving generalization.

In this paper, we introduce a novel data augmentation method that resolves the aforementioned issues. The key idea is to synthesize novel classes and their embedding vectors through a conditional generative model, which is an auxiliary module trained together with the main embedding network. We thus call it *Learning to Augment Novel Classes*, dubbed *L2A-NC*. Specifically, the conditional generative model, given a class label and a noise (*i.e.*, a latent variable), produces a random embedding vector of the class. The model is trained by a metric learning loss like the main embedding network so that the novel classes and their

[1] The fixed size of training data is 1,975 and # of samples per training class is uniform.

embedding vectors become discriminative. At the same time, it is regularized to produce realistic embedding vectors by minimizing divergence between distributions of synthetic and real embedding vectors. As a result, novel classes have distributions that fit in between those of real classes in the learned embedding space (Figure 1 (right)). The proposed method thus can synthesize realistic and discriminative novel classes thanks to the powerful expressiveness of deep neural networks trained with carefully designed loss functions. Consequently, these novel classes unavailable in original data help the main embedding network learn a better generalized embedding space. In summary, the contribution of this paper is three-fold:

- We introduce a novel data augmentation framework for deep metric learning, called L2A-NC, which *synthesizes novel classes and corresponding embedding vectors* to be augmented as additional training data through a conditional generative model.
- We design architecture and its training strategy that enable the generator to define novel classes and produce realistic and discriminative embedding vectors.
- It is demonstrated on public benchmarks that L2A-NC clearly enables non-trivial performance improvement of proxy-based losses.

2 Related Work

Proxy-based losses for deep metric learning. The loss functions for metric learning can be categorized into two types as *pair-based* and *proxy-based* losses. The pair-based losses are based on the pairwise relations between data in the embedding space. However, they have high training complexity since the number of tuples increases exponentially with the number of training data, forcing a careful tuple sampling technique. Proxy-based losses are proposed to alleviate the complexity issue by replacing samples with a small number of proxies, which are learnable parameters representing each class. Proxy-NCA [26] is the first proxy-based method that pushes a sample to its positive proxy but repels against its negative proxies. Similarly, SoftTriple [28] assigns multiple proxies within one class to reflect intra-class variance. Proxy-Anchor [16] leverages data-to-data relations via forming a proxy as an anchor.

Sample generation for deep metric learning. Sample generation methods are motivated to provide potentially informative samples which do not exist in the original data. [8, 21, 38] exploit generative models to synthesize synthetic embedding vectors. To reduce training complexity, [11, 18] are proposed to generate synthetic embedding vectors by simple algebraic operation in the embedding space. However, these techniques are only coupled with pair-based losses and limited to synthesizing embedding vectors of existing classes.

Virtual class synthesis. Recently, several approaches have been proposed to utilize virtual classes in various areas. Virtual softmax [4] injects additional weight as a virtual negative class for softmax function. Proxy Synthesis [10] exploits virtual classes synthesized by linear interpolation between data representations of real classes. Different from them, our method synthesizes novel classes by a generative model, which utilizes the expressiveness of deep neural networks. In addition, most recently, VirFace [20] has been proposed to exploit unlabeled data as samples of virtual classes. Similar to our method, this approach introduces a VAE network to generate instances of virtual classes, but it also requires additional unlabeled data to train a generative model.

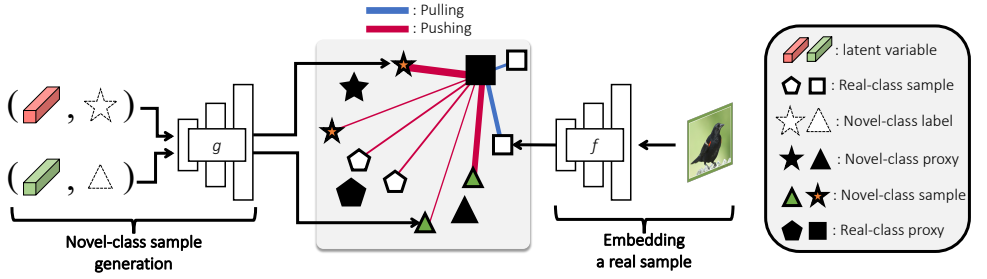

Figure 2: The overall framework of L2A-NC. Given a novel class label and a latent variable, the conditional generative model g produces an embedding vector of the class while the main model f computes that of training images. In our framework, a proxy-based loss takes both of real and synthetic embedding vectors (Please refer Sec. 3 for details).

