Deep Factorized Metric Learning

Chengkun Wang^{1,2,*} Wenzhao Zheng^{1,2,*} Junlong Li^{1,2} Jie Zhou^{1,2} Jiwen Lu^{1,2,†}
^{1}Department of Automation, Tsinghua University, China
^{2}Beijing National Research Center for Information Science and Technology, China
{wck20,zhengwz18,junlong-20}@mails.tsinghua.edu.cn; {jzhou,lujiwen}@tsinghua.edu.cn

Abstract

Learning a generalizable and comprehensive similarity metric to depict the semantic discrepancies between images is the foundation of many computer vision tasks. While existing methods approach this goal by learning an ensemble of embeddings with diverse objectives, the backbone network still receives a mix of all the training signals. Differently, we propose a deep factorized metric learning (DFML) method to factorize the training signal and employ different samples to train various components of the backbone network. We factorize the network to different sub-blocks and devise a learnable router to adaptively allocate the training samples to each sub-block with the objective to capture the most information. The metric model trained by DFML capture different characteristics with different sub-blocks and constitutes a generalizable metric when using all the sub-blocks. The proposed DFML achieves state-of-the-art performance on all three benchmarks for deep metric learning including CUB-200-2011, Cars196, and Stanford Online Products. We also generalize DFML to the image classification task on ImageNet-1K and observe consistent improvement in accuracy/computation trade-off. Specifically, we improve the performance of ViT-B on ImageNet (+0.2% accuracy) with less computation load (-24% FLOPs). [1]

1. Introduction

Learning good representations for images has always been the core of computer vision, yet measuring the similarity between representations after obtaining them is an equally important problem. Focusing on this, metric learning aims to learn a discriminative similarity metric under which the interclass distances are large and the intraclass distances are small. Using a properly learned similarity metric can improve the performance of downstream tasks and has been employed in many applications such as semantic instance segmentation [7, 21, 37], remote sensing [5, 10, 31], and room layout estimation [77].

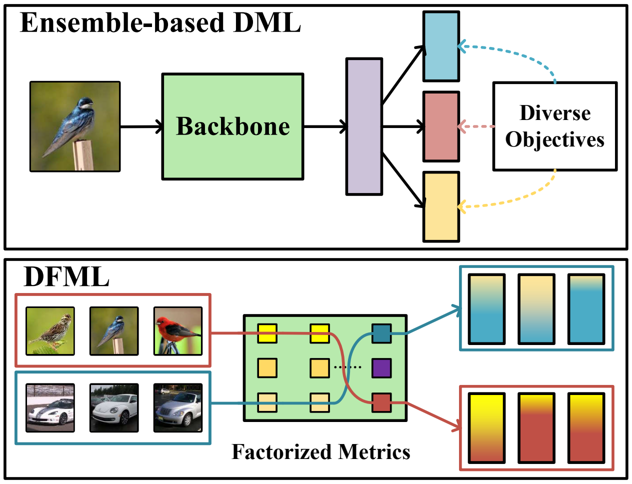

Figure 1. Comparisons between ensemble-based deep metric learning methods and DFML. Ensemble-based DML learns an ensemble of embeddings where diverse objectives are employed. Differently, DFML factorizes the backbone and learns a certain routine for each sample to achieve the diversity of features, which further boosts the generalization ability of the model on unseen classes. (Best viewed in color.)

Modern metric learning methods [44, 55, 56, 78] usually exploit deep neural networks to map an image to a single embedding and use the Euclidean distance or cosine similarity between embeddings to measure the similarity. As a single embedding might not be able to fully characterize an image, a number of methods [1, 43, 47, 49, 72, 79, 80] begin to explore using an ensemble of embeddings to represent an image, where each embedding describes one attribute of the image. The key to ensemble-based methods lies in how to enforce diversity in the ensemble of embeddings so that they can capture more characteristics. They achieve this by using a diversity loss [47, 49], selecting different samples [53, 72, 80], and adopting various tasks [43, 79], etc. Most existing methods adopt a shared backbone network to extract a common feature and only apply a single fully connected layer to obtain each specialized embedding.

[*] Equal contribution.
[†] Corresponding author.
[1] Code is available at: https://github.com/wangck20/DFML.

However, the shared backbone limits the diversity of the ensemble and hinders its ability to capture more generalizable features. It still receives a mix of all the training signals and can hardly produce diverse embeddings.

To address this, we propose a deep factorized metric learning (DFML) method to adaptively factorize the training signals to learn more generalizable features, as shown in 1. We first factorize each block of the metric backbone model to a number of sub-blocks, where we make the summed features of all the sub-blocks to be equal to that of the full block. As different samples may possess distinct characteristics [80], we devise a learnable router to adaptively allocate the training samples to the corresponding sub-blocks. We learn the router using a reconstruction objective to encourage each sample to be processed by the most consistent sub-block. We demonstrate the proposed DFML framework is compatible with existing deep metric learning methods with various loss functions and sampling strategies and can be readily applied to them. Due to the better modularity of vision transformers (ViTs) [15,61], we mainly focus on factorizing ViTs and further benchmark various existing deep metric learning methods on ViTs. Extensive experiments on the widely used CUB-200-2011 [63], Cars196 [35], and Stanford Online Products [56] datasets show consistent improvements of DFML over existing methods. We also provide an in-depth analysis of the proposed DFML framework to verify its effectiveness. Specifically, we show that backbone models trained by our DFML achieve better accuracy/computation trade-off than the original model on ImageNet-1K [52] and even improve the performance of ViT-B (+0.2% accuracy) with less computation load (-24% FLOPs).

2. Related Work

Deep Metric Learning: Deep metric learning methods focus on mapping an image to an effective embedding space, in which we can measure the semantic distances among samples. To achieve this, various methods devise discriminative losses on the image embeddings and aim at enlarging the interclass Euclidean distance while reducing the intraclass Euclidean distance [4, 23, 55, 56, 59, 60, 65, 67, 73]. For example, the widely used triplet loss [9, 54, 64] imposes a constraint within a triplet that the distance between a negative pair should be larger than that between a positive pair according to a margin. Proxy-based methods [33, 44, 51, 60] have attracted increasing attention in recent years. Roth *et al.* proposed NIR to leverage Normalizing Flows and enforce unique translatability of samples around respective proxies. Instead of random sampling tuples in the training data, hard mining strategies to select false positive tuples have been proven helpful for effective training [16,17,22,26,29,54,75]. For example, mining hard but discriminative negative samples improves the performance of the triplet loss and boosts the convergence speed [26,29,54,75]. Additionally, a variety of methods explores other sampling strategies from different perspectives to improve the training process [22,39,41,44,50,58,76].

