A HIERARCHICAL MULTI-PROXY LOSS WITH DYNAMIC MAIN-PROXY FOR DEEP METRIC LEARNING

Lei Zhao[1], *Xiao-Lei Zhang*[1,2*]

[1]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
[2]Research and Development Institute of Northwestern Polytechnical University in Shenzhen, China

ABSTRACT

Proxy-based approaches in deep metric learning have recieved wide interest due to their efficient training process and rapid network convergence in the past few years. However, existing single-proxy methods aim to learn a common feature representation for each class by assigning a separate proxy for each class, which contradicts the inherent intra-class variance of samples from the same class, impeding more fine-gained similarity retrieval. In this paper, we propose a hierarchical multi-proxy method named dynamic main-proxy anchor (DMA) to address this issue. The approach first assigns multiple sub-proxies to learn different intra-class features and then utilizes a dynamically constructed main-proxy to handle class-related characteristics. In addition, we propose a regularization method to ensure closeness between similar sub-proxies and distance between dissimilar ones. Experimental results on three widely-used datasets show the superiority of the proposed DMA over the state-of-the-art methods in both retrieval and clustering tasks.

Index Terms— deep metric learning, proxy-based loss, multi-proxy, image retrieval

1. INTRODUCTION

Deep metric learning endeavors to train neural networks for a discriminative embedding space, enabling effective similarity estimation between samples. In this embedding space, the samples sharing similar characteristics exhibit closer spatial proximity, while the samples bearing dissimilar attributes demonstrate distinct spatial separations. To that end, different loss functions are designed to optimize the embedding space, which can be divided into two categories: pair-based methods and proxy-based methods.

The pair-based losses are constructed on the pairwise distances between data points in the embedding space. An exemplary pair-based loss is the contrastive loss [1, 2], which aims to minimize the distance between a pair of data samples if their class labels are identical, while also seeking to maximize their separation if the class labels are different. Another pair-based approach is the triplet method [3], which formulates the comparison of three instances, namely the anchor, a positive example, and a negative example. An essential requirement is that the distance between the anchor and the positive example should be smaller than the distance between the anchor and the negative example, surpassing a predefined margin. However, the majority of deep models are trained using Stochastic Gradient Descent (SGD), which operates on mini-batches of data during each iteration, therefore, the information contained within a mini-batch becomes inherently limited in contrast to the complete original dataset. In order to mitigate this issue, an efficacious sampling methodology must be devised for generating the mini-batches, and then extracting triplet constraints from them. Some pair-based sampling strategies have been proposed for acquire constraint [4–6]. For example, [4] suggests sampling the semi-hard negative examples. [5] employs the inclusion of all negative examples falling within the margin for each positive pair. [6] introduces distance weighted sampling, which involves sampling examples based on their distance from the anchor example.

Unlike pair-based methods, Proxy-based methods do not focus on the sample-to-sample relation, and hence avoid investigating sophisticated sampling strategies. The ProxyNCA loss [7] is one of the pioneering investigations that introduced this paradigm. It considers proxies as clustering centers in embedding space. It focuses on modeling the relationship between data instances and the proxies, resulting in a considerable reduction in computational load. The ProxyAnchor loss [8] is an improvement of the ProxyNCA loss. It employs a weighted optimization mechanism to adapt the intensity of optimization according to the similarity between the sample and the proxy. The Smooth ProxyAnchor loss [9] introduces a confidence module to mitigate the impact of the noisy labels in the data.

The above proxy-based methods use predefined representations to enforce discriminative representations on samples from different classes. Nevertheless, the variability observed in samples cannot solely be attributed to class attributes; it is also influenced by latent features such as viewpoint, postures, background, illumination, and other factors [10]. As a result, it becomes imperative to establish a more discernible representation that effectively captures differences beyond class-

* Xiao-Lei Zhang is the corresponding author.

irrelevant characteristics, thereby facilitating more refined instance retrieval.

To address the above issue, this study proposes a hierarchical multi-proxy method to enhance the generalization ability of the learned class-irrelevant features. Fundamentally, it assigns multiple sub-proxies to each class for representing the data distribution properly. Our main contributions are summarized as follows:

- We propose a novel hierarchical multi-proxy loss called dynamic main-proxy anchor (DMA) to handle both class-relevant and class-irrelevant characteristics.
- We propose a regularization term to ensure the proximity between sub-proxies from the same class, and keep the distance between sub-proxies from different class.
- We compare the proposed DMA with the state-of-the-art methods. Experimental results verify the effectiveness of the proposed method in both retrieval and clustering.

2. PROPOSED METHOD

2.1. Review of Proxy-Anchor Loss

The Proxy-Anchor loss [8] assigns a proxy to each class and then takes each proxy as an anchor, associating it with the entire data in a batch. The loss is given by:

$$
\begin{aligned}
L_{PA} = & \frac{1}{|P^+|} \sum_{p \in P^+} \log \left(1 + \sum_{x \in \chi_p^+} e^{-\alpha\left(x^T p - \delta\right)} \right) \\
& + \frac{1}{|P|} \sum_{p \in P} \log \left(1 + \sum_{x \in \chi_p^-} e^{\alpha\left(x^T p + \delta\right)} \right),
\end{aligned} \tag{1}
$$

where P denotes the set of all proxies and P^+ indicates the set of proxies in a batch; For each proxy p, the set of embedding vectors X is partitioned into two subsets: X_P^+ and X_P^-, which represent the set of positive and negative embedding vectors of p respectively; α is a scaling factor, and δ is a margin.

The Proxy-Anchor loss utilizes data-to-data relations during training, which is able to provide the embedding networks richer supervisory signals than other Proxy-based method. However, the way of assigning only one proxy to each class is difficult to capture intra-class features, which results in poor performance on the datasets that have large intra-class variances.

2.2. Dynamic Main-proxy Proxy-Anchor Loss

The proposed DMA method is designed to overcome the limit of the Proxy