3 Our Approach

As a way to improve the generalization of deep metric learning, we propose a new data augmentation method called L2A-NC, which synthesizes novel classes and their embedding vectors. Our method learns and utilizes a conditional generative network that models novel classes and produces their embedding vectors, which are incorporated with any proxy-based losses to help them construct a more discriminative embedding space. The overall framework of L2A-NC is illustrated in Figure. 2. In the rest of this section, we review the proxy-based metric learning losses, present details of the conditional generator, describe the training procedure incorporating L2A-NC, and analyze the effectiveness of novel classes with comparison to an existing class augmentation method.

3.1 Background: Proxy-based Losses

Suppose that we aim to learn an embedding network f parameterized by θ_f and learnable proxies $P = [p_1, ..., p_C]$. Let $X = [x_1, ..., x_N]$ be embedding vectors (*i.e.*, the outputs of f) and $Y = [y_1, ..., y_N]$ be their corresponding labels, where $y_i \in \{1, ..., C\}$. A proxy-based loss optimized with respect to θ_f and P is denoted by $J_{met}(X, Y, P)$.

In practice, however, proxy-based losses can be further enhanced in the sense that they are inherently limited only to training classes. From this perspective, this paper proposes a new data augmentation method to synthesize novel classes and their embedding vectors.

3.2 Conditional Generator

To synthesize novel classes, we train a conditional generator g which produces embedding vectors $\tilde{X} = [\tilde{x}_1, ..., \tilde{x}_M]$ of novel classes given corresponding labels $\tilde{Y} = [\tilde{y}_1, ..., \tilde{y}_M]$, where $\tilde{y}_j \in \{C+1, ..., C+\tilde{C}\}$, and latent variables Z for the stochastic generation:

$$\tilde{X} = g(\tilde{Y}, Z). \tag{1}$$

Besides, corresponding proxies of novel classes $\tilde{P} = [p_{C+1}, ..., p_{C+\tilde{C}}]$ are learnable, thus, first randomly initialized and updated via a specified proxy-based loss just as real-class proxies P. Along with this conditional generation process, it is necessary to regularize our generator to produce realistic embedding vectors which are not far from the distribution of real

embedding vectors; this is in line with previous data augmentation methods for deep metric learning [8, 30, 38]. Furthermore, novel classes have to be discriminative so that they become diverse and independent from each other like real classes of the original dataset.

Loss functions. To guarantee that our generator produces realistic and discriminative embedding vectors, we introduce two loss functions for its training: A divergence loss J_{div} and a proxy-based loss J_{met}. First, the generator is encouraged to fit its generated distributions in between those of real embedding vectors by minimizing the divergence loss.

As the Wasserstein distance has demonstrated its effectiveness in generative models [1, 9] and other applications [22, 39], it is a good candidate for the divergence loss defined as:

$$\mathcal{W}(p,q) = \inf_{\gamma \in \Pi_{(p,q)}} E_{x_p,\, x_q \sim \gamma}[c(x_p, x_q)], \tag{2}$$

where $\Pi(p,q)$ is the set of all joint distributions $\gamma(x_p, x_q)$ and $c(\cdot,\cdot)$ denotes a cost function. This distance is usually interpreted as the minimum cost to turn the distribution q into the distribution p. However, since the optimization problem in Eq. (2) is generally intractable, we resort to the entropy-regularized Sinkhorn distance [6]. In addition, to evaluate the Wasserstein distance on given mini-batches of X_p, X_q, we choose *Sinkhorn AutoDiff* [9] proposed as an approximate of the distance:

