Diversified Mutual Learning for Deep Metric Learning

Wonpyo Park[⋆1], Wonjae Kim[⋆1], Kihyun You[2,3], and Minsu Cho[2]

[1] Kakao Corporation, Seongam, Korea
[2] POSTECH, Pohang, Korea
[3] Kakao Enterprise, Seongam, Korea

Abstract. Mutual learning is an ensemble training strategy to improve generalization by transferring individual knowledge to each other while simultaneously training multiple models. In this work, we propose an effective mutual learning method for deep metric learning, called *Diversified Mutual Metric Learning*, which enhances embedding models with diversified mutual learning. We transfer relational knowledge for deep metric learning by leveraging three kinds of diversities in mutual learning: (1) *model diversity* from different initializations of models, (2) *temporal diversity* from different frequencies of parameter update, and (3) *view diversity* from different augmentations of inputs. Our method is particularly adequate for *inductive transfer learning* at the lack of large-scale data, where the embedding model is initialized with a pretrained model and then fine-tuned on a target dataset. Extensive experiments show that our method significantly improves individual models as well as their ensemble. Finally, the proposed method with a conventional triplet loss achieves the state-of-the-art performance of Recall@1 on standard datasets: 69.9 on CUB-200-2011 and 89.1 on CARS-196.

Keywords: model diversification, mutual learning, deep metric learning, inductive transfer learning

1 Introduction

Mutual learning [39] is an effective ensemble training strategy to improve generalization ability of learners. In mutual learning, multiple models, *i.e.*, *cohort*, learn and teach each other simultaneously during the training time. For example, in the work of [39], two classifier models with different initializations are trained to predict similar output distributions given the same input data. Despite its effectiveness, previous methods of mutual learning [1,39] are limited to the task of classification. In this work, we propose a mutual learning method for deep metric learning, which aims at learning deep embedding models that maps similar instances to nearby points on a manifold in the embedding space and dissimilar instances apart from each other.

[⋆] Equal contribution

In metric learning, due to small scales of training datasets, the embedding models are commonly initialized using an ImageNet-pretrained network such as Inception-BN [9] or ResNet-50 [6], and then fine-tuned on a target dataset. The common use of inductive transfer learning in metric learning limits the diversity of embedding models in mutual learning. Unlike the previous settings [1,39], where models in the cohort are trained from scratch with diverse initializations, these models with the same backbone network prevents mutual learning from distilling knowledge from each other during training.

To tackle the issue, we propose an effective mutual learning method for deep metric learning, called **D**iversified **M**utual **M**etric Learning (DM^2), which enhances the generalization ability of embedding models with diversified mutual learning. We transfer the relational knowledge with pairwise distance between instances while diversifying the parameter update paths of models of the cohort with our three diversities: (1) **model diversity** to leverage different initializations of models, (2) **temporal diversity** to diversify the frequency of parameter update, and (3) **view diversity** to exploits diverse inputs with different augmentations. Note that the model diversity is naturally induced by training multiple models as in the work of [39], thus being the core of mutual learning per se. In this work, we introduce two additional diversities to mutual learning and also propose a mutual learning method using relation matrices for metric learning.

Extensive experiments on standard image retrieval datasets [29,13,17] for deep metric learning show that the proposed method, DM^2, significantly improves performance of individual models as well as their ensemble, compared to conventional deep metric learning. Moreover, DM^2 is also an effective regularizer that prevents severe overfitting on small training datasets in the latter part of the training. The benefit of DM^2 monotonically grows as we increase the number of models in the cohort. The proposed method combined with a conventional triplet loss achieves the state-of-the-art performance of Recall@1 on the standard datasets: 69.9 on CUB-200-2011 and 89.1 on CARS-196.

2   Related Work

Our work encompasses the studies of deep metric learning, knowledge transfer, and mutual learning. In this section, we summarize each and explain their relations to our work.

2.1   Deep Metric Learning

Loss Function Devising a better loss function is one of the main challenges of the recent deep metric learning studies. One family of the loss functions is pairwise distance-based loss which samples positive or negative pairs within a given mini-batch. The objective of the losses is to let the distance of the positive pair to be small, while that of negative pair to be large. Contrastive loss [5] samples pairs of any two examples, while the Triplet loss [26] samples triplets of anchor, positive and negative examples. In Triplet loss, the distance between

the anchor and positive is trained to be smaller than the distance between anchor and negative by a certain margin. Extended from the two losses, several methods [32,17,28] are proposed to fully explore pairwise relation within a mini-batch. Other than pairwise ones, losses using proxy [16,38,21] are also proposed. Those methods assign single or multiple proxy vectors for each class, and distances are calculated with the proxies rather than individual embeddings. For our experiments, we adopt a simple pairwise distance-based loss: Triplet loss.

Pair Sampling Sampling pairs within a mini-batch is crucial to ensure a stable convergence and a better performance for the pairwise distance-based losses. In [26], semi-hard negative examples are sampled rather than the hardest ones, which improves the performance of Triplet loss for very large face datasets. Wu *et al.*[31] propose the distance-weighted sampling considering the distribution of random points in a unit-hyper sphere. Recently, Wang *et al.*[30] cast a sampling problem to a pair weighting problem based on a gradient analysis. Other than the pair sampling within a mini-batch, Suh *et al.*[27] devised a batch sampling method which considers an inter-class relationship. For our experiments, we used the distance-weighted sampling as the pair sampling method.

