Multi-Proxy Learning from an Entropy Optimization Perspective

Yunlong Yu[1] , **Dingyi Zhang**[1] , **Yingming Li**[1] and **Zhongfei Zhang**[2]
[1]College of Information Science and Electronic Engineering, Zhejiang University
[2] Computer Science Department, Binghamton University
{yuyunlong, dyzhang, yingming}@zju.edu.cn, zhongfei@cs.binghamton.edu

Abstract

Deep Metric Learning, a task that learns a feature embedding space where semantically similar samples are located closer than dissimilar samples, is a cornerstone of many computer vision applications. Most of the existing proxy-based approaches usually exploit the global context via learning a single proxy for each training class, which struggles in capturing the complex non-uniform data distribution with different patterns. In this work, we present an easy-to-implement framework to effectively capture the local neighbor relationships via learning multiple proxies for each class that collectively approximate the intra-class distribution. In the context of large intra-class visual diversity, we revisit the entropy learning under the multi-proxy learning framework and provide a training routine that both minimizes the entropy of intra-class probability distribution and maximizes the entropy of inter-class probability distribution. In this way, our model is able to better capture the intra-class variations and smooth the inter-class differences and thus facilitates to extract more semantic feature representations for the downstream tasks. Extensive experimental results demonstrate that the proposed approach achieves competitive performances. Codes and an appendix are provided [1].

1 Introduction

Deep Metric Learning (DML) is a core component of a variety of computer vision applications including face recognition [Meng *et al.*, 2021], fine-grained retrieval [Liu *et al.*, 2016], and few-shot learning [Snell *et al.*, 2017], which involves learning an effective similarity measure between samples. The basic idea for DML is to learn a feature embedding space via pulling together samples from the same class and pushing apart samples from the different classes. The existing approaches usually learn such a desired space by optimizing different loss functions, such as triplet loss [Hoffer and Ailon, 2015], contrastive loss [Chopra *et al.*, 2005], and proxyNCA loss [Movshovitz-Attias *et al.*, 2017]. These losses are roughly divided into two categories: pair-based losses and proxy-based losses. The former captures the data relations with either pair, triplets, or a group of samples in a mini-batch, which could provide rich supervisory data-to-data information for training the feature embedding space. However, they empirically suffer from high training complexity and sampling issues as the number of pairs, triplets, or tuplets of samples is exponentially increased. In contrast, the latter assigns a learnable proxy for each class and optimizes the similarity between the feature embeddings and the class proxies in a data-to-class way, which has been demonstrated to perform competitively with the pair-based approaches [Musgrave *et al.*, 2020; Teh *et al.*, 2020; Boudiaf *et al.*, 2020; Roth *et al.*, 2020].

In reality, the fine-grained visual samples are commonly not uniformly distributed but have a complex intra-class structure. The samples in the different clusters are often related to different characteristics, such as color, pose, and location. To this end, assigning each class with only one proxy often leads to poor local minima, due to inappropriately exploiting the embedding space. In this paper, we design a novel approach called Multi-Proxy Learning (MPL) under the data-to-class framework to capture the non-uniform intra-class patterns via learning a set of proxies for each category. The captured patterns could then be used for the downstream tasks of interest. Concretely, our approach introduces multiple proxies for each category to assist in learning a succinct, high-level semantic feature embedding space for improving the model's generalization ability.

Several prior approaches exploit the class distribution with a set of proxies for each category via either merging similar centers [Qian *et al.*, 2019] or learning a large amount of proxies [Zhu *et al.*, 2020]. However, these approaches implicitly differentiate the intra-class proxies, which may suffer from the mode collapse issue during the training stage, i.e., the intra-class proxies tend to deteriorate to be identical.

In this paper, we revisit the entropy regime and design two novel regularization terms respectively for the intra-class and the inter-class proxies. Our approach is based on the underlying fact: the entropy of the probability logit vectors generated by the feature embeddings and the class proxies is a measure of the "confidence" that the sample belongs to the corresponding class. On one hand, learning the feature representation model that has a higher value of the output entropy reduces the "confidence" of the sample to be correctly

[1] https://github.com/yunlongyu/MPL

classified, which smooths the predictions from all the classes and would alleviate the overfitting issue. On the other hand, learning the model that has a lower value of output entropy among the intra-class proxies increases the "confidence" of the sample to be one of the intra-class proxies, leading to a better discrimination ability among the intra-class proxies.