Ensemble Learning: Conventional deep metric learning methods focus on producing discriminative representation resulting in poor generalizability. Since ensemble learning is widely used in machine learning tasks, such as reinforcement learning [3, 36, 69] and unsupervised learning [19,27,62], it is known that an elaborate combination of several weak learners often performs better generalization compared with the best single learner [25]. Additionally, Breiman [2] advances that the diversity of the intermediate outcomes in the ensemble is critical to the effectiveness of the ensemble. Hence, recently proposed ensemble learning methods combines diverse embeddings from several relatively weak learners as the final representation for similarity measure [34,43,46,53]. To encourage diversity, existing art instantiates different learners by combining multi-level features from different layers [75], adaptively selecting samples for different learners [53, 72, 80], or applying multiple attention masks for different learners [34]. For example, Zheng *et al.* [79] employed several learnable compositors for different combination strategies and employed a self-reinforced loss to enhance the compositor diversity.

Vision Transformers: After the transformer architecture has shown great success in the natural language processing field, Dosovitskiy *et al.* [15] first revealed that a pure vision transformer (ViT) architecture can attain comparable performance with state-of-the-art convolutional neural networks (CNNs). Using alternating self-attention layer and MLP layer instead of convolutions endows ViTs with less inductive bias and more representation capacity. The following improved variants [11, 24, 30, 38, 40, 42, 61, 66, 74] of vision transformer demonstrated even better accuracy/FLOPs than the counterpart CNNs, further pushing the performance limit on several core computer vision tasks such as image classification [11, 40, 61], object detection [6, 12, 13, 81], and semantic segmentation [8, 57]. El-Nouby *et al.* [18] first introduced the transformer architecture to the image retrieval task and achieved state-of-the-art results. However, they simply used the original ViT to replace CNN as the backbone feature extractor. Differently, we take advantage of the modularity of ViTs to perform factorization for better generalization of the learned metric.

3. Proposed Approach

In this section, we first formulate the problem of existing ensemble-based deep metric learning approaches and define the factorization learning form. Then, we present our metric factorization structure and the learnable router for factorization learning. Lastly, we elaborate on the deep factorized metric learning framework.

3.1. Ensemble v.s. Factorization

Given an image set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and the corresponding label set $\mathbf{L} = \{l_1, l_2, ..., l_N\}$, deep metric learning extracts features from each image $\mathbf{x}_i$ and obtains a transformed embedding, denoted as $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$. The widely adopted Euclidean distance metric is defined upon pairwise samples, illustrated as follows:

$$D(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{y}_i - \mathbf{y}_j||_2 = ||\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_j)||_2, \quad (1)$$

where $||\cdot||_2$ is the L2 norm.

Generally, conventional deep metric learning methods aim at enlarging distances between images from different classes while pulling close positive samples, which ensures the discrimination among classes:

$$J_{dis}(\mathbf{Y}) = w_p I(l_i, l_j) D(\mathbf{x}_i, \mathbf{x}_j) - w_n(1 - I(l_i, l_j)) D(\mathbf{x}_i, \mathbf{x}_j), \quad (2)$$

where w_p and w_n are positive coefficients, $I(l_i, l_j)$ outputs 1 if $l_i = l_j$ and 0 otherwise.

Nevertheless, this strategy ignores the intraclass variations that images from the same class might encode similar characteristics, such as color and illumination. Furthermore, these characteristics might benefit the generalization on unseen classes thus boosting testing performances. Under such circumstances, ensemble-based deep metric learning approaches have been proposed to train an ensemble of embeddings that encode diverse characteristics of each image $\mathbf{x}_i$, denoted as $\{\mathbf{y}_i^1 = \mathbf{h}_1(\mathbf{f}(\mathbf{x}_i)), ..., \mathbf{y}_i^M = \mathbf{h}_M(\mathbf{f}(\mathbf{x}_i))\}$. They employ different objectives on these embeddings in addition to the overall objective on the complete embedding, which can be formulated as follows:

$$J_e(\mathbf{Y}) = J_{dis}(\mathbf{Y}) + \sum_{m=1}^{M} \lambda_m J_m(\mathbf{Y}^m), \quad (3)$$

where J_{dis} is the conventional deep metric learning loss term, J_m denotes the mth objective, and λ_m balances different losses.

Although existing ensemble-based methods employ diverse objectives on these embeddings, the backbone network $\mathbf{f}(\cdot)$ receives mixed training signals as follows:

$$\frac{\partial J_e(\mathbf{Y})}{\partial \theta_f} = \frac{\partial J_{dis}(\mathbf{Y})}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \theta_f} + \sum_{m=1}^{M} \lambda_m \frac{\partial J_m(\mathbf{Y}^m)}{\partial \mathbf{Y}^m} \cdot \frac{\partial \mathbf{Y}^m}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \theta_f}, \quad (4)$$

where θ_f denotes the parameters of the backbone network. Therefore, the backbone network learns to extract mixed characteristics from the input images, which still limits the generalization ability of the model.

Differently, we directly factorize the backbone network as $\mathbf{F} = \{\mathbf{f}_1(\cdot), \mathbf{f}_2(\cdot), ...\}$ and each factorized network $\mathbf{f}_i(\cdot)$ comprises several components of $\mathbf{f}(\cdot)$. A non-linear mapping $\mathbf{G} : \mathcal{X} \rightarrow \mathcal{F}$ from the image space $\mathcal{X}$ to the factorized network space $\mathcal{F}$ determines which factorized network an input image $\mathbf{x}_i$ passes through. Therefore, the corresponding factorized embedding of $\mathbf{x}_i$ is denoted as $\mathbf{y}_i^f = \mathbf{G}[\mathbf{x}_i](\mathbf{x}_i)$ on which the metric learning objective is employed. Under such circumstances, we factorize the training signal by employing different samples to train distinguished components of $\mathbf{f}(\cdot)$ to extract more diverse characteristics, which further boosts the generalization ability of the learned model on unseen classes compared with the ensemble-based approaches.

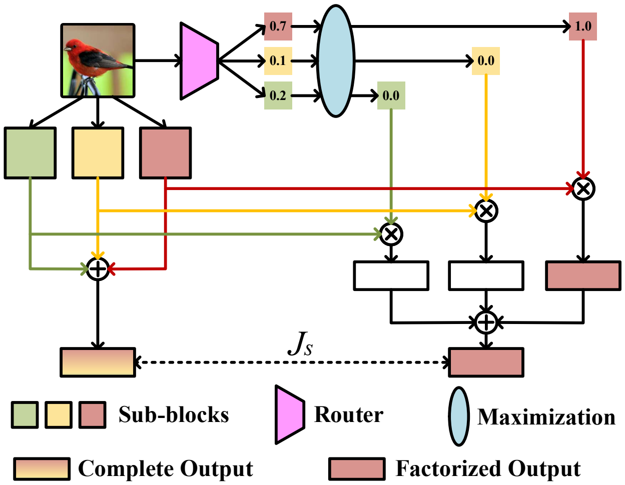

Figure 2. The learning process of DFML. We minimize the L2 norm of the complete output and the factorized output to optimize the parameters of each router. (Best viewed in color.)