Multiple Heads Similarly to the ensemble of totally separate models, multiple heads that share the backbone network for better embeddings is studied intensively in recent work. The main focus is to diversify the output of each head for robust embedding. For example, the importance of data examples is re-weighted differently for each head based on the gradients of other heads [18] or difficulty of the examples [35]. In [25], a dataset is divided into multiple subsets and each head is trained for different subsets. Kim *et al.*[11] applied gating attention mechanism to heads, and the gates are trained to minimize their spatial overlapping regions. In [10], each head receives different high-order moments of the feature. Although multiple heads approach can be adopted to further improve the performance of the deep metric learning model, it is beyond the scope of the proposed method; the models we used in the experiments are single-headed.

2.2 Knowledge Transfer and Mutual Learning

Knowledge Transfer Knowledge distillation [7,22,36,8,14] aims to transfer the knowledge from a teacher to a student model. The typical setup is to train a student model to mimic the outputs of a teacher model. Traditional knowledge distillation [7] uses class probability as the knowledge to transfer. However, the encoding models of embedding learning do not produce a certain class probability. Therefore in deep metric learning, DarkRank [2] transfers similarity ranks between embeddings, and concurrently, relational knowledge distillation [19] and compact networks [34] proposed direct transfer of pairwise distance between embeddings. Multi-type knowledge [14] transfers graph structure lies in instance features and applied the method to the classification task. We also set a pairwise relation matrix of embeddings within a mini-batch, as the knowledge to transfer.
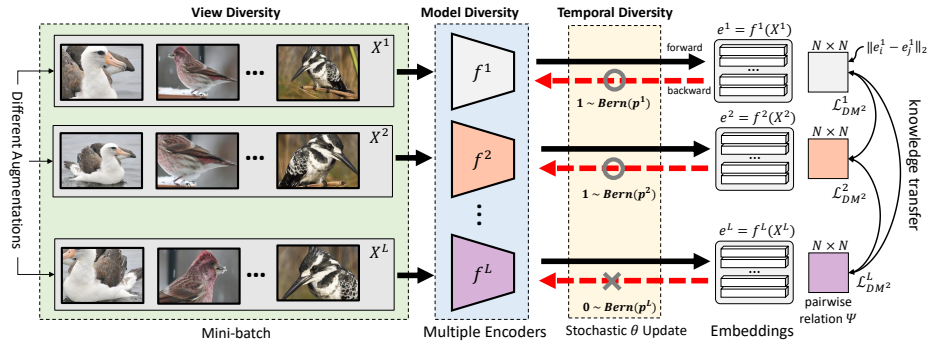
Fig. 1: Given a mini-batch, DM^2 applies view diversity that exploits different augmentations for each encoder which are simultaneously trained. To differentiate the frequency of parameter update among the cohort, DM^2 applies temporal diversity that stochastically updates the parameter of models according to their update probability. The relational knowledge of each model is mutually shared between the models by transferring their pairwise distance of embedding vectors.

Mutual Learning The learning procedure of knowledge distillation requires a fully-trained teacher model. Therefore, practitioners should endure tedious and time-consuming two-stage training pipeline: train a teacher model, and then distill the teacher to a student. To overcome this shortcoming, deep mutual learning [39] proposes a single stage training pipeline for knowledge distillation. The method minimizes the KL divergence between the class probability of multiple models on-the-fly in the training phase, and [1] scaled the method up to hundreds of GPUs by adopting distributed setup. In this online setting of knowledge distillation, there is no clear distinction between the teacher and student as in the classical setting of knowledge distillation, but the *diversity* among the cohort should be guaranteed to make the knowledge be transferred.

Although the effectiveness of mutual learning is proven for classification tasks, all these schematics have not been applied in other tasks. In this paper, DM^2 aims to provide adequate techniques to diversify the models of the cohort. The diversification techniques are particularly important in deep metric learning where the effect of original *model diversity* is not guaranteed due to the same initial weights of the pretrained models.

3   Diversified Mutual Metric Learning

In the framework of the **D**iversified **M**utual **M**etric Learning (DM^2), multiple models, *i.e.*, *cohort*, are involved in the training process and transfer their knowledge to each other. By diversifying the cohort, we have observed more diversity brings better generalization power for every single model and their ensemble. In this section, we will define our objective function and deliver detailed explanations of our three diversities.

Notation Our cohort is comprised of multiple diverse models f^i, where i is an index that goes from one to the size of the cohort L. The model, or the encoder, f^i maps an image x into an embedding vector $e^i \in \mathbb{R}^d$, *i.e.*, $e^i = f^i(x)$; for deep metric learning, the typical choice of d is 128 or 512 [30,31]. Then the distance between two images x_1 and x_2 can be measured via Euclidean distance of their respective embeddings: $\left\| f^i(x_1) - f^i(x_2) \right\|_2 = \left\| e^i_1 - e^i_2 \right\|_2$. Deep metric learning loss optimizes the encoders such that the distance between semantically similar images to be close and otherwise far apart. Note that the architecture of the models can either be homogeneous or heterogeneous and its parameters can be diverse or identical, which we will treat as *model diversity* in later section 3.2.