In a nutshell, our MPL intrinsically reshapes the feature representation learning from the following three aspects. First, a united multi-proxy learning framework. The proposed MPL introduces multiple proxies for each class and enjoys the merits from both the proxy-based and pair-based paradigms, which automatically selects the nearest proxy from each negative class as a hard sample for optimization. Second, the intra-class diversity. During training, we enforce the intra-class proxies to have a large diversity to prevent them from being too similar and alleviate the collapse issue, which helps to characterize the intra-class variation sufficiently. Finally, the inter-class smoothness. The inter-class smoothness regularization term encourages the smoothness of the inter-class predictions, bringing in a better generalization ability when combined with the intra-class diversity regularization term.

2 Related Work

2.1 Deep Metric Learning

DML is an important component in both machine learning and computer vision communities. Most of the existing DML approaches focus on the design of the loss functions, which are divided into pair-based approaches (e.g., N-pair loss [Sohn, 2016], lifted structure loss [Oh Song *et al.*, 2016], Margin [Wu *et al.*, 2017], MS loss [Wang *et al.*, 2019]) and proxy-based approaches (e.g., ProxyNCA loss [Movshovitz-Attias *et al.*, 2017], Proxy-Anchor loss [Kim *et al.*, 2020], Proxy synthesis [Gu *et al.*, 2021], ProxyNCA++ [Teh *et al.*, 2020]) based on the loss formulation.

The pair-based approaches focus on optimizing pairwise constraints, which minimize the distances between the intra-class samples while maximize the distances between the inter-class samples. However, such approaches potentially suffer from the high training complexity and sampling issues. To alleviate these issues, the proxy-based approaches [Kim *et al.*, 2020; Teh *et al.*, 2020] assign each class with a learnable proxy to provide a global context during each training iteration. The current literature has shown that the proxy-based approaches could achieve a comparable performance with the pair-based approaches. In [Sun *et al.*, 2020], Sun et al. demonstrated that both proxy-based and pair-based approaches could be formulated into a united framework.

In most formulations of the existing proxy-based losses, emphasis has been paid to learning a single proxy for each of the classes. In contrast, we focus instead on learning several proxies for each class to characterize the fine-grained intra-class distribution. This formulation has also been explored in the two previous works [Qian *et al.*, 2019; Zhu *et al.*, 2020]. The difference is that we develop the multi-proxy learning framework by minimizing the entropy of intra-class proxies to prevent them from being too similar and maximizing the entropy of inter-class proxies to improve the model's generalization ability.

2.2 Entropy Learning

Entropy in information science is usually used to describe a probability distribution and has been widely explored in different tasks, from supervised learning [He *et al.*, 2016; Dubey *et al.*, 2018] to unsupervised learning [Melacci and Gori, 2012; Rutquist, 2019]. For the supervised classification task, a model is basically optimized by minimizing cross-entropy between the predictions and their ground truth. In the context of either semi-supervised [Grandvalet and Bengio, 2005; Saito *et al.*, 2019; Sohn *et al.*, 2020] or unsupervised learning [Melacci and Gori, 2012; Rutquist, 2019], minimizing the entropy value of the predictions performs as a regularization term to shape a model and to obtain appealing predictions. In addition to employing the entropy principle to constrain the distribution consistency between the predictions and the ground-truth, we further employ it to regularize the predictions, to achieve intra-class diversity and inter-class smoothness.

3 Method

3.1 Preliminaries

Given a training set D_{tr} of n instances from C classes, the deep metric learning aims at learning an embedding function $f_{\mathbf{\Theta}}$ that maps the visual instance x to the feature embedding $\mathbf{v}$ with $\mathbf{v} = f_{\mathbf{\Theta}}(x)$, such that the similar instances are close to each other and the dissimilar instances are far away from each other in the feature embedding space. The classification [Zhai and Wu, 2018] or the proxy-based approaches [Teh *et al.*, 2020] introduce a faithfully represented proxy for each class stored as the learnable parameters and train the feature embedding model via pulling the samples to their corresponding class proxies and repelling them from the other class proxies. The probability of instance x to be predicted into class $c, c \in \{1, .., C\}$ is

$$p(c|x; \mathbf{P}, \mathbf{\Theta}) = \frac{\exp(s(\mathbf{p}_c, \mathbf{v})/\tau)}{\sum_{j=1}^{C} \exp(s(\mathbf{p}_j, \mathbf{v})/\tau)}, \tag{1}$$

where $\mathbf{p}_c$ denotes the proxy or the weight for class c, $s()$ denotes a similarity metric, e.g., cosine similarity, negative Euclidean distance, and inner product; τ is the temperature. During training, both the model parameter $\mathbf{\Theta}$ and the proxy matrix $\mathbf{P}$ are collectively optimized by minimizing the expected KL (Kullback-Liebler)-divergence of the predicted probability distribution from the true class label vector over the training set D_{tr}:

$$\mathbf{\Theta}^*, \mathbf{P}^* = \arg\min_{\mathbf{\Theta}, \mathbf{P}} \mathbb{E}_{x,y \sim D_{tr}} \mathbb{D}_{KL}(y || p(y|x; \mathbf{P}, \mathbf{\Theta})). \tag{2}$$

For the most metric learning paradigms, the model parameters $\mathbf{\Theta}$ are initialized with a pre-trained model and fine-tuned with the training set.