$$\mathcal{W}_c(X_p, X_q) = \inf_{M \in \mathcal{M}} [M \odot C], \tag{3}$$

where the cost function c becomes the cost matrix C, where $C_{i,j} = c(x_i^p, x_j^q)$, and the coupling distribution γ becomes the soft matching matrix M whose all rows and columns sum to one. Although it is able to perform efficient optimization on GPUs, its gradients become no longer an unbiased gradient estimator when using mini-batches. Therefore, we finally adopt *Mini-batch Energy Distance* [29], which results in unbiased mini-batch gradients, as the divergence loss which is given by

$$J_{div}(X, \tilde{X}) = 2E[\mathcal{W}_c(X_1, \tilde{X}_1)] - E[\mathcal{W}_c(X_1, X_2)] - E[\mathcal{W}_c(\tilde{X}_1, \tilde{X}_2)], \tag{4}$$

where X divided into X_1 and X_2 is a mini-batch from real data and $\tilde{X}$ divided into $\tilde{X}_1$ and $\tilde{X}_2$ is a mini-batch from generated data. For a cost function c, we adopt the cosine distance.

Second, we train the generator to produce discriminative embedding vectors. To this end, the generator aims to minimize a proxy-based loss that takes not only novel-class data but real-class data so that novel classes become diverse and offer richer class relations to an embedding model. Formally, the objective is given by

$$J_{met}(X \cup \tilde{X}, Y \cup \tilde{Y}, P \cup \tilde{P}). \tag{5}$$

3.3 Proxy-based Metric Learning with L2A-NC

This section illustrates the overall pipeline of our method. We first pretrain the embedding function f alone via a specific proxy-based loss J_{met}:

$$\min_{\theta_f, P} J_{met}(X, Y, P). \tag{6}$$

Then, we pretrain the conditional generator to optimize $J_{div}(X, \tilde{X})$ and $J_{met}(\tilde{X}, \tilde{Y}, \tilde{P})$ in advance since it is difficult for the generator to synthesize realistic and discriminative embedding vectors from scratch. Finally, in the joint training phase, the two networks f and g are

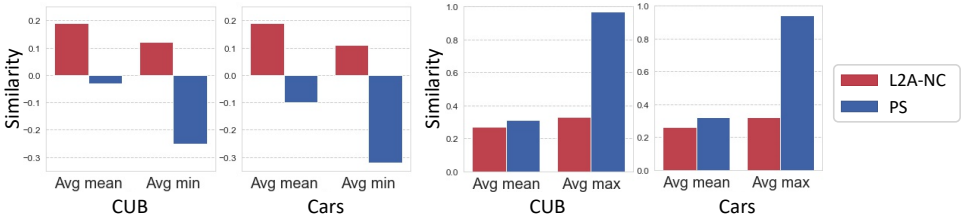

Figure 3: Cosine-similarity comparison. **Left**: One between embedding vectors and proxies of the same novel class. **Right**: One between proxies of real classes and those of novel classes.

learned by optimizing the following common objective:

$$\min_{\theta_f,\theta_g,P,\tilde{P}} J_{met}(X \cup \tilde{X}, Y \cup \tilde{Y}, P \cup \tilde{P}) + \lambda_{div} J_{div}(X, \tilde{X}), \tag{7}$$

where λ_{div} is a hyperparameter to balance the two losses. Note that J_{div} is optimized with respect to $\tilde{X}$ only, and encourages the generator to produce realistic embedding vectors in the joint training phase also. The overall training pipeline of L2A-NC is summarized in Section 1 of the supplementary material.

3.4 Analysis of L2A-NC

In this section, we briefly review Proxy Synthesis (PS) [10], an existing class augmentation method. Next, we analyze and compare the effectiveness of novel classes from PS and ours.