3.2. Metric Factorization

Metric factorization exploits different components of the backbone network to obtain the corresponding embedding for each sample, which is conducted by designing the factorization pattern of the network and the mapping form of $\mathbf{G}$. Specifically, supposing that the backbone network $\mathbf{f}(\cdot)$ is composed of multiple blocks, denoted as $\{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_T\}$, we formulate the process of metric factorization for the ith block as $\mathbf{b}_i = \{\mathbf{b}_i^1, \mathbf{b}_i^2, ..., \mathbf{b}_i^K\}$. We restrict that each sub-block of $\mathbf{b}_i$ has the same structure. Additionally, the summation of the outputs of $\{\mathbf{b}_i^j\}_{j=1}^K$ equals the output of $\mathbf{b}_i$, which is formulated as follows:

$$\mathbf{b}_i(\mathbf{z}_{i,n}) = \sum_{j=1}^{K} \mathbf{b}_i^j(\mathbf{z}_{i,n}), \quad (5)$$

where $\mathbf{z}_{i,n}$ denotes the input for the ith block of $\mathbf{x}_n$. (5) ensures the consistency of our proposed metric factorization that we can still handle conventional deep metric learning problems with our factorized network.

To encode more diverse characteristics from images, we force $\mathbf{G}$ to choose one from $\{\mathbf{b}_i^j\}_{j=1}^K$ in the ith block for

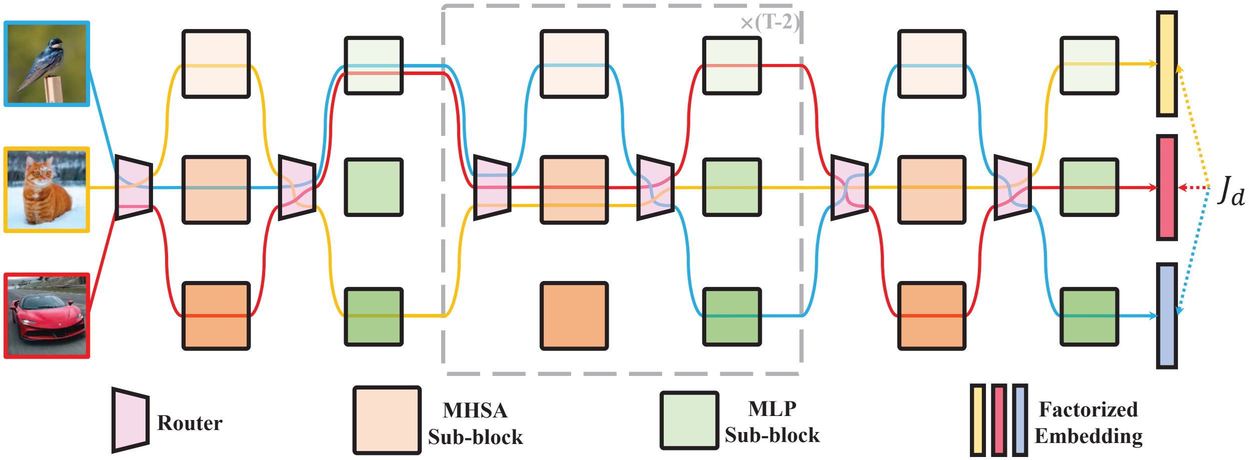

Figure 3. The overall framework of the proposed DFML method. We factorize the MHSA and MLP blocks into multiple sub-blocks. Each training sample will be passed through diverse sub-blocks chosen by the routers. The routines are distinguished with different colors. We impose the discriminative deep metric learning loss on the factorized embeddings. (Best viewed in color.)

each input sample individually. To be specific, we assume that $\mathbf{G}$ is a composition of K routers denoted as $\mathbf{G} = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_K\}$. Each router $\mathbf{g}_i$ maps the input feature $\mathbf{z}_i$ to a certain sub-block in $\{\mathbf{b}_i^j\}_{j=1}^K$, denoted as $\mathbf{g}_i : \mathcal{Z} \to \mathcal{B}_i$. Then $\mathbf{z}_{i,n}$ will pass through the chosen sub-block and serve as the input for the $(i+1)$th block, which is shown as follows:

$$\mathbf{b}_i^j = \mathbf{g}_i(\mathbf{z}_{i,n}), \quad \mathbf{z}_{i+1,n} = \mathbf{b}_i^j(\mathbf{z}_{i,n}), \tag{6}$$

where $i = \{1, 2, ..., T\}$ ($\mathbf{z}_{i+1,n}$ denotes the final output when $i = T$) and $j = \{1, 2, ..., K\}$.

Generally, the proposed metric factorization strategy is capable of encoding K different characteristics from the input samples in each block. Therefore, the whole network possesses K^T distinguished routines for images resulting in relatively diverse feature encoding. In contrast, existing ensemble-based deep metric learning approaches merely encode no more than M characteristics and $M \ll K^T$, which demonstrates that the proposed metric factorization strategy extracts more diverse information thus boosting the generalization ability on unseen classes.

3.3. Learning to Factorize

We consider that the learning to factorize process needs to guarantee both discrimination and significance of each factorized embedding $\mathbf{y}_i^f$. Specifically, we hope that the learned factorized embeddings satisfy the original deep metric learning objective to be discriminative among classes that positive pairs are pulled together while negative samples are pushed away. Therefore, we directly employ the discriminative metric learning loss on these embeddings $\mathbf{Y}^f$, which can be formulated as follows:

$$J_d(\mathbf{Y}) = J_{dis}(\mathbf{Y}^f), \tag{7}$$

where J_{dis} is the conventional deep metric learning loss term similar to (3).

In addition, we deem that the factorized embeddings should encode the most significant information from the input images to take full advantage of the data. To achieve this, we propose to devise learnable routers $\{\mathbf{g}_i\}_{i=1}^K$ and optimize these routers to select the dominant sub-blocks for training samples. Each router $\mathbf{g}_i$ contains a fully connected layer with the softmax activation function employed on the input feature $\mathbf{z}_{i,n}$ to obtain the weighting score:

$$c_i^k(\mathbf{z}_{i,n}) = \frac{exp(\mathbf{w}_i^{k^T}\mathbf{z}_{i,n} + b_i^k)}{\sum_{j=1}^K exp(\mathbf{w}_i^{j^T}\mathbf{z}_{i,n} + b_i^j)}, \tag{8}$$

where $\mathbf{w}_i$ and b_i are the learnable parameters of the fully connected layer, c_i^k denotes the kth component of the weighting score.