3.2 Multi-Proxy Learning (MPL)

Considering that the samples in each fine-grained class may roughly be clustered into a few groups due to the intra-class variation, we present a multi-proxy learning framework that assigns a set of proxies $\mathbf{p}_y^r$, $(r = 1, 2, ..., R)$, for each class, to capture the intra-class distribution and increase the model's

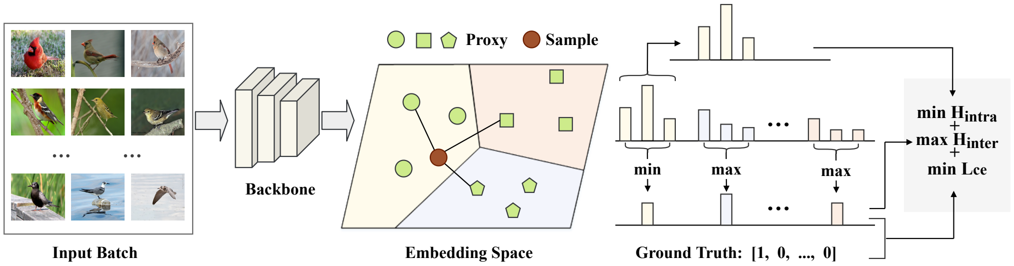

Figure 1: Illustration of the proposed Multi-Proxy Learning framework. The input images are projected into the feature embedding space, where three constraints are imposed, i.e., a classification loss L_{ce}, an inter-class smoothness regularization H_{inter}, and an intra-class diversity regularization H_{intra}. The solid lines in the embedding space denote the proxies selected for the classification loss.

flexibility, where $R \geq 1$ is the proxy number of each class. As illustrated in Figure 1, the proposed MPL consists of a basic classification loss and two regularization terms, which will be introduced below.

Multi-Proxy Cross-Entropy Loss. For an instance x, we first project it into an embedding space with the feature representation model and then construct a multi-proxy cross-entropy loss to ensure the instance to be correctly classified. Specifically, the multi-proxy cross-entropy loss enforces the feature embedding of the instance x to be closer to its most dissimilar intra-class proxy than the other most similar proxies from all the negative classes. Thus, the conditional probability distribution over C classes is formulated as:

$$p(c|x;\mathbf{P},\mathbf{\Theta}) = \frac{\exp(\min_{\mathbf{p}_c^r \in P_c} s(\mathbf{p}_c^r,\mathbf{v})/\tau)}{\exp(\min_{\mathbf{p}_c^r \in P_c} s(\mathbf{p}_c^r,\mathbf{v})/\tau) + \sum_{j=1,j\neq c}^{C} \exp(\max_{\mathbf{p}_j^r \in P_j} s(\mathbf{p}_j^r,\mathbf{v})/\tau)}, \quad (3)$$

where P_c denotes the intra-class proxy set of class c, $\mathbf{p}_c^r$ denotes the r-th proxy of the class c. In this way, the model parameters $\mathbf{\Theta}$ and the proxy matrix $\mathbf{P}$ are collectively optimized by minimizing the cross-entropy loss of the predicted probability distribution from the true class label vector over the training set D_{tr}:

$$\arg\min_{\mathbf{\Theta},\mathbf{P}} L_{ce} = -\sum_{(x,y)\sim D_{tr}} \sum_{c=1}^{C} \mathbb{I}(c=y) \log p(c|x;\mathbf{P},\mathbf{\Theta}), \quad (4)$$

where $\mathbb{I}$ denotes the indicator function.

Intuitively, a proxy is a representative of a subset of intra-class training data that captures the global context of a subset of the class. In this way, the model mines the hardest intra-class proxy as well as the hardest inter-class proxies to ensure the instances to be classified correctly, which could be seen to simulate the mechanism of pair-based approaches to capture the informative and fine-grained relationships. Thus, it could benefit from both the proxy-based and the pair-based paradigms.