Review of Proxy Synthesis (PS). As an existing method, PS synthesizes a synthetic proxy and a synthetic embedding vector by linear interpolation between proxies of different real classes, and embedding vectors of different real classes, respectively as

$$(\tilde{p}, \tilde{x}) = (I_{\lambda_{ps}}(p_i, p_j), I_{\lambda_{ps}}(x_i, x_j)) \tag{8}$$

where $y_i \neq y_j$, $\tilde{x} \in \tilde{X}$, $\tilde{p} \in \tilde{P}$, and $I_{\lambda_{ps}}(a,b) = \lambda_{ps} a + (1 - \lambda_{ps} b)$ is a linear interpolation function with $\lambda_{ps} \sim Beta(\alpha, \alpha)$ for $\alpha \in (0, \infty)$, and $\lambda_{ps} \in [0,1]$.

Comparison to PS on the validity of novel classes. As previously discussed, learning with diverse classes improves performance as they allow to provide richer semantic relations. In this context, we verify that the proposed method can generate semantic and diverse classes like real classes, and compare it with PS[2]. To this end, suppose $s(v_i, v_j)$ denotes the cosine-similarity between two vectors, v_i and v_j. Let $\tilde{x}_i$ and $\tilde{p}_i = p_{C+i}$ be an embedding vector and proxy of an arbitrary novel class label $\tilde{y}_i$. Next, we consider two cosine-similarities: one between embedding vectors and a proxy of the same novel class and another between proxies of real classes and those of novel classes (*i.e.* $s(\tilde{x}_i, \tilde{p}_i)$ and $s(p_j, \tilde{p}_i)$, $\forall i \in \{1, ..., \tilde{C}\}, \forall j \in \{1, ..., C\}$). As shown in Figure 3 (left), L2A-NC clearly shows high values of $s(\tilde{x}_i, \tilde{p}_i)$ while PS shows negative values on both mean and minimum on average. This suggests that L2A-NC generates novel classes that better preserve semantic properties while classes generated by PS fail to preserve their own semantics. Figure 3 (right) shows that PS shows higher values of $s(p_j, \tilde{p}_i)$ than L2A-NC on both mean and maximum on average. This suggests that

[2]We adapt the official code from https://github.com/navervision/proxy-synthesis

Method	Batch	CUB			Cars			SOP			In-Shop		
		R@1	R@2	R@4	R@1	R@2	R@4	R@1	R@10	R@100	R@1	R@10	R@20
Norm-softmax	128	64.9	76.0	84.3	83.3	89.7	94.1	78.6	90.5	96.0	90.4	97.7	98.5
+ PS [40]	128	66.0	76.6	85.0	84.7	90.7	94.6	**79.6**	90.9	**96.2**	91.5	98.1	98.7
+ L2A-NC	128	**66.8**	**77.0**	**85.6**	**86.0**	**91.8**	**95.2**	79.4	**91.0**	**96.2**	**91.9**	**98.2**	**98.8**
Cosface [43]	128	65.7	76.2	84.7	83.6	89.9	94.2	78.6	90.4	95.8	90.7	97.6	98.3
+ PS [40]	128	66.6	76.8	84.6	84.6	90.8	94.3	**79.3**	90.7	95.9	91.4	97.8	98.5
+ L2A-NC	128	**67.6**	**77.5**	**85.3**	**85.2**	**90.8**	**94.7**	**79.3**	**91.0**	**96.2**	**91.9**	**98.2**	**98.7**
Proxy-NCA [29]	128	65.1	76.1	85.0	83.7	90.4	94.1	78.1	90.0	95.9	90.0	97.7	98.4
+ PS [40]	128	66.4	76.8	85.1	84.5	90.8	94.4	79.1	90.6	95.9	91.4	98.0	98.7
+ L2A-NC	128	**67.7**	**77.9**	**86.1**	**85.9**	**91.9**	**95.3**	**79.3**	**91.0**	**96.3**	**91.7**	**98.3**	**98.9**
SoftTriple [33]†	128	66.3	76.8	84.7	84.9	90.5	94.3	79.0	90.7	96.1	91.1	97.8	98.4
+ PS [40]	128	66.6	76.8	85.1	85.3	91.0	94.8	**79.5**	90.6	96.0	**91.8**	98.1	**98.7**
+ L2A-NC	128	**68.0**	**78.0**	**85.4**	**86.0**	**91.4**	**95.0**	79.4	**91.1**	**96.3**	91.6	**98.2**	**98.7**
Proxy-Anchor [29]‡	180	69.1	78.9	86.1	86.4	91.9	95.0	79.2	90.7	96.2	91.9	98.1	98.7
+ PS [40]	180	69.2	**79.5**	**87.2**	86.9	92.4	95.2	79.8	90.9	**96.4**	91.9	98.2	**98.8**
+ L2A-NC	180	**69.7**	79.1	86.4	**87.9**	**92.8**	**95.4**	**79.9**	**91.2**	96.1	**92.3**	**98.3**	98.7
Average boost	PS [40]	(+0.7)	(+0.5)	(+0.4)	(+0.8)	(+0.7)	(+0.3)	**(+0.8)**	(+0.3)	(+0.1)	(+0.8)	(+0.3)	(+0.2)
	L2A-NC	**(+1.7)**	**(+1.1)**	**(+0.8)**	**(+1.8)**	**(+1.3)**	**(+0.8)**	**(+0.8)**	**(+0.6)**	**(+0.2)**	**(+1.0)**	**(+0.5)**	**(+0.3)**