The weighting score demonstrates the contribution of each sub-block for $\mathbf{z}_{i,n}$. We force the k value corresponding to the maximum $\{c_i^k\}_{k=1}^K$ to be the chosen sub-block that extracts features from $\mathbf{z}_{i,n}$, formulated as follows:

$$j = \arg\max_k \; c_i^k(\mathbf{z}_{i,n}), \quad \mathbf{z}_{i+1,n} = \mathbf{b}_i^j(\mathbf{z}_{i,n}), \tag{9}$$

After that, we calculate the original output of $\mathbf{b}_i$, which we define as the complete output of the ith block, denoted as $\mathbf{z}_{i+1,n}^c = \mathbf{b}_i(\mathbf{z}_{i,n})$ and we employ a reconstruction objective by minimizing the L2 norm of two outputs as the significance loss:

$$J_s^{i,n}(\mathbf{Y}) = ||\mathbf{z}_{i+1,n} - \mathbf{z}_{i+1,n}^c||_2, \tag{10}$$

where $J_s^{i,n}$ denotes the loss term corresponding to the ith block and the nth input image, which forces the router $\mathbf{g}_i$ to choose the sub-block that extracts the characteristics closest to the complete ones. Note that the significance loss only optimizes the parameters of the learnable routers and thus we detach the features before they are processed by each router. The learning process is illustrated in Figure 2.

Generally, the final objective of our proposed factorization learning is as follows:

$$J_f(\mathbf{Y}) = J_d(\mathbf{Y}) + \lambda_s \frac{1}{N \times T} \sum_{n=1}^{N} \sum_{i=1}^{T} J_s^{i,n}(\mathbf{Y}), \quad (11)$$

where λ_s balances the effect of the significance loss.

3.4. Deep Factorized Metric Learning

We present the formulation of our DFML framework, as illustrated in Figure 3. We adopt modern vision transformers (ViTs) [15] as our backbone network, which are composed of multiple spatial attention blocks and channel processing blocks. Specifically, the attention blocks run H self-attention (SA) operations and concatenate the outputs for projection: (We ignore the Layernorm for simplicity.)

$$MHSA(\mathbf{z}) = [SA_1(\mathbf{z}); SA_2(\mathbf{z}); ...; SA_H(\mathbf{z})]\mathbf{Q}, \quad (12)$$

where $\mathbf{z} \in \mathcal{R}^{W \times D}$ denotes the input feature, W is the token number for per image, D is the channel dimension, $MHSA$ denotes the multi-head self-attention operation, H is the head number, and $\mathbf{Q} \in \mathcal{R}^{D \times D}$ is a projection matrix.

We conduct the metric factorization for $MHSA$ by dividing the head into K copies and adjusting the dimension of the projection matrix. Each sample will pass through a chosen $MHSA$ sub-block following (9). We formulate the ith $MHSA$ operation as follows: $(i = 1, 2, ..., K)$

$$MHSA_i(\mathbf{z}) = [SA_{(i-1)\cdot\frac{H}{K}+1}(\mathbf{z}); ...; SA_{i\cdot\frac{H}{K}}(\mathbf{z})]\mathbf{Q}_i, \quad (13)$$

where $\mathbf{Q}_i \in \mathcal{R}^{D/K \times D}$ projects the features from the reduced dimension $\frac{D}{K}$ to the original dimension D. Specially, we force $MHSA(\mathbf{z}) = \sum_{i=1}^{K} MHSA_i(\mathbf{z})$ following (5).

A channel processing block in ViTs utilizes an MLP block with two fully connected layers $\{\mathbf{fc}^1, \mathbf{fc}^2\}$ that first projects the input feature $\mathbf{z}$ from D to $r \times D$ and then from $r \times D$ back to D, where r denotes the ratio of the hidden dimension: (We ignore the activation function for simplicity.)

$$MLP(\mathbf{z}) = \mathbf{fc}^2(\mathbf{fc}^1(\mathbf{z})). \quad (14)$$

Differently, we factorize the MLP block by constructing K sub-blocks with the hidden dimension of $\frac{r \times D}{K}$. The ith sub-block is formulated as follows: $(i = 1, 2, ..., K)$

$$MLP_i(\mathbf{z}) = \mathbf{fc}_i^2(\mathbf{fc}_i^1(\mathbf{z})), \quad (15)$$

where $\mathbf{fc}_i^1 \in \mathcal{R}^{D \times (rD/K)}$ and $\mathbf{fc}_i^2 \in \mathcal{R}^{(rD/K) \times D}$. Similarly, we ensure the consistency by restricting $MLP(\mathbf{z}) = \sum_{i=1}^{K} MLP_i(\mathbf{z})$.

In consequence, we set a series of routers in the spatial attention blocks and the channel processing blocks to choose a certain routine for each input image. The significance loss in (10) optimizes the parameters of these routers to focus on the most important sub-blocks. Note that we employ a discriminative objective on the complete output of the backbone network to further maintain the discrimination of the learned metric. The overall objective of DFML can be formulated as follows:

$$J(\mathbf{Y}) = J_{dis}(\mathbf{Y}) + \lambda_f J_f(\mathbf{Y}), \quad (16)$$

where λ_f balances the effect between the discriminative loss and the factorization loss.

Our DFML framework is compatible with various loss functions. For example, we can instantiate J_{dis} with the widely adopted ProxyAnchor loss [33] as follows:

$$\begin{aligned} J_{dis}(\mathbf{Y}) = & \frac{1}{|\mathbf{P}^+|} \sum_{\mathbf{p} \in \mathbf{P}^+} log(1 + \sum_{l_i = l_\mathbf{p}} e^{-\alpha(s(\mathbf{y}_i, \mathbf{p}) - \delta)}) \\ & + \frac{1}{|\mathbf{P}|} \sum_{\mathbf{p} \in \mathbf{P}} log(1 + \sum_{l_i \neq l_\mathbf{p}} e^{\alpha(s(\mathbf{y}_i, \mathbf{p}) + \delta)}), \end{aligned} \quad (17)$$

where $\mathbf{P}$ is the set of all proxies, $\mathbf{P}^+$ denotes the set of the positive proxies, $|\cdot|$ outputs the size of the set, $s(\mathbf{y}_i, \mathbf{p})$ computes the cosine similarity between $\mathbf{y}_i$ and $\mathbf{p}$, α and δ are pre-defined hyper-parameters.