3.1 Objective Function

We set the knowledge to be shared among the cohort to pairwise relation matrix of embeddings that each model produces for a given same mini-batch. More concretely, for embeddings $\{e^i_k\}_{k=1}^N$ of model f^i, where N is the size of mini-batch; we define relation matrix of f^i as $\Psi^i_{k,l} = \left\| e^i_k - e^i_l \right\|_2$. The knowledge sharing from f_k to f_l is defined as a difference between the relation matrix Ψ as follows:

$$\mathcal{L}^{l \leftarrow k}_{\mathrm{DM}^2} = \frac{1}{N^2} \sum_i^N \sum_j^N \left\| \Psi^l_{i,j} - \Psi^k_{i,j} \right\|_2^2. \tag{1}$$

For each model f^l, the knowledge is transferred to all other models in the cohort $\{f^k\}_{k \neq l}$, defining the $\mathcal{L}^l_{\mathrm{DM}^2}$ as follows:

$$\mathcal{L}^l_{\mathrm{DM}^2} = \frac{1}{L-1} \sum_{k \neq l}^L \mathcal{L}^{l \leftarrow k}_{\mathrm{DM}^2}. \tag{2}$$

The final objective $\mathcal{L}^l$ for the model f^l in our diversified mutual metric learning is the sum of the deep metric learning loss $\mathcal{L}^l_{\mathrm{DML}}$ and $\mathcal{L}^l_{\mathrm{DM}^2}$ using Lagrangian multiplier λ_{DM^2}:

$$\mathcal{L}^l = \mathcal{L}^l_{\mathrm{DML}} + \lambda_{\mathrm{DM}^2} \cdot \mathcal{L}^l_{\mathrm{DM}^2}. \tag{3}$$

Throughout the experiments, we mostly used Triplet loss [26] for all $\{\mathcal{L}^l_{\mathrm{DML}}\}_{l=1}^L$. We applied a linear warm-up strategy for λ_{DM^2}, which makes each encoder to focus more on the metric learning loss during the early phase of the training.

Algorithm 1 Training procedure of DM^2

Require: L models $\{f^l\}_{l=1}^L$ with parameters $\{\theta^l\}_{l=1}^L$ ▶ *Model Diversity* (MD)
1: **for** the number of iterations **do**
2:      Sample a mini-batch $\{x_k\}_{k=1}^N$ of size N
3:      **for** $l = 1{:}L$ **do**   ▶ Done in parallel
4:         $\{A_k\}_{k=1}^N \sim \mathcal{A}$ ▶ Sample the augmentations
5:         $\{x_k'\}_{k=1}^N = \{A_k(x_k)\}_{k=1}^N$ ▶ *View Diversity* (VD)
6:         $\{e_k^l\}_{k=1}^N = \{f^l(x_k')\}_{k=1}^N$ ▶ Infer embeddings
7:      **end for**
8:      (Barrier) Wait until the inference of entire cohort finished
9:      **for** $l = 1{:}L$ **do**   ▶ Done in parallel
10:        Compute relation matrix Ψ^l from $\{e_k^l\}_{k=1}^N$ for f^l
11:        Compute $\mathcal{L}_{\mathrm{DML}}^l$ using $\{e_k^l\}_{k=1}^N$ for f^l
12:        Compute $\mathcal{L}_{\mathrm{DM}^2}^l$ using Ψ^l and $\{\Psi^k\}_{k\neq l}$ for f^l
13:        $\mathcal{L}^l = \mathcal{L}_{\mathrm{DML}}^l + \lambda_{\mathrm{DM}^2} \cdot \mathcal{L}_{\mathrm{DM}^2}^l$
14:        $p \sim Bern(p^l)$ ▶ Sample p from a Bernoulli distribution
15:        **if** p is 1 **then**   ▶ *Temporal Diversity* (TD)
16:          Update θ^l w.r.t $\mathcal{L}^l$
17:        **end if**
18:      **end for**
19: **end for**

3.2 Diversities

We now deliver detailed explanations of three diversities that could make the cohort to share more rich knowledge for mutual learning. The combination of the three diversities aims to diversify the update paths of multiple models in the parameter space, and the diversified paths lead to different local optima for each model. Due to the diversities, DM^2 can form a cohort of diverse models. We concretely depict the general framework of DM^2 in algorithm 1.

Model Diversity The first is a *model diversity* (MD), which diversifies the cohort by leveraging different initializations of models to encourage the individual models to explore a different region and settle at different local optima after training. MD is naturally induced by the learning procedure of mutual learning [39] where multiple models mutually teach each other. However, this diversity is hard to be achieved in the setting of deep metric learning because models are initialized with the same pretrained parameters, which is essential in inductive transfer learning, do not allow much diverse models, and even forcing them to be identical.