Table 1: Comparison with the state-of-the-art methods. Image retrieval performance is measured as Recall@K (%) on the public benchmark datasets. †: For a fair comparison, we reproduced SoftTriple with the batch size of 128 using the author's official code and replace the original SoftTriple whose batch size is 32. ‡: It is reported by the authors.

Method	CUB					Cars				
	NMI	F1	R@1	R@2	R@4	NMI	F1	R@1	R@2	R@4
Triplet†	58.1	24.2	48.3	61.9	73.0	57.4	22.6	60.3	73.4	83.5
+ EE [18]	**60.5**	**27.0**	51.7	63.5	74.5	**63.1**	**32.0**	71.6	80.7	87.5
+ PS [40]	58.1	24.8	50.9	62.0	72.8	57.9	24.0	62.8	73.8	82.3
+ L2A-NC	59.4	26.0	**53.6**	**65.3**	**75.6**	61.6	29.2	**73.0**	**81.9**	**88.2**
MS [36]	62.8	31.2	56.2	68.3	79.1	62.4	30.2	75.0	83.1	89.5
+ EE [18]	63.3	32.5	57.4	68.7	79.5	63.5	33.5	76.1	84.2	89.8
+ PS [40]	61.1	29.5	55.9	68.1	78.1	58.0	25.0	71.8	80.9	87.6
+ L2A-NC	**66.1**	**35.8**	**60.6**	**72.5**	**82.2**	**68.1**	**37.9**	**81.2**	**88.4**	**93.0**
EE [18]	(+1.4)	(+2.0)	(+2.3)	(+1.0)	(+0.9)	(+3.4)	(+6.5)	(+6.1)	(+4.2)	(+2.1)
PS [40]	(-0.8)	(-0.5)	(+1.1)	(0.0)	(-0.6)	(-1.9)	(-1.9)	(-1.9)	(-0.3)	(-1.5)
L2A-NC	**(+2.3)**	**(+3.2)**	**(+4.9)**	**(+3.8)**	**(+2.9)**	**(+4.9)**	**(+7.1)**	**(+9.5)**	**(+6.9)**	**(+4.1)**

Table 2: Comparison with the existing pair-based losses. NMI and F1 (%) are measured for clustering performance. Recall@K (%) is measured for retrieval performance. † denotes the triplet loss with hard tuple mining.

PS synthesizes classes that are highly redundant to real classes and lead to limited signals while L2A-NC generates diverse classes which provide richer semantic relations.