In addition, DFML can be applied to the image classification task by adding a classifier head to the final embedding to obtain the logits, denoted as $\mathbf{u}_i = CLS(\mathbf{y}_i)$. Then we utilize classification-based losses to optimize the training process. For instance, J_{dis} instantiated with the softmax loss is as follows:

$$J_{dis}(\mathbf{Y}) = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{\mathbf{u}_{i,l_i}}}{\sum_j e^{\mathbf{u}_{i,j}}}, \quad (18)$$

where $\mathbf{u}_{i,j}$ denotes the jth component of the logits.

4. Experiments

In this section, we evaluated DFML on the image retrieval task and generalizes to image classification as well.

4.1. Datasets

For image retrieval, we conducted various experiments on three widely adopted benchmark datasets: CUB-200-2011 [63], Cars196 [35], and Stanford Online Products [55]. The CUB-200-2011 dataset contains 11,788 images including 200 bird species. The first 100 classes with 5,864 images are used for training and the remaining 100 classes with 5,924 images are for testing. The Cars196 dataset comprises 196 car models of 16,185 images. The training set contains the first 98 classes with 8,054 images and the test set includes the rest 96 classes with 8,131 images. The Stanford Online Products dataset is composed of 22,634 products of 120,053 images. The first 11,318 products with 59,551 images are for training the rest 11,316

Table 1. Experimental results (%) on the CUB-200-2011, Cars196, and Stanford Online Products datasets compared with state-of-the-art methods. * denotes our reproduced results under the same settings.

		CUB-200-2011					Cars196					Stanford Online Products				
Method	Setting	R@1	R@2	NMI	M@R	RP	R@1	R@2	NMI	M@R	RP	R@1	R@10	NMI	M@R	RP
SoftTriple [48]	512BN	65.4	76.4	69.3	-	-	84.5	90.7	70.1	-	-	78.3	90.3	92.0	-	-
MIC [49]	128R	66.1	76.8	69.7	-	-	82.6	89.1	68.4	-	-	77.2	89.4	90.0	-	-
RankMI [32]	128R	66.7	77.2	71.3	-	-	83.3	89.8	69.4	-	-	74.3	87.9	90.5	-	-
CircleLoss [59]	512R	66.7	77.4	-	-	-	83.4	89.8	-	-	-	78.3	90.5	-	-	-
PADS [50]	128BN	67.3	78.0	69.9	-	-	83.5	89.7	68.8	-	-	76.5	89.0	89.9	-	-
ProxyNCA++ [60]	512R	69.0	79.8	73.9	-	-	86.5	92.5	73.8	-	-	80.7	92.0	-	-	-
NIR [51]	512R	70.5	80.6	72.5	-	-	89.1	93.4	75.0	-	-	80.7	91.5	90.9	-	-
Ensemble-based methods:																
A-BIER [47]	512G	57.5	68.7	-	-	-	82.0	89.0	-	-	-	74.2	86.9	-	-	-
Ranked [68]	1536BN	61.3	72.7	66.1	-	-	82.1	89.3	71.8	-	-	79.8	91.3	90.4	-	-
DREML [72]	9216R	63.9	75.0	67.8	-	-	86.0	91.7	76.4	-	-	-	-	-	-	-
D & C [53]	128R	65.9	76.6	69.6	-	-	84.6	90.7	70.3	-	-	75.9	88.4	90.2	-	-
DRML [80]	512BN	68.7	78.6	69.3	-	-	86.9	92.1	72.1	-	-	79.9	90.7	90.1	-	-
DiVA [43]	512R	69.2	79.3	71.4	-	-	87.6	92.9	72.2	-	-	79.6	91.2	90.6	-	-
IRT$_\mathbf{R}$ [18]	384D-S	76.6	85.0	-	-	-	-	-	-	-	-	84.2	93.7	-	-	-
ABE-8* [34]	384D-S	77.0	86.1	78.5	35.6	45.3	87.1	92.6	74.0	29.3	39.5	82.6	92.8	93.0	57.1	60.4
Hyp [20]	384D-S	77.8	86.6	-	-	-	86.4	92.2	-	-	-	83.3	93.5	-	-	-
DCML-PA* [79]	384D-S	78.4	86.4	78.6	36.1	46.1	87.8	92.8	74.1	30.0	39.8	83.4	93.4	93.7	58.8	61.5
Triplet-SH* [54]	384D-S	74.0	83.3	74.5	30.8	41.4	84.1	89.5	70.9	27.6	37.7	79.9	91.2	91.3	53.7	56.6
DFML-TSH	384D-S	**75.8**	**84.2**	**76.9**	**33.8**	**44.4**	**85.2**	**91.2**	**72.2**	**29.3**	**39.1**	**81.6**	**92.4**	**92.3**	**55.5**	**58.2**
N-Pair* [55]	384D-S	75.5	83.9	76.4	33.9	44.1	83.6	89.3	70.5	27.5	37.5	79.1	90.2	90.5	52.3	55.3
DFML-NP	384D-S	**76.8**	**84.9**	**77.5**	**35.6**	**46.0**	**85.4**	**91.3**	**72.4**	**29.8**	**39.5**	**80.0**	**91.0**	**91.3**	**53.9**	**56.8**
Margin-DW* [71]	384D-S	76.2	84.2	77.2	34.2	44.6	85.2	91.1	71.9	27.4	37.2	81.5	92.2	92.0	55.5	58.3
DFML-MDW	384D-S	**77.4**	**85.4**	**78.8**	**35.6**	**45.5**	**86.1**	**92.1**	**72.9**	**29.8**	**39.7**	**82.3**	**92.8**	**92.5**	**56.8**	**59.8**
ProxyNCA* [44]	384D-S	76.4	84.8	77.1	35.2	45.3	84.3	90.0	71.1	27.9	37.9	82.1	92.7	92.6	56.7	59.6
DFML-PN	384D-S	**78.1**	**85.6**	**78.1**	**36.6**	**46.5**	**85.8**	**91.2**	**72.5**	**29.7**	**39.5**	**83.2**	**93.2**	**93.6**	**58.1**	**61.2**
Contrastive* [28]	384D-S	76.4	84.7	77.4	34.5	44.4	86.0	91.9	72.8	28.4	38.4	82.3	93.0	92.7	56.7	59.8
DFML-Con	384D-S	**77.2**	**85.2**	**77.9**	**35.8**	**46.0**	**87.2**	**92.6**	**73.9**	**29.6**	**39.3**	**83.1**	**93.2**	**93.5**	**58.9**	**61.0**
ProxyAnchor* [33]	384D-S	77.5	85.7	79.0	35.7	45.8	87.7	92.9	74.1	29.9	39.6	82.7	93.1	93.1	58.0	60.9
DFML-PA	384D-S	**79.1**	**86.8**	**80.2**	**37.5**	**47.3**	**89.5**	**93.9**	**76.8**	**31.0**	**40.6**	**84.2**	**93.8**	**94.1**	**59.7**	**62.6**

products 60,502 images are for testing. For image classification, we evaluated our DFML framework on ImageNet-1K [52], which contains 1,000 categories of images. The training set consists of 1,200,000 images and the test set comprises 50,000 samples.