Temporal Diversity The second is a *temporal diversity* (TD), which diversifies the cohort by discriminating their frequency of parameter update. To discriminate the frequency of parameter update, we let the models stochastically update their parameters according to their update probabilities. We assign diverse update probabilities $\{p^l = 2^{-(l-1)}\}_{l=1}^L$ to each model according to its index. The

first model has the highest update probability which is one, and the last model has the lowest update probability. When the update probability is high, the model moves fast along the update path and it results in a more fitted model on the training dataset. On the contrary, when the update probability is low, the model results in less fitted on the dataset. Combining the models with different temporal steps in update helps to form a more diversified cohort.

View Diversity The third is a *view diversity* (VD), which diversifies the cohort by exploiting diverse inputs with different augmentations for each model. Since the gradients of each model are computed using input data and activations, different augmentations to the mini-batch of each model yield diversified gradients. In addition, VD helps to learn a more robust embedding model, as the models learn to construct the relational structure Ψ that is invariant to noise-injected data.

We defer the implementation details regarding the parameter initialization of the cohort ($\{\theta^l\}_{l=1}^L$) and the augmentation family $\mathcal{A}$ to section 4.2.

3.3   Distributed parallel learning

Computing $\mathcal{L}_{\text{DM}^2}^l$ requires only the relation matrices from the other encoders $\{\Psi^k\}_{k \neq l}$. Thus the computation including generating mini-batches with different views, inferring the encoder, and calculating the gradients can be done in parallel by allocating encoders to separate computational resources, *e.g.*, GPU. As a result, the training time of DM^2 almost stayed the same compared to that of a single model at the cost of computational resources.

4   Experiment

In this section, we report the efficacy of the DM^2, and the implementation details used in our experiments. We conduct experiments on the following standard datasets for metric learning, and follow the evaluation protocol, and train/test splits as proposed in [17].

4.1   Dataset

We run experiments on four datasets: CUB-200-2011 [29], Cars-196 [13], Stanford Online Products (SOP) [17] and In-Shop Clothes Retrieval (In-Shop) [15].

CUB-200-2011 consists of images of 200 bird species. Half of 200 classes (5,864 images) are used for the training and the remaining 100 classes (5,924 images) are used for the testing. Cars-196 consists of images of 196 car models. As in the CUB-200-2011, half of 196 classes (8,052 images) are used for the training and the remaining 98 classes (8,131 images) are used for the testing.

SOP, which consists of images containing 22,634 product classes, is a much larger dataset compared to the former two datasets. For the training, 11,318 classes (59,551 images) are used and the remaining 11,316 classes (60,499 images) are used for the testing. In-Shop consists of images of 11,735 clothes products. For the training, 3,997 classes (25,882 images) are used, and the remaining 7970 classes (26,830 images) are split into two subsets (query set and gallery set) for the testing.

4.2   Implementation Detail

Following the standard practice of deep metric learning studies, we use the inception network with the batch normalization (BN-Inception) [9] and the ResNet-50 [6] pretrained on ILSVRC 2012-CLS (ImageNet) [24] as our backbone networks. We note again that these pretrained backbones make the effect of model diversity weaker. However at the same time, the pretraining is strongly required for deep metric learning since the size of the datasets is very small to learn a visual feature extraction in an end-to-end manner, which makes the use of ImageNet pretraining essential.

We add a linear projection layer after the last pooling layer of a backbone network and apply l_2 normalization on the output embedding e of the linear projection layer, $e.g.$, $\frac{e}{\|e\|_2}$. We follow the image pre-processing and the augmentation family of a recent state-of-the-art method, HORDE [10], which uses 256×256 random crop from randomly resized images and apply a random horizontal flip.

We train the parameters of batch normalization layers [9] in backbone networks for SOP and In-Shop datasets and freeze them for CUB-200-2011 and Cars-196 datasets following the procedure done in [30,21].

Regardless of the model architecture and dataset choice, the training was done with Adam optimizer [12] with the initial learning rate of $3 \cdot 10^{-5}$ and weight decay of $5 \cdot 10^{-4}$; and the mini-batch size was 120. All experiments were conducted using pytorch 1.3.1 [20] with CUDA 10.2 on Nvidia V100 GPU.

To make enough positive pairs within a mini-batch, we follow the batch construction of FaceNet [26]. For every mini-batch, we randomly sample a certain number of classes and sample 5 images per class for small datasets (CUB-200-2011, Cars-196) and 2 images per class for large datasets (SOP, In-Shop). We apply a warm-up strategy for λ_{DM^2}: linearly increasing it from 0 to 20 for the first three epochs of the training.

Unless specified, we adopt Triplet loss [26] as our deep metric learning loss $\mathcal{L}_{\mathrm{DML}}$ and the used distance weighted sampling [31] as a pair sampling method. Note that we carefully tuned our hyper-parameters to build strong baseline models. The source codes will be released in public upon acceptance.

4.3   Efficacy of Diversities

The result of table 1 shows the effect of the diversities in the application of mutual learning for deep metric learning. We used a cohort of size 4 for these

Table 1: Recall@1 performance on CUB-200-2011 and Cars-196 datasets. We used cohort size of 4 for DM^2, and report how the performance increases with the added diversities: model diversity (MD), temporal diversity (TD), and view diversity (VD). We highlight the best performing single model and ensembled model.