4 Experiments

In this section, to demonstrate the superiority of our framework, we compare L2A-NC with state-of-the-art methods and provide an in-depth analysis. Especially, we remark that L2A-NC also can be seamlessly incorporated with pair-based losses. Therefore, we further evaluate L2A-NC on pair-based losses as well as proxy-based losses.

4.1 Setup

Datasets. Combinations of L2A-NC and proxy-based losses are evaluated on benchmark datasets for deep metric learning: CUB-200-2011 (CUB) [34], Cars-196 (Cars) [19], Stanford Online Product (SOP) [32], and In-shop Clothes Retrieval (In-Shop) [23]. For splitting

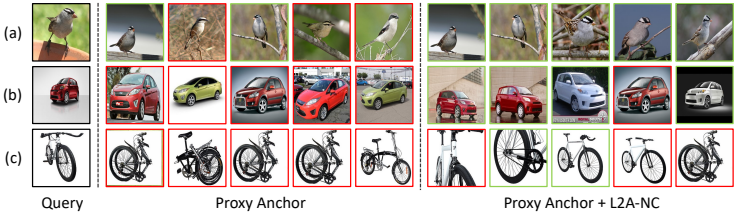

Figure 4: Qualitative results of the vanilla Proxy-Anchor [16] and that combined with L2A-NC on the (a) CUB, (b) Cars, and (c) SOP datasets. Images with green boundary are correct results while those with red boundary are failure cases.

training and test sets, we directly follow the widely used setting in [32]. For comparison with existing pair-based losses, CUB and Cars are adopted.

Network architectures. We adopt ImageNet pre-trained BatchNorm Inception [13] and GoogleNet [33] for experiments associated with proxy-based losses and pair-based losses, respectively. For all experiments, the dimensionality of embedding vectors is 512. The conditional generator consists of 4 fully connected layers; a conditional batch normalization layer [7] is inserted between every pair of layers so that the class information is injected. Also, the dimension of input noise for the generator is fixed to 16 for all experiments.

Details of training. In the pretraining stage, the main embedding models are trained by directly following the setting (*e.g.* batch size) presented in PS [10], while the conditional generator is optimized by AdamW [24] with the learning rate of 1e-4 for all datasets. In the joint training stage, the learning rate of the embedding network is set to 5e-5 for all datasets. For the main embedding models incorporated with pair-based losses, we directly follow the setting in [18] and apply our framework described above.

4.2 Comparison with State of the Art

Results on proxy-based losses. As summarized in Table 1, we observe that L2A-NC incorporated with proxy-based losses achieves non-trivial performance boosts in all datasets. Especially, on the CUB and Cars datasets, our method outperforms the vanilla method and PS [16] by a non-trivial margin (by 1.7%p and 1.0%p, respectively, in average Recall@1.). However, on the SOP and In-Shop datasets, we find the tendency that performance boosts of L2A-NC decreases compared to those on the CUB and Cars datasets, though L2A-NC still shows competitive or better performance boosts than PS. We conjecture this is because the SOP and In-Shop datasets have already a lot of training classes (11,318 and 3,997, respectively) which are about 113 and 40 times compared to those of CUB or Cars. Nevertheless, compared to PS, L2A-NC achieves more performance boosts except for R@1 on the SOP.

Results on pair-based losses. As shown in Table 2, L2A-NC outperforms not only the vanilla loss but also PS. We find that PS fails to boost pair-based losses in most cases even though it shows its effectiveness with proxy-based ones. Since pair-based losses are known to be more vulnerable to noisy labels or outliers than proxy-based losses, we conjecture it is because noisy synthetic classes and their embedding vectors of PS hamper model generalization from scratch, which is discussed in Sec. 3.4. Furthermore, even when compared to the current state-of-the-art sample generation method, Embedding Expansion (EE) [18], dedicated to pair-based losses, our method achieves larger performance improvements. Note that