4.2. Implementation Details

We adopted the conventional deep metric learning setting [55] to evaluate the proposed DFML framework. We employed the ImageNet [52] pretrained DeiT-Small (D-S) model provided by Timm [70] as the backbone, which contains a knowledge distillation token for efficient training [61]. The parameters of the factorized sub-blocks were loaded according to (13) and (15). We abandoned the last classification head and fixed the embedding size to 384. Ablation studies of other architectures are involved as well. During training, we randomly cropped the training images to 224 × 224 with a random horizontal flipping of 50% probability. We set the batch size to 120 and adopted the AdamW optimizer with a learning rate of $1e^{-4}$. The number of the sub-blocks K was set to 3 in the main experiments. We fixed the hyper-parameters λ_s and λ_f to 1.0 and set the training epoch to 50. During testing, we resized the images to 256 × 256 and then center-cropped them to 224 × 224. Specifically, we utilized the complete output of the backbone to conduct the evaluation process. We provided various evaluation metrics [45, 55] including Recall@Ks, normalized mutual information (NMI), Mean Average Precision at R (M@R), and R-Precision (RP).

4.3. Results and Analysis

Comparisons with State-of-the-art Methods for Image Retrieval. We compared our DFML framework with state-of-the-art deep metric learning approaches for image retrieval, including conventional deep metric learning methods and ensemble-based methods. To verify the effectiveness of our framework, we applied DFML to various loss functions and sampling strategies including the triplet loss with semi-hard sampling (Triplet-SH) [54], the N-Pair loss [55], the margin loss with distance-weighted sampling (Margin-DW) [71], the ProxyNCA loss [44], the contrastive loss [28], and the ProxyAnchor loss [33].

Table 2. Effect of different numbers of sub-blocks.

	CUB-200-2011		Cars196		SOP	
Number	R@1	M@R	R@1	M@R	R@1	M@R
1	77.5	35.7	87.7	29.9	82.7	58.0
2	78.8	37.3	89.4	**31.0**	**84.3**	**59.7**
3	**79.1**	**37.5**	**89.5**	**31.0**	84.2	**59.7**
6	78.4	37.0	89.2	30.6	84.0	59.6

Table 3. Ablation study of different loss functions.

	CUB-200-2011		Cars196		SOP	
Method	R@1	M@R	R@1	M@R	R@1	M@R
DFML w/o J_{dis}	70.6	25.5	53.8	8.2	56.5	23.6
DFML w/o J_d	77.5	35.7	87.7	29.9	83.2	57.9
DFML w/o J_f	78.5	36.7	88.5	30.6	83.7	59.3
DFML-PA	**79.1**	**37.5**	**89.5**	**31.0**	**84.2**	**59.7**

Table 4. Experimental results of the soft gating mechanism.

	CUB-200-2011		Cars196		SOP	
Method	R@1	M@R	R@1	M@R	R@1	M@R
ProxyAnchor	77.5	35.7	87.7	29.9	82.7	58.0
Soft-gating-PA	77.4	35.7	87.8	30.0	82.5	58.1
DFML-PA	**79.1**	**37.5**	**89.5**	**31.0**	**84.2**	**59.7**

The experimental results on three datasets are shown in Table 1. The bold numbers indicate the improvement of the proposed DFML framework compared with the baseline methods. The best results and the second best results are presented with red and blue colors, respectively. We observe that DFML achieves a constant performance boost to existing methods on three datasets. In particular, DFML performs the best when applied to the ProxyAnchor loss and achieves the best results on all the datasets., surpassing the original performance by 1.8% at Recall@1 and 2.7% at NMI on the Cars196 dataset. This is because DFML factorizes the backbone architecture and extracts more diverse information from the training samples, thus improving the generalization ability on unseen classes.

Number of Sub-blocks. The proposed DFML framework divides each block in the vision transformer backbone into K sub-blocks to conduct the metric factorization. We analyzed the effect of the number of sub-blocks on three datasets. We applied DFML to the ProxyAnchor loss on DeiT-Small with the original attention head number of 6 and the MLP hidden dimension of 1536. We used 1, 2, 3, and 6 as the number of sub-blocks, rendering the head number of each MHSA sub-block to 6, 3, 2, 1 and the hidden dimension of each MLP sub-block to 1536, 768, 512, 256. The experimental results are illustrated in Table 2. We observe that the performance generally improves as the number of sub-blocks rises and DFML achieves the best results when $K = 3$ for the CUB-200-2011 and Cars196 datasets. This demonstrates that increasing K essentially enhances the generalization ability on unseen classes because more routines are available. However, the results become worse when $K = 6$, which means that the model might fail to extract sufficient information from images when we further compress the head number of the MHSA sub-block and the hidden dimension of the MLP sub-block.

Table 5. Effect of different backbone architectures. § denotes the backbone pretrained on ImageNet-21K.

	CUB-200-2011		Cars196	
Method	R@1	M@R	R@1	M@R
BN (baseline)	68.5	27.2	86.2	29.8
DFML-BN	69.3	27.8	88.4	30.6
D-T (baseline)	72.2	31.0	82.4	24.6
DFML-D-T	73.1	31.7	83.1	25.5
D-S (baseline)	77.5	35.7	87.7	29.9
DFML-D-S	79.1	37.5	89.5	31.0
D-B (baseline)	80.2	39.3	90.3	34.7
DFML-D-B	81.2	39.9	**91.4**	**36.3**
V-T (baseline)	68.2	27.1	79.8	22.8
DFML-V-T	69.1	28.0	80.9	23.2
V-S (baseline)	73.8	31.1	85.5	27.9
DFML-V-S	75.9	32.8	86.4	28.7
V-B (baseline)	78.7	36.6	89.5	33.4
DFML-V-B	79.8	37.0	90.7	33.9
V-T (baseline) §	78.3	37.7	78.9	22.8
DFML-V-T §	79.1	39.5	79.4	23.1
V-S (baseline) §	85.7	50.2	87.3	31.6
DFML-V-S §	86.3	51.2	88.4	32.0
V-B (baseline) §	87.1	53.6	89.4	33.3
DFML-V-B §	**87.8**	**54.0**	90.5	34.5

Table 6. Experimental results (%) of the proposed DFML on the ImageNet-1K dataset. † denotes that the metric factorization was only applied to the MLP blocks.