Methods	CUB-200-2011					Cars-196				
	Net A	Net B	Net C	Net D	Ens	Net A	Net B	Net C	Net D	Ens
Independent	66.07	66.18	67.21	65.97	70.71	85.24	84.78	85.22	85.26	89.87
MD	67.93	67.12	67.18	67.32	69.60	85.79	86.07	86.31	86.29	87.86
MD+TD	69.45	69.01	68.10	67.59	71.25	89.00	88.49	87.54	86.21	90.49
MD+VD	68.26	68.72	68.42	68.45	71.69	87.91	87.22	87.39	87.49	90.10
MD+TD+VD	**69.90**	69.50	68.86	69.11	**72.89**	**89.10**	88.29	87.65	86.33	**91.48**

experiments, the size of the embedding vector is set to 512, and the networks are all BN-Inception architecture initialized with the same parameter pretrained on ImageNet. The only differences in the parameter are induced by the random initialization of the linear projection layer.

The first row of table 1 (independent) indicates the results of four independently trained networks (*i.e.*, $\lambda_{DM^2} = 0$) and their concatenation-based ensemble performance. The MD only results are identical to the naive adoption of mutual learning. As expected, the naive adoption brings worse result than the individual learning without construct cohort. On the contrary, one can see that the addition of our proposed novel diversities TD, VD brings superior results; and applying all diversities brings the state-of-the-art single model performance (see table 2).

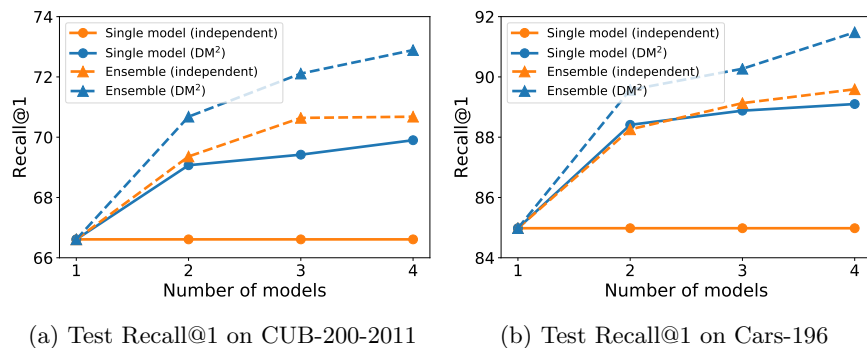(a) Test Recall@1 on CUB-200-2011   (b) Test Recall@1 on Cars-196

Fig. 2: Plots showing Recall@1 on the evaluation sets of CUB-200-2011 and Cars-196 with various cohort size (1, 2, 3, 4). For the single model results of DM^2, we report the result of the first model ($l=1$) among the multiple models of the cohort.
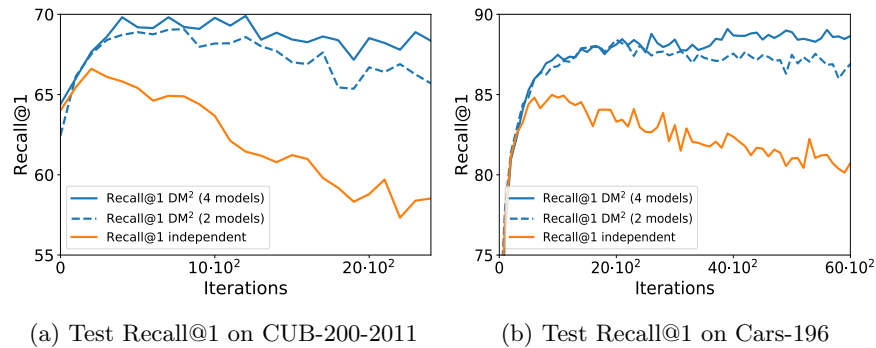
(a) Test Recall@1 on CUB-200-2011   (b) Test Recall@1 on Cars-196

Fig. 3: Plots showing the changes of Recall@1 on the evaluation sets of CUB-200-2011 and Cars-196 during the training. For DM^2, the Recall@1 results are from the first model (l=1) among the multiple models of the cohort.

4.4   Scalability of DM^2

Figure 2 delivers how the performance increases for larger cohort size. For both datasets, CUB-200-2011 and Cars-196, the mutual learning with all three diversities (*i.e.*, DM^2) results in a monotonic increment of Recall@1 along with the cohort size. On the contrary, the ensemble performance of individually trained models converges faster at the lower Recall@1 than that of DM^2.

Figure 3 depicts how mutual learning effects as a regularizer, which makes the model more robust. The plot shows individually trained models suffer a problem of severe overfitting on the latter part of the training. However, the model trained with DM^2 keeps outperforming the baseline without severe overfitting. Note that its robustness keeps increase for a larger cohort size.

4.5   Comparison with Other Deep Metric Learning Studies

Since the single model performance improved significantly with DM^2, we compared the recall performance of DM^2 with the current state-of-the-art deep metric learning methods. For fair comparison, the number of embedding dimension and the backbone network were set according to those of compared methods. The cohort size of DM^2 is fixed to 4.