Dataset	+0%	+25%	+50%	+100%	+200%
CUB	69.1	69.3	69.6	69.5	**69.7**
SOP	79.2	79.3	79.5	79.5	**79.9**
In-Shop	91.9	92.0	92.2	92.2	**92.3**

Dataset	Vanilla	L2A-EC	L2A-NC (Ours)
CUB	69.1	69.2	**69.7**
Cars	86.4	86.8	**87.9**
In-Shop	91.9	91.8	**92.3**

Table 3: Ablation studies on proposed L2A-NC. **Left**: Recall@1 versus the number of novel classes. **Right**: Comparison between L2A-EC and L2A-NC in Recall@1.

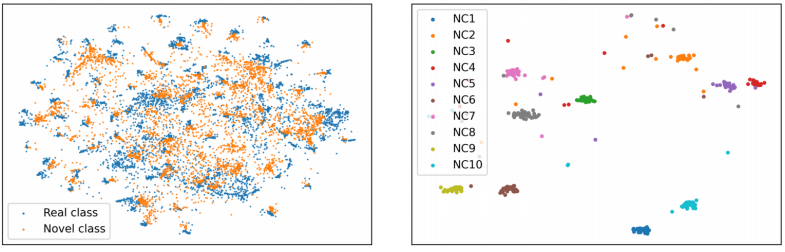

Figure 5: t-SNE visualizations of the learned embedding space. **Left**: Embedding vectors of both real and novel classes. **Right**: Embedding vectors of novel classes only.

no proxies are introduced in any procedure of ours and PS in Table 2 for a fair comparison.

4.3 In-depth Analysis on L2A-NC

Image retrieval results. We further demonstrate the superiority of L2A-NC through qualitative results of image retrieval. Figure. 4 presents image retrieval examples of Proxy-Anchor loss incorporating L2A-NC and those of the vanilla Proxy-Anchor. We observe that results of the vanilla Proxy-Anchor are mostly incorrect and biased towards backgrounds rather than target objects, whereas L2A-NC enables retrieval of correct images regardless of background. For example, the vanilla version failed to retrieve even a single correct image for the Cars and SOP. However, the version incorporating L2A-NC retrieved correct images although colors or viewpoints of target objects are substantially different from those of the query. We demonstrate that L2A-NC functions as desired to regularize the vanilla method not to be biased towards certain prevailing features in the dataset (*e.g.*, background or viewpoint) by augmenting novel classes unavailable in the original data.

Impact of the number of novel classes. Thanks to the architecture of our conditional generator, L2A-NC can define and synthesize an arbitrary number of novel classes. To investigate the impact of the number of novel classes, we evaluate the performance of L2A-NC incorporated with Proxy-Anchor loss on the CUB, SOP, and In-Shop while varying the number of novel classes. As shown in Table 3 (left), the performance increases by adding more novel classes to some degree, but there is an upper bound of the effectiveness; it is natural since too many novel classes may prevent the embedding model from understanding relations between classes, leading to unstable training. In experiments, we fixed the number of novel classes for all datasets: 200% of the number of existing classes.

Importance of generating novel classes. To verify that our performance boost is attributed to the augmentation of embedding vectors of novel classes, we compare L2A-NC with its variant that produces synthetic embedding vectors of *existing classes*; we call it L2A-EC. L2A-NC and L2A-EC are incorporated with Proxy-Anchor loss and compared on the CUB and the Cars in terms of Recall@1. As shown in Table 3 (right), compared to L2A-NC, L2A-EC provided smaller improvements and even slightly worsened the vanilla method. These

Class pair	Mean of minimum KL divergence
Train–Test	56.7
Novel–Test	**27.9**

Table 4: Quantitative analysis about how closely training and novel classes approximate unseen test classes on the CUB.

Method	sec / iter	# of parameters	R@1 boost
Vanilla	0.28	11.85M	-
PS [10]	0.37	11.85M	1.1
Ours	0.43	12.24M	1.9

Table 5: Performance boost over training complexity.

results are consistent with our hypothesis that novel classes unavailable in the original dataset would provide richer signals than existing class data.