The loss function described above is capable of extracting discriminative information from the training classes, but still suffers from the following drawbacks. First, the proxies from the same class are not regularized, which may fail to capture the variations of the class as expected. Second, the loss function emphasizes to correctly classify the training instances and may suffer from the overfitting issue, which would lead to poor adaptation on the novel classes. To this end, we further propose two kinds of regularizations to mitigate these issues.

Intra-class Diversity Regularization. To capture the variation of the intra-class distribution, we introduce an intra-class proxy diversity regularization term to minimize the intra-class distribution entropy conditioned on both the input instances and the intra-class proxies themselves, which is formulated:

$$\arg\min_{\mathbf{\Theta},\mathbf{P}} H_{intra} = \sum_{x\sim D_{tr}} \mathrm{H}[\hat{p}(\cdot|x;\mathbf{\Theta},\mathbf{P})] + \sum_{i=1}^{C\times R} \sum_{j=1}^{C\times R} \mathbb{I}(i=j) \log p(j|\mathbf{p}_i;\mathbf{P}), \quad (5)$$

where $\mathrm{H}[\hat{p}(\cdot|x;\mathbf{\Theta},\mathbf{P})]$ denotes the entropy of conditional probability distribution of the input instance with regard to only intra-class proxies, which is given by:

$$\mathrm{H}[\hat{p}(\cdot|x;\mathbf{\Theta},\mathbf{P})] = -\sum_{r=1}^{R} \hat{p}(r|x;\mathbf{\Theta},\mathbf{P}) \log \hat{p}(r|x;\mathbf{\Theta},\mathbf{P}), \quad (6)$$

where

$$\hat{p}(r|x;\mathbf{\Theta},\mathbf{P}) = \frac{\exp(s(\mathbf{p}_{c(x)}^r,\mathbf{v})/\tau)}{\sum_{\mathbf{p}_{c(x)}^k \in P_{c(x)}} \exp(s(\mathbf{p}_{c(x)}^k,\mathbf{v})/\tau)}, \quad (7)$$

where $P_{c(x)}$ denotes the intra-class proxy set and $c(x)$ denotes the class that instance x belongs to. The entropy can be seen as a measure of the diversity of the predicted distribution. If an instance is significantly close to one of its intra-class proxies, then most of the mass will be concentrated at this proxy, resulting in an entropy close to 0. Conversely, if an instance is equally close to all the intra-class proxies, we obtain the maximum of the entropy. In the problem that the intra-class instances differ significantly, it is reasonable to encourage minimizing the entropy of the conditional probability distribution among the intra-class proxies such that the similarity between the instance and one intra-class proxy is distinguished from the rest. The second term of Eq. 5 diversifies the proxies via encouraging each proxy to be different from the other proxies.

In this way, the model would prevent the intra-class proxies from being degenerated via decentralizing the proxies and thus mitigates the mode collapse issue. To this end, the feature representation model produces more diverse intra-class feature representations, which benefits in the adaptation of the subsequent downstream tasks.

Inter-class Smoothness Regularization. Since the fine-grained classes are visually similar to each other, it is possibly detrimental to enforce the model to produce too confident outputs. If two classes are similar semantically, their representations should be close to each other in the feature embedding space. Thus, it is undesirable to force the model to give a zero cross-entropy loss during training. Instead, we encourage the class prediction being distributed smoothly under the premise that the predicted class label of an instance coincides with the ground truth. Specifically, this goal is achieved via maximizing both the overall entropy over all training set and the proxies themselves, which is formulated:

$$\arg\max_{\mathbf{\Theta},\mathbf{P}} H_{inter} = \sum_{x\sim D_{tr}} \mathrm{H}[p(\cdot|x;\mathbf{\Theta},\mathbf{P})] + \sum_{i=1}^{C} \mathrm{H}[p(\cdot|\bar{\mathbf{p}}_i;\mathbf{P})], \quad (8)$$

where

$$\mathrm{H}[p(\cdot|x;\mathbf{\Theta},\mathbf{P})] = -\sum_{c=1}^{C} p(c|x;\mathbf{\Theta},\mathbf{P}) \log p(c|x;\mathbf{\Theta},\mathbf{P}) \quad (9)$$

denotes the entropy of the conditional probability distribution among all the training classes; $p(c|x;\mathbf{\Theta},\mathbf{P})$ denotes the class prediction of an instance x obtained with Eq. (3),

$$\mathrm{H}[p(\cdot|\bar{\mathbf{p}}_i;\mathbf{P})] = -\sum_{c=1}^{C} p(c|\bar{\mathbf{p}}_i;\mathbf{P}) \log p(c|\bar{\mathbf{p}}_i;\mathbf{P}) \qquad (10)$$

denotes the entropy of the conditional probability distribution among all the proxies, $\bar{\mathbf{p}}_i$ is the average value of the intra-class proxies from class i.