Method	Epoch	#param	FLOPs	Top-1 Acc
V-S (baseline)	300	22M	4.6G	79.8
DFML-V-S	100	22M	2.4G (-48%)	79.0 (-0.8)
DFML-V-S	300	22M	2.4G (-48%)	79.3 (-0.5)
DFML-V-S†	100	22M	3.5G (-24%)	79.5 (-0.3)
DFML-V-S†	300	22M	3.5G (-24%)	79.8 (+0.0)
V-B (baseline)	300	86M	17.5G	81.8
DFML-V-B	100	86M	9.1G (-48%)	80.8 (-1.0)
DFML-V-B	300	86M	9.1G (-48%)	81.4 (-0.4)
DFML-V-B†	100	86M	13.3G (-24%)	81.6 (-0.2)
DFML-V-B†	300	86M	13.3G (-24%)	82.0 (+0.2)

Ablation Study of Each Loss Function. We conducted an ablation study of each loss function in the DFML framework on three datasets. The results for comparison are presented in Table 3. We can see that combining all three losses in the experiment achieves the most competitive performance on both datasets, which indicates the effectiveness of the adopted loss functions.

Ablation Study of the Gating Mechanism. We conducted experiments to compare the effect of different gating mechanisms. Specifically, we introduced a soft gating mechanism for all self-attention heads and FFN channels, as shown in Table 4. We observe that simply using the soft gating mechanism cannot improve the final performances.

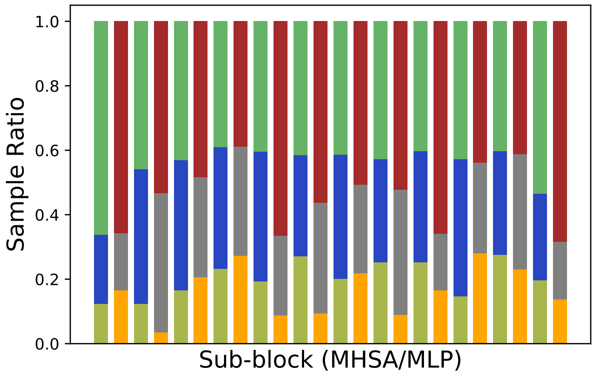

Figure 4. Sample ratio of each MHSA and MLP sub-block.

Effect of Different Backbone Architectures. We tested the performance of DFML on different backbone architectures. Specifically, we employed BN-Inception (BN), DeiT-X and ViT-X (X=Tiny, Small, Base) as the backbone, respectively. DeiT-X models are pretrained on the ImageNet-1K [52] dataset with an extra token to distill knowledge from teacher models. Differently, ViT-X models are pretrained without the distillation token but might be on the larger ImageNet-21K [14] dataset. We instantiated DFML with the ProxyAnchor loss for all the architectures. We provided the results on the CUB-200-2011 and Cars196 datasets in Table 5. We can see that DFML constantly improves the performance of the baseline models. Specially, we observe the best results on ViT-Base pretrained on ImageNet-21K for CUB-200-2011 and on DeiT-Base for Cars196, demonstrating the effectiveness of DFML.

Effectiveness of the Factorized Network. We also evaluated the effect of DFML on the learned backbone. We followed mainstream methods of network architecture to conduct experiments on the ImageNet-1K [52] dataset. As they usually adopt the accuracy/computation trade-off to measure the performance of a backbone network, we use the factorized network for comparisons. The factorized network can greatly reduce the inference computation (measured by floating-point operations per second, FLOPs) since each token is only routed to only one sub-block at each layer. As shown in Table 6, we observe that DFML can effectively maintain the performance of the original network with significantly less computation, demonstrating a better trade-off. In particular, when only factoring the MLP blocks, DFML even achieves +0.2 performance with 24% fewer FLOPs. This further verifies the effectiveness of enforcing different sub-modules to focus on diverse features.

Diversity of the Factorized Network. The diversity of the factorized network is essential to the generalization ability on unseen classes. We conducted experiments on CUB-200-2011 and provided the sample ratio of each MHSA and MLP sub-block, as shown in Figure 4, where the attention blocks and the MLP blocks are staggered with different colors. We fixed the number of sub-blocks for each MHSA and MLP block to 3 and calculated the percentage of samples that were passed through each factorized sub-block. We observe that different sub-blocks learn to process diverse training samples and each sub-block is chosen by at least a certain proportion of samples. This verifies that different images are employed to update the parameters of various components of the backbone, which indicates that DFML encodes more diverse characteristics from the input images.

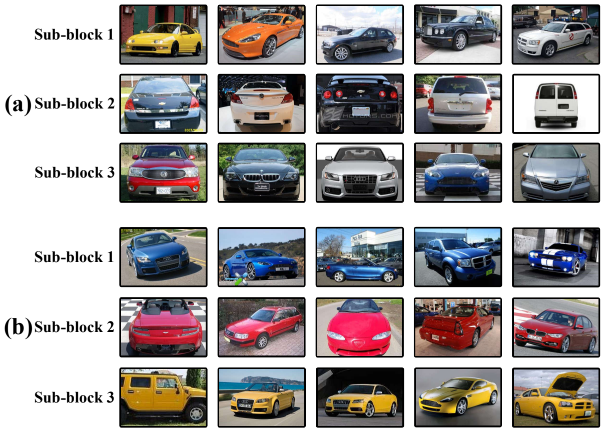

Figure 5. Qualitative results of samples passing through the last (a) MHSA block and (b) MLP block on the Cars196 dataset.

Qualitative Results. We qualitatively provided several samples passing through the last MHSA block and MLP block on the Cars196 dataset, as shown in Figure 5. We observe that the last MHSA block tends to encode the pose features of the cars. Differently, the final MLP block distinguishes the samples by their colors. This demonstrates that DFML indeed extracts diverse characteristics from the input images in addition to the class information.

5. Conclusion

In this paper, we have presented a deep factorized metric learning framework to factorize the backbone network and optimize the components of the network with different training samples. We have factorized the network to various sub-blocks and allocated the samples to each sub-block chosen by learnable routers. We have applied discriminative losses on the factorized output of the network and updated the parameters of the routers with a reconstruction objective. We have performed experiments on four benchmark datasets and demonstrated the effectiveness of DFML.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 62125603, and in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI).