The result of table 2 shows the recalls measured on CUB-200-2011, Cars-196, and SOP. For ResNet-50, the model of 512 embedding dimension trained with DM^2 and Triplet loss achieves the best Recall@1 for all three datasets: CUB-200-2011, Cars-196, and SOP. For BN-Inception, it achieved the best Recall@1 for CUB-200-2011, Cars-196, and the second best Recall@1 for SOP. For the results on In-shop dataset shown in table 3, our method achieves comparable performance with the best, less than 1% point difference.

Table 2: Recall@K comparison with state-of-the-art metric learning methods. We divide methods with same backbone network. "Dim." refers to the embedding dimension of the model. Boldface refers to the highest performance among same backbone. The numbers under the datasets refer to recall at K. The results of DM^2 are from a *single* model for fair comparison, which is the first model (l=1) among multiple models of cohort.

B	Methods	Dim.	CUB-200-2011				Cars-196				SOP			
			1	2	4	8	1	2	4	8	1	10	10^2	10^3
ResNet-50	Margin [31]	128	63.6	74.4	83.1	90.0	79.6	86.5	91.9	95.1	72.7	86.2	93.8	98.0
	DC [25]	128	65.9	76.6	84.4	**90.6**	84.6	90.7	94.1	96.5	75.9	88.4	94.9	98.1
	MIC [23]	128	66.1	76.8	**85.6**	-	82.6	89.1	93.2	-	77.2	89.4	95.6	-
	TML [33]	512	62.5	73.9	83.0	89.4	86.3	92.3	**95.4**	**97.3**	78.0	91.2	96.7	99.0
	Triplet + DM^2	128	64.1	75.0	83.2	89.4	84.7	89.9	93.5	95.8	79.9	91.3	96.6	99.0
	Triplet + DM^2	512	**66.7**	**77.1**	85.2	90.5	**86.9**	**92.0**	94.5	96.8	**80.4**	**91.8**	**96.8**	**99.1**
BN-Inception	HTL [4]	512	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7	74.8	88.3	94.8	98.4
	NSM [37]	512	59.6	72.0	81.2	88.4	81.7	88.9	93.4	96.0	73.8	88.1	95.0	-
	MS [30]	512	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0	98.7
	SoftTriplet [21]	512	65.4	76.5	84.5	90.4	84.5	90.7	94.5	96.9	78.3	90.3	95.9	-
	HORDE [10]	512	66.3	76.7	84.7	90.6	83.9	90.3	94.1	96.3	**80.1**	**91.3**	96.2	98.7
	Triplet + DM^2	512	**69.9**	**79.7**	**86.5**	**91.4**	**89.1**	**93.3**	**95.8**	**97.6**	78.8	90.9	**96.3**	**98.9**

Table 3: Recall@K comparison with state-of-the-art metric learning methods on In-Shop. Dim. refers to the embedding dimension of the model. Bold-face refers the highest performance among all models. The results of DM^2 are from a *single* model for fair comparison, which is the first model (l=1) among multiple models of cohort.

Methods	Dim.	In-Shop			
		1	10	20	30
FashionNet [15]	4096	53.0	73.0	76.0	77.0
NSM [37]	512	88.6	97.5	98.4	**98.8**
A-BIER [18]	512	83.1	95.1	96.9	97.5
ABE-8[11]	512	87.3	96.7	97.9	98.2
MS [30]	512	89.7	**97.9**	98.5	**98.8**
HORDE [10]	512	**90.4**	97.8	98.4	98.7
Triplet + DM^2	512	89.6	97.8	**98.6**	**98.8**

4.6 Combining DM^2 with Knowledge Distillation

We combine the ensembled model of DM^2 with *relational knowledge distillation* [19]. For a detailed comparison, we also report the results of other recently proposed knowledge distillation methods for deep metric learning [2,34]. To build a strong teacher model for the distillation methods, we build an ensemble of 4 independently trained models. For DM^2, the cohort of size 4 is mutually trained

Table 4: Recall@1 comparison between knowledge distillation methods for metric learning and DM^2. **P** indicates whether the teacher and student are trained in a parallel manner. For knowledge distillation methods, **T** and **S** refer to a teacher and a student respectively. For DM^2, **T** and **S** refer to an ensemble and a single model (the first model where $l=1$) respectively. '→' means that two methods were applied in a sequence.

| Methods | **P** | CUB-200-2011 | | | | Cars-196 | | | |
| | | 2×models | | 4×models | | 2×models | | 4×models | |
		T	**S**	**T**	**S**	**T**	**S**	**T**	**S**
Baseline	X	-	66.61	-	66.61	-	84.98	-	84.98
RKD-DA [19]	X	69.36	70.13	70.68	70.64	88.26	88.82	89.59	89.95
DarkRank [2]	X	69.36	68.28	70.68	68.09	88.26	87.92	89.59	87.99
Compact [34]	X	69.36	68.70	70.68	67.98	88.26	**89.50**	89.59	89.48
DM^2	O	**70.68**	69.07	**72.89**	69.90	**89.57**	88.41	**91.48**	89.10
DM^2 →RKD-DA	X	**70.68**	70.54	**72.89**	**71.22**	**89.57**	89.14	**91.48**	**90.21**

in parallel. Each model is based on BN-Inception with the embedding dimension of 512, and we transfer the knowledge of the ensemble to a single model (student). We apply the methods on the final representations, *i.e.*, embeddings. We describe each method briefly as follows:

- **Relational Knowledge Distillation** [19] transfers relative distance and angle formed by points in the embedding space of the model. Between the three combinations of loss they proposed, we used distance and angle combination (RKD-DA). Following the original paper, we did not apply $\mathcal{L}_{DML}$ while distillation and set the hyper-parameters $\lambda_{distance}$ and λ_{angle} to 20 and 40 respectively.
- **DarkRank** [2] transfers similarity ranks between examples. Between the two losses they proposed, we use HardRank loss. We carefully tune the hyper-parameters for DarkRnak using a grid search. The hyper-parameters α, β, and $\lambda_{DarkRank}$ are set to 3, 3, and 1, respectively.
- **Compact Networks** [34] transfers absolute distance among data examples. Following the original paper, the hyper-parameter $\lambda_{compact}$ is set to 10.

As shown in table 4, the result of pure DM^2 is on par with other two-staged distilled performances. By combining DM^2 with the RKD-DA, which showed the best distillation performance, the final single model performance reaches the highest Recall@1 71.22 and 90.21 for CUB-200-2011 and Cars-196, respectively. These results show that the ensemble model of DM^2 indeed holds better knowledge to transfer to the student (single model).

4.7   Heterogenous Cohort

For this experiment, we conducted experiments on heterogeneous cohort, which comprises models with diverse architecture. For that, we examined various con-

Table 5: Recall@1 comparison varying the combinations of heterogenous cohorts. BNI refers to BN-Inception and R50 refers to ResNet-50. $\times$ indicates the number of cohorts with the architecture. The results of the Single are from the first model (l=1) among multiple models of cohort. Baseline is a single model trained independently without mutual learning with cohorts.

Combinations	CUB-200-2011			Cars-196		
	BNI (Single)	R50 (Single)	Ens	BNI (Single)	R50 (Single)	Ens
Baseline	66.61	64.01	-	84.98	82.70	-
4×BNI	**69.90**	-	72.89	**89.10**	-	91.48
3×BNI 1×R50	68.69	66.63	**74.17**	87.13	**88.76**	**91.76**
2×BNI 2×R50	68.80	**66.90**	73.90	86.55	87.62	91.46
1×BNI 3×R50	68.08	66.44	71.59	86.41	86.20	90.21
4×R50	-	66.71	68.87	-	86.93	88.62

struction for a cohort of size 4 using BN-Inception and ResNet-50, and the embedding dimension was 512.

Table 5 shows the result of the experiment, and the row 4×BNI is identical to our previous experimental setting. We set the temporal diversity individually to the network type, for example, 3×BNI 1×R50 setting has the update rate of $(1, 0.5, 0.25, 1)$ for three BNI and one R50, respectively. The results show a feeble possibility of architecture diversity, and it is noteworthy that the ensemble performances of CUB-200-2011 and Cars-196 were found to be better for the heterogeneous cohort than the homogenous cohort. Also, one can see that the single model performance of R50 grows with the addition of BNI to its cohort.

4.8   DM^2 with Other $\mathcal{L}_{\mathrm{DML}}$

DM^2 can be effective with any deep metric learning objectives. To validate that, we conduct an experiment of combining DM^2 with widely adopted deep metric learning losses, *e.g.*, Triplet, Contrastive, Proxy-NCA, Binomial-Deviance, and ArcFace. The size of the cohort is set to 4, the models of BN-Inception with 512 embedding dimension are adopted. We describe the hyper-parameters of each method.

- **Triplet** [26]: we follow the hyper parameter specified in section 4.2.
- **Contrastive** [5]: we set the margin to 1.0 and λ_{DM^2} to 20.
- **Proxy-NCA** [16]: we set the number of the proxies to 1 and λ_{DM^2} to 10^2.
- **Binomial-Deviance** [32]: we set the alpha and beta to 2 and 0.5 respectively following [32], and λ_{DM^2} to 20.
- **ArcFace** [3]: we set the scale factor and margin to 64 and 0.5 respectively following [3], and λ_{DM^2} to $2 \cdot 10^3$.

The result in table 6 delivers that DM^2 improves the performance of a *single model* with any deep metric learning objectives.

Table 6: Applicability of DM^2 on different deep metric learning objectives. 'Independent' refers to the case when a model is trained independently with only $\mathcal{L}_{DML}$. For DM^2, the results of Single Model are from the first model (l=1) among multiple models of cohort.

(a) Recall@1 on CUB-200-2011.

$\mathcal{L}_{DML}$	Independent		DM^2	
	Single Model	Ensemble	Single Model	Ensemble
Contrastive [5]	61.97	63.94	66.40	67.45
Triplet [26]	66.07	70.71	69.90	72.89
Binomial-Deviance [32]	64.83	69.01	67.92	69.90
Proxy-NCA [16]	60.11	67.42	64.95	68.67
ArcFace [3]	62.09	69.05	67.74	70.32

(b) Recall@1 on Cars-196.