Once the inter-class proxy smoothness term is employed to regularize the feature representation learning, the model is encouraged to produce smoother class predictions, leading to more semantic meaningful feature representations. In the experiments, we will show that for fine-grained datasets with low inter-class diversity, the inter-class proxy smoothness regularization improves the model's performances when combined with the intra-class proxy diversity regularization.

Overall Loss Function. With both regularization terms on proxies, our final loss function becomes

$$\mathbf{\Theta}^*, \mathbf{P}^* = \arg\min_{\mathbf{\Theta},\mathbf{P}} L_{ce} - \alpha H_{inter} + \beta H_{intra}, \qquad (11)$$

where α and β are the hyper-parameters. We train the whole model end-to-end to yield a discriminative metric space, which is then applied to the downstream tasks, such as image retrieval, clustering, and few-shot classification.

4 Experiments

We comprehensively evaluate the effectiveness of the proposed MPL on two tasks, i.e., fine-grained image retrieval and clustering. We show that MPL is competent for both tasks. More experiments are reported in the supplementary.

4.1 Settings

We conduct experiments on three benchmark datasets: CUB [Wah *et al.*, 2011], Cars196 [Krause *et al.*, 2013], and Stanford Online Products (SOP) [Oh Song *et al.*, 2016]. CUB dataset consists of 11,788 images from 200 bird species, where the first 100 species are used for training and the rest 100 species are used for evaluation. The Cars196 dataset covers 16,185 images from 196 car classes, where the first 98 classes are used for training and the rest 98 classes are used for evaluation. SOP contains 59,551 images from 11,318 classes for training and 60,502 images of the rest 11,316 classes for evaluation. For the data pre-processing, we follow [Teh *et al.*, 2020] and obtain the training samples by randomly cropping 224×224 images from resized 256×256 images and applying random horizontal flipping for data augmentation. During the evaluation, we use a single center crop.

In the experiments, we employ the commonly used ImageNet pre-trained Resnet50 model [He *et al.*, 2016] and Inception with batch normalization (BN) [Ioffe and Normalization,] as our backbone with the feature embedding dimensionality as 512. If not specific, the results are obtained with Resnet50. The model is optimized with Adam [Kingma and Ba, 2015] for 50 epochs. We adopt the P-K sampling strategy to construct each batch with P=8 and K=4, where P is the class number in each batch and K is the sample number of each class. For both CUB and Cars196 datasets, we set the proxy number of each class to 5 during training. For the SOP dataset, the proxy number of each class is set to 2. The temperature value is set to $\frac{1}{9}$ across all the datasets. For the image retrieval task, Recall@k is used for evaluation. For the clustering task, we report the Normalized Mutual Information (NMI) to measure the clustering quality with the K-means clustering algorithm.

4.2 Comparisons with the SOTA Methods

For the fine-grained image retrieval and clustering tasks, we compare MPL with the eight competitors in Table 1. From the results, we observe that MPL achieves state-of-the-art performances in most of the cases. On one hand, MPL clearly outperforms the counterparts under different metrics on both CUB and Cars196 datasets. When comparing the results with BN, MPL obtains 0.3% performance gain for CUB and 0.7% gain for Cars196 over the second-best competitor for the image retrieval task, and obtains 1.1% performance gain for CUB and 1.2% gain for Cars196 over the second-best competitor for the clustering task. When comparing the results with Resnet50, MPL achieves 70.4% and 88.1% R@1 performances on CUB and Cars196 datasets for the image retrieval task, outperforming the second-best approaches by 1.9% and 0.8%, respectively. For the clustering task, MPL obtains 1.4% and 2.3% gains over the second-best approaches on CUB and Cars196, respectively. On the other hand, MPL does not achieve the best performance on the SOP dataset. The reason may be that each class of the SOP dataset only consists of 5 images on average, which makes it hard to construct a multi-center structure in each class and therefore goes against the advantage of learning multiple local cluster centers. However, MPL still performs competitively on the SOP dataset due to the benefits from both the proxy-based and pair-based learning paradigms.

4.3 Ablation Study

Proxy number per class. To evaluate the effects of the proxy number of each class, we conduct experiments on CUB

Method	Arch	Emb	CUB			Cars196			SOP
			R@1	R@2	NMI	R@1	R@2	NMI	R@1
ProxyNCA [Movshovitz-Attias *et al.*, 2017]	BN	128	49.2	61.9	59.5	73.2	82.4	64.9	73.7
MS+S2SD [Roth *et al.*, 2021]	BN	512	68.2	79.1	**71.6**	86.3	**92.2**	72.0	78.9
ProxyGML [Zhu *et al.*, 2020]	BN	512	66.6	77.6	69.8	85.5	91.8	72.4	78.0
SoftTriple [Qian *et al.*, 2019]	BN	512	65.4	76.4	69.3	84.5	90.7	70.1	78.3
Proxy-Anchor [Kim *et al.*, 2020]	BN	512	68.4	79.2	-	86.1	91.7	-	**79.1**
MPL (Ours)	BN	512	**68.7**	**79.4**	70.9	**86.8**	**92.2**	**73.6**	78.4
DCML-MDW [Zheng *et al.*, 2021]	R50	512	68.4	77.9	71.8	85.2	91.8	73.9	79.8
NormSoftMax [Zhai and Wu, 2018]	R50	512	61.3	73.9	-	84.2	90.4	-	78.2
Margin [Wu *et al.*, 2017]	R50	512	64.4	75.4	68.4	82.2	89.0	68.1	78.3
Trip-DiVA [Milbich *et al.*, 2020]	R50	512	68.5	78.5	71.1	87.3	92.8	72.1	**79.4**
ProxyNCA++†[Teh *et al.*, 2020]	R50	512	68.1	79.1	73.2	86.1	91.9	71.4	78.4
MPL (Ours)	R50	512	**70.4**	**80.6**	**74.6**	**88.1**	**93.1**	**74.4**	79.2

Table 1: Recall@k (in %) and NMI on both CUB and Cars196 datasets, and Recall@1 (%) on SOP dataset. "Arch" and "Emb" denote the network architecture and the dimensionality of the feature embedding, respectively. BN: Inception with batch normalization, R50: Resnet50. † indicates the methods implemented by ourselves with the released codes. Best results are marked in bold.

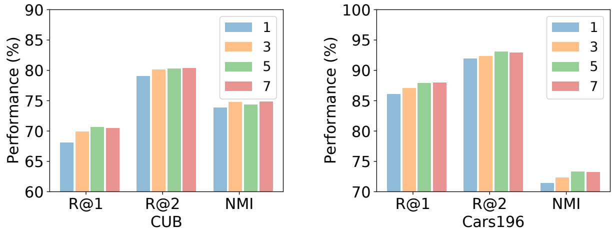

Figure 2: An ablation study about the effects of the proxy number of each class on the CUB and the Cars196 datasets.

Inter	Intra	CUB			Cars196		
		R@1	R@2	NMI	R@1	R@2	NMI
✗	✗	68.9	79.2	72.8	86.4	92.4	71.8
✓	✗	69.2	79.4	71.7	85.9	91.2	62.2
✗	✓	69.8	79.6	73.7	86.5	91.9	73.1
✓	✓	70.4	80.6	74.6	88.1	93.1	74.4

Table 2: An ablation study (in %) of the effects of intra-class diversity and inter-class smoothness regularization terms on both datasets.

and Cars196 datasets with different proxy numbers for each class. As illustrated in Figure 2, the performances gradually increase with a growing number of intra-class proxies, and the best performances are achieved with N=5 for most metrics on both CUB and Cars196 datasets. This indicates that the number of proxies matters for effectively capturing the intra-class distributions. We also observe that the performance plateaus when the number of proxy is above 5, possibly because the sample number for both CUB and Cars196 datasets are limited such that there is no further margin for improvement.

Impacts of hyper-parameters α and β. To evaluate the impacts of the two hyper-parameters on the performance, we conduct experiments on the Cars196 dataset varying their values from 0.5 to 2 with an interval of 0.5. From the results in Figure 3, we observe that the retrieval task is robust to both α and β, while the clustering task is a little sensitive to the variation of the hyper-parameters. On one hand, with the increase of α, the NMI drops slightly, which indicates that when the model overemphasizes smoothness, the discrimination ability is weakened correspondingly. On the other hand, with the increase of β, the NMI increases gradually and reaches a plateau, indicating that encouraging the intra-class proxies to be distributed yields benefits in improving the performance.

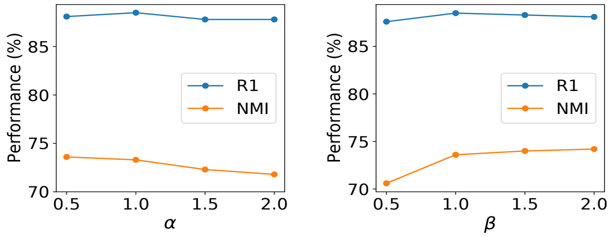

Figure 3: Impacts of the hyper-parameters α and β on the Cars196 dataset.

Size	Emb	CUB		Cars196	
		R@1	NMI	R@1	NMI
224×224	512	70.4	74.6	88.1	74.4
	1024	71.8	75.9	89.4	74.8
	2048	72.4	74.6	89.5	75.1
256×256	512	72.5	75.7	88.4	73.4
	1024	74.3	76.3	90.4	75.8
	2048	74.5	77.4	90.6	75.0

Table 3: Evaluation (%) on the input size and the dimensionality of embedding on both the CUB and the Cars196 datasets.

Impacts of regularization terms. In Table 2, we evaluate the impacts of two regularization terms on both CUB and Cars196 datasets. From the results, we have the following

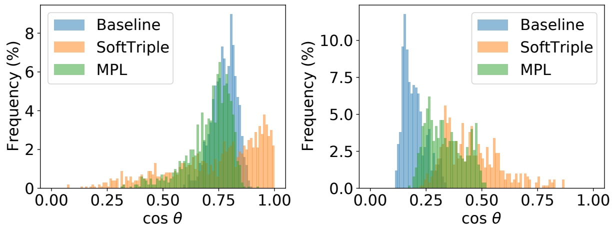

Figure 4: The cosine distributions of the intra-class proxies and the proxies to their inter-class nearest proxies on the training set of the CUB dataset.

observations. First, the inter-class regularization term brings a small improvement in the retrieval task on the CUB dataset but hurts the performances on the Cars196 dataset for both retrieval and clustering tasks. We speculate that the model would spoil the discriminative patterns with the inter-class regularization alone that smooths the prediction logit but neglects to distinguish the correct class. Second, the intra-class regularization term contributes to the performance improvements on both CUB and Cars196 datasets, which indicates the effectiveness of differentiating intra-class proxies. Finally, when the inter-class and the intra-class regularization terms are combined, the performances further improve on both CUB and Cars196 datasets. We argue that when the correct class is distinguished from the rest classes by assigning each sample to its nearest intra-class proxy, the smoothing regularization term benefits the generalization ability of the model.

Impacts of embedding dimensionality and input image size. To evaluate the impacts of the embedding dimensionality and input image size, we report the results in Table 3. From the results, we observe that larger input images significantly improve the performances on CUB and Cars196 datasets for both retrieval and clustering tasks. In general, the performances of input size with 256×256 obtain about 2% gains over those with 224×224 on the CUB dataset. Besides, we observe that the performances generally increase with the growing feature embedding dimensionality on both CUB and Cars196 datasets.

4.4 Further Analysis

Figure 4 gives the cosine distributions of the intra-class proxies and the proxies to their inter-class nearest proxies on the CUB training set. We take the model without the proposed two regularization terms as the baseline and select SoftTriple [Qian *et al.*, 2019] as a competitor. We observe that both the baseline and SoftTriple tend to suffer from the mode collapse issue since most of the intra-class proxies are close to each other (i.e., cosine similarities are close to 1). In contrast, the intra-class proxy cosine similarities of MPL distribute a more concentrated interval, which verifies that the intra-class regularization terms could help to shape the feature distribution. Besides, we observe that the cosine similarities between the proxies and their inter-class nearest proxies of MPL are larger than those of the baseline, which indicates that the inter-class regularization term smooths the inter-class distribution.

Figure 5 shows the cosine distributions of all positive (ap) pairs and all negative (an) pairs on the CUB dataset with the baseline and the MPL. We observe that the cosine distributions of the positive and negative pairs of MPL are more separated than those of the baseline on the test sets, which indicates that the proposed two regularization terms help to adapt to test classes.

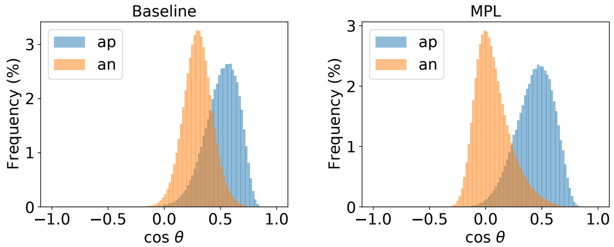

Figure 5: The cosine distributions of the baseline and MPL on the test set of the CUB dataset.

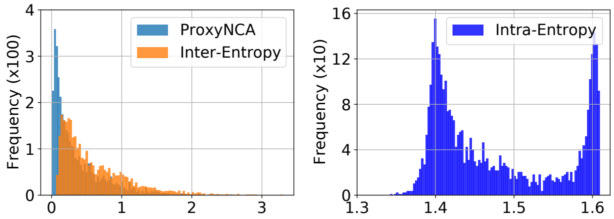

Figure 6: The entropy distributions of ProxyNCA and MPL on CUB dataset.

Deep evaluation of inter- and intra-entropy. Fig. 6 visualizes the distributions of the entropy values of ProxyNCA and MPL on CUB. We observe that the inter-entropy constraint smooths the class predictions as the inter-entropy values are distributed in larger values than ProxyNCA, which helps the model to alleviate the overfitting issue. Besides, the values of the intra-class entropy are in a bimodal distribution, indicating that some samples are distributed around the corresponding class centers while some samples are distributed close to one of the intra-class proxies. Thus, the inter- and intra-entropy constraints respectively help the model to alleviate the overfitting issue and preserve the diverse intra-class distribution.

5 Conclusion

In this paper, we have proposed a novel approach dubbed multi-proxy learning (MPL) for fine-grained feature representation learning, which introduces multiple proxies for each class and exploits the data-to-proxy relationships through an entropy learning scheme. By explicitly enforcing a large diversity among the intra-class proxies and encouraging the class predictions to be distributed smoothly, MPL effectively captures the complex non-uniform data distribution. The experimental results on two different tasks demonstrate that the proposed MPL achieves competitive results on both tasks.

Acknowledgments

This work is supported in part by Science and Technology Innovation 2030 –"New Generation Artificial Intelligence"

Major Project (No. 2018AAA0100904), NSFC (62002320, U19B2043, 61672456), the Key R&D Program of Zhejiang Province, China (2021C01119).

References

[Boudiaf *et al.*, 2020] Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *ECCV*, pages 548–564, 2020.

[Chopra *et al.*, 2005] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546, 2005.

[Dubey *et al.*, 2018] Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Maximum-entropy fine-grained classification. In *NeurIPS*, 2018.

[Grandvalet and Bengio, 2005] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *CAP*, pages 281–296, 2005.

[Gu *et al.*, 2021] Geonmo Gu, Byungsoo Ko, and Han-Gyu Kim. Proxy synthesis: Learning with synthetic classes for deep metric learning. In *AAAI*, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hoffer and Ailon, 2015] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *ICLR Workshop*, pages 84–92, 2015.

[Ioffe and Normalization,] Sergey Ioffe and Christian Szegedy Batch Normalization. Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*.

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

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Krause *et al.*, 2013] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *CVPR workshop*, pages 554–561, 2013.

[Liu *et al.*, 2016] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, pages 1096–1104, 2016.

[Melacci and Gori, 2012] Stefano Melacci and Marco Gori. Unsupervised learning by minimal entropy encoding. *TNNLS*, 23(12):1849–1861, 2012.

[Meng *et al.*, 2021] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. *arXiv:2103.06627*, 2021.

[Milbich *et al.*, 2020] Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation for deep metric learning. In *ECCV*, pages 590–607, 2020.

[Movshovitz-Attias *et al.*, 2017] 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.

[Musgrave *et al.*, 2020] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, pages 681–699, 2020.

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

[Qian *et al.*, 2019] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *CVPR*, pages 6450–6458, 2019.

[Roth *et al.*, 2020] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, pages 8242–8252, 2020.

[Roth *et al.*, 2021] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. Simultaneous similarity-based self-distillation for deep metric learning. In *ICML*, pages 9095–9106, 2021.

[Rutquist, 2019] Per Rutquist. Unsupervised learning through temporal smoothing and entropy maximization. In *CDC*, 2019.

[Saito *et al.*, 2019] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, pages 8050–8058, 2019.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

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

[Sun *et al.*, 2020] 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.

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

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. http://www.vision.caltech.edu/datasets/cub_200_2011, 2011.

[Wang *et al.*, 2019] 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.

[Wu *et al.*, 2017] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, pages 2840–2848, 2017.

[Zhai and Wu, 2018] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *BMVC*, 2018.

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

[Zhu *et al.*, 2020] Yuehua Zhu, Muli Yang, Cheng Deng, and Wei Liu. Fewer is more: A deep graph metric learning perspective using fewer proxies. In *NeurIPS*, pages 17792–17803, 2020.