References

[1] Nicolas Aziere and Sinisa Todorovic. Ensemble deep manifold similarity learning using hard proxies. In *CVPR*, pages 7299–7307, 2019. 1

[2] Leo Breiman. Random forests. *ML*, 45(1):5–32, 2001. 2

[3] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *NeurIPS*, pages 8224–8234, 2018. 2

[4] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, pages 1861–1870, 2019. 2

[5] Rui Cao, Qian Zhang, Jiasong Zhu, Qing Li, Qingquan Li, Bozhi Liu, and Guoping Qiu. Enhancing remote sensing image retrieval using a triplet deep metric learning network. *IJRS*, 41(2):740–751, 2020. 1

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 2

[7] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, pages 1189–1198, 2018. 1

[8] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021. 2

[9] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, pages 1335–1344, 2016. 2

[10] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *TGRS*, 56(5):2811–2821, 2018. 1

[11] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. 2021. 2

[12] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *ICCV*, pages 2988–2997, 2021. 2

[13] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, pages 1601–1610, 2021. 2

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 8

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2, 5

[16] Yueqi Duan, Lei Chen, Jiwen Lu, and Jie Zhou. Deep embedding learning with discriminative sampling policy. In *CVPR*, pages 4964–4973, 2019. 2

[17] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *CVPR*, pages 2780–2789, 2018. 2

[18] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *arXiv*, abs/12102.05644, 2021. 2, 6

[19] Haytham Elghazel and Alex Aussem. Unsupervised feature selection with ensemble learning. *ML*, 98(1-2):157–180, 2015. 2

[20] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *CVPR*, pages 7409–7419, 2022. 6

[21] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv*, abs/1703.10277, 2017. 1

[22] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R Scott. Deep metric learning with hierarchical triplet loss. In *ECCV*, pages 269–285, 2018. 2

[23] Soumyadeep Ghosh, Richa Singh, and Mayank Vatsa. On learning density aware embeddings. In *CVPR*, pages 4884–4892, 2019. 2

[24] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. 2021. 2

[25] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *TPAMI*, 12(10):993–1001, 1990. 2

[26] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *ICCV*, pages 2840–2848, 2017. 2

[27] Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *PR*, 41(9):2742–2756, 2008. 2

[28] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, pages 1875–1882, 2014. 6

[29] Chen Huang, Chen Change Loy, and Xiaoou Tang. Local similarity-aware deep feature embedding. In *NeurIPS*, pages 1262–1270, 2016. 2

[30] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv*, abs/2106.03650, 2021. 2

[31] Jian Kang, Ruben Fernandez-Beltran, Zhen Ye, Xiaohua Tong, Pedram Ghamisi, and Antonio Plaza. Deep metric learning based on scalable neighborhood components for remote sensing scene characterization. *TGRS*, 58(12):8905–8918, 2020. 1

[32] Mete Kemertas, Leila Pishdad, Konstantinos G Derpanis, and Afsaneh Fazly. Rankmi: A mutual information maximizing ranking loss. In *CVPR*, pages 14362–14371, 2020. 6

[33] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, pages 3238–3247, 2020. 2, 5, 6

[34] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *ECCV*, pages 760–777, 2018. 2, 6

[35] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013. 2, 5

[36] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *ICLR*, 2018. 2

[37] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *CVPR*, pages 9256–9266, 2019. 1

[38] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv*, abs/2104.05707, 2021. 2

[39] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *ECCV*, pages 689–704, 2018. 2

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. 2021. 2

[41] Jing Lu, Chaofan Xu, Wei Zhang, Ling-Yu Duan, and Tao Mei. Sampling wisely: Deep image embedding by top-k precision optimization. In *ICCV*, pages 7961–7970, 2019. 2

[42] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. 2021. 2

[43] Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation fordeep metric learning. In *ECCV*, 2020. 1, 2, 6

[44] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368, 2017. 1, 2, 6

[45] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, 2020. 6

[46] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Bier - boosting independent embeddings robustly. In *ICCV*, pages 5189–5198, 2017. 2

[47] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with bier: Boosting independent embeddings robustly. *TPAMI*, 2018. 1, 6

[48] Qi Qian, Lei Shang, Baigui Sun, and Juhua Hu. Softtriple loss: Deep metric learning without triplet sampling. In *ICCV*, 2019. 6

[49] Karsten Roth, Biagio Brattoli, and Bjorn Ommer. Mic: Mining interclass characteristics for improved metric learning. In *ICCV*, pages 8000–8009, 2019. 1, 6

[50] Karsten Roth, Timo Milbich, and Bjorn Ommer. Pads: Policy-adapted sampling for visual similarity learning. In *CVPR*, pages 6568–6577, 2020. 2, 6

[51] Karsten Roth, Oriol Vinyals, and Zeynep Akata. Non-isotropy regularization for proxy-based deep metric learning. In *CVPR*, pages 7420–7430, 2022. 2, 6

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 2, 6, 8

[53] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *CVPR*, pages 471–480, 2019. 1, 2, 6

[54] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 2, 6

[55] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, pages 1857–1865, 2016. 1, 2, 5, 6

[56] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016. 1, 2

[57] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. 2021. 2

[58] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *CVPR*, pages 7251–7259, 2019. 2

[59] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *CVPR*, pages 6398–6407, 2020. 2, 6

[60] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *ECCV*, 2020. 2, 6

[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 2, 6

[62] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *ECCV*, pages 550–564, 2018. 2

[63] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 5

[64] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014. 2

[65] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *ICCV*, pages 2593–2601, 2017. 2

[66] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. 2021. 2

[67] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, pages 5022–5030, 2019. 2

[68] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In *CVPR*, pages 5207–5216, 2019. 6

[69] Marco A Wiering and Hado Van Hasselt. Ensemble algorithms in reinforcement learning. *TSMC*, 38(4):930–936, 2008. 2

[70] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 6

[71] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, pages 2859–2867, 2017. 6

[72] Hong Xuan, Richard Souvenir, and Robert Pless. Deep randomized ensembles for metric learning. In *ECCV*, pages 723–734, 2018. 1, 2, 6

[73] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *ICCV*, pages 6490–6499, 2019. 2

[74] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. 2021. 2

[75] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *ICCV*, pages 814–823, 2017. 2

[76] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *CVPR*, pages 72–81, 2019. 2

[77] Wenzhao Zheng, Jiwen Lu, and Zhou Jie. Structural deep metric learning for room layout estimation. In *ECCV*, 2020. 1

[78] Wenzhao Zheng, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. *TPAMI*, 2020. 1

[79] Wenzhao Zheng, Chengkun Wang, Jiwen Lu, and Jie Zhou. Deep compositional metric learning. In *CVPR*, pages 9320–9329, 2021. 1, 2, 6

[80] Wenzhao Zheng, Borui Zhang, Jiwen Lu, and Jie Zhou. Deep relational metric learning. In *ICCV*, pages 12065–12074, 2021. 1, 2, 6

[81] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 2