$\mathcal{L}_{DML}$	Independent		DM^2	
	Single Model	Ensemble	Single Model	Ensemble
Contrastive [5]	78.93	82.97	83.10	86.72
Triplet [26]	85.24	89.87	89.10	91.48
Binomial-Deviance [32]	83.94	89.16	88.33	89.20
Proxy-NCA [16]	83.70	90.94	86.74	89.01
ArcFace [3]	85.11	91.13	88.45	90.16

5   Conclusion

In this paper, we propose **D**iversified **M**utual **M**etric Learning (DM^2) to effectively apply mutual learning on the task of deep metric learning. In DM^2, three diversities are proposed to enrich the knowledge shared by the cohort of mutual learning: model diversity, temporal diversity, and view diversity. We proved that the temporal and view diversities were especially essential to the application of mutual learning since deep metric learning requires pretrained models due to its small size dataset. By combining all three diversities carefully, DM^2 results the state-of-the-art performance on widely adopted deep metric learning datasets: CUB-200-2011 and Cars-196, and SOP.

Acknowledgement

References

1. Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G.E., Hinton, G.E.: Large scale distributed neural network training through online distillation. arXiv preprint arXiv:1804.03235 (2018) 1, 2, 4

2. Chen, Y., Wang, N., Zhang, Z.: Darkrank: Accelerating deep metric learning via cross sample similarities transfer. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018) 3, 11, 12

3. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019) 13, 14

4. Ge, W.: Deep metric learning with hierarchical triplet loss. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 269–285 (2018) 11

5. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). vol. 2, pp. 1735–1742. IEEE (2006) 2, 13, 14

6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2, 8

7. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 3

8. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017) 3

9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015) 2, 8

10. Jacob, P., Picard, D., Histace, A., Klein, E.: Metric learning with horde: High-order regularizer for deep embeddings. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2019) 3, 8, 11

11. Kim, W., Goyal, B., Chawla, K., Lee, J., Kwon, K.: Attention-based ensemble for deep metric learning. In: The European Conference on Computer Vision (ECCV) (2018) 3, 11

12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 8

13. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13) (2013) 2, 7

14. Liu, Y., Cao, J., Li, B., Yuan, C., Hu, W., Li, Y., Duan, Y.: Knowledge distillation via instance relationship graph. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7096–7104 (2019) 3

15. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1096–1104 (2016) 7, 11

16. Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. CoRR abs/1703.07464 (2017), http://arxiv.org/abs/1703.07464 3, 13, 14

17. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2, 3, 7

18. Opitz, M., Waltner, G., Possegger, H., Bischof, H.: Deep metric learning with bier: Boosting independent embeddings robustly. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018). https://doi.org/10.1109/TPAMI.2018.2848925 3, 11
19. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3967–3976 (2019) 3, 11, 12
20. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017) 8
21. Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., Jin, R.: Softtriple loss: Deep metric learning without triplet sampling. arXiv preprint arXiv:1909.05235 (2019) 3, 8, 11
22. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. International Conference on Learning Representations (2015) 3
23. Roth, K., Brattoli, B., Ommer, B.: Mic: Mining interclass characteristics for improved metric learning. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019) 11
24. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y 8
25. Sanakoyeu, A., Tschernezki, V., Buchler, U., Ommer, B.: Divide and conquer the embedding space for metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 471–480 (2019) 3, 11
26. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 2, 3, 5, 8, 13, 14
27. Suh, Y., Han, B., Kim, W., Lee, K.M.: Stochastic class-based hard example mining for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7251–7259 (2019) 3
28. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss. In: Advances in Neural Information Processing Systems. pp. 4170–4178 (2016) 3
29. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011) 2, 7
30. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5022–5030 (2019) 3, 5, 8, 11
31. Wu, C.Y., Manmatha, R., Smola, A.J., Krahenbuhl, P.: Sampling matters in deep embedding learning. In: The IEEE International Conference on Computer Vision (ICCV) (2017) 3, 5, 8, 11
32. Yi, D., Lei, Z., Li, S.: Deep metric learning for practical person re-identification. ArXiv e-prints (2014) 3, 13, 14
33. Yu, B., Tao, D.: Deep metric learning with tuplet margin loss. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019) 11
34. Yu, L., Yazici, V.O., Liu, X., Weijer, J.v.d., Cheng, Y., Ramisa, A.: Learning metrics from teachers: Compact networks for image embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2907–2916 (2019) 3, 11, 12

35. Yuan, Y., Yang, K., Zhang, C.: Hard-aware deeply cascaded embedding. In: Proceedings of the IEEE international conference on computer vision. pp. 814–823 (2017) 3
36. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. International Conference on Learning Representations (2017) 3
37. Zhai, A., Wu, H.Y.: Classification is a strong baseline for deep metric learning. arXiv preprint arXiv:1811.12649 (2018) 11
38. Zhai, A., Wu, H.Y., San Francisco, U.: Classification is a strong baseline for deep metric learning (2019) 3
39. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4320–4328 (2018) 1, 2, 4, 6