***t*-SNE visualization.** To illustrate how embedding vectors of novel classes offer additional signals, we visualize the learned embedding space. As shown in Figure 5, the generator synthesizes embedding vectors that are realistic and discriminative. In detail, Figure 5 (left) shows that embedding vectors of novel classes are located in between those of real classes with forming their own clusters where embedding vectors of real classes do not exist. In addition, Figure 5 (right) shows how novel classes are discriminative to each other. With these results, we demonstrate how novel classes unavailable in the original dataset can offer additional signals, which lead to better generalization on unseen classes.

Relation between novel classes and test classes. To demonstrate that the novel classes we generate affect the robustness on unseen classes, we investigate how well the novel and unseen test classes are aligned to each other in the learned embedding space. Specifically, we quantify the degree of alignment through KL divergence: Each novel class is first matched with its nearest test class with the minimum KL divergence, then the minimum divergence values of all novel classes are averaged. We measure this score for training classes as well for comparison. The results in Table 4 demonstrate that novel classes better approximate test classes than training classes even though the conditional generator of L2A-NC is not aware of test data; this suggests that the proposed method has the potential to improve the generalization of the learned embedding space.

Training complexity. To be a promising option for practical usage, it is quite desirable for L2A-NC to offer an appealing trade-off between performance boost and training complexity. As shown in Table 5[3], compared to PS [10], L2A-NC achieves about 1.7 times more performance boost averaged over every benchmark when incorporated with Proxy-NCA at the cost of only 16% and 3% increase in training time and parameters. This result is brought by our lightweight design: the generator consists of only a few layers and holding in memory novel-class proxies is also negligible as well.

5 Conclusion

We have presented a novel data augmentation method for deep metric learning. Distinct from existing techniques, the proposed method synthesizes novel classes and their embedding vectors through a conditional generative model. Thanks to the carefully designed loss functions and its architecture, the generator synthesizes novel classes that are realistic and discriminative so it can offer richer semantic relations to an embedding model. As a result, our method enabled both proxy-based and pair-based losses to improve the quality and search performance of the learned embedding space.

[3] All the results were produced on a NVIDIA TITAN RTX GPU.

Acknowledgements

This work was supported by the NRF grant, the IITP grant, and R&D program for Advanced Integrated-intelligence for IDentification, funded by Ministry of Science and ICT, Korea (No.2019-0-01906 Artificial Intelligence Graduate School Program–POSTECH, NRF-2021R 1A2C3012728–50%, NRF-2018M3E3A1057306–50%).

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proc. International Conference on Machine Learning (ICML)*, 2017.

[2] Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2018.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. International Conference on Machine Learning (ICML)*, 2020.

[4] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[6] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2013.

[7] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2017.

[8] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[10] Byungsoo Ko Geonmo Gu and Han-Gyu Kim. Proxy synthesis: Learning with synthetic classes for deep metric learning. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[11] Geonmo Gu and Byungsoo Ko. Symmetrical synthesis for deep metric learning. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[12] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

[14] Pierre Jacob, David Picard, Aymeric Histace, and Edouard Klein. Metric learning with horde: High-order regularizer for deep embeddings. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[15] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[17] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[18] Byungsoo Ko and Geonmo Gu. Embedding expansion: Augmentation in embedding space for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[20] Wenyu Li, Tianchu Guo, Pengyu Li, Binghui Chen, Biao Wang, Wangmeng Zuo, and Lei Zhang. Virface: Enhancing face recognition via unlabeled shallow data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[22] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[25] Deen Dayal Mohan, Nishant Sankaran, Dennis Fedorishin, Srirangaraj Setlur, and Venu Govindaraju. Moving in the right direction: A regularization for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[27] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with bier: Boosting independent embeddings robustly. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[28] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

[29] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[31] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2016.

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

[33] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011.

[35] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[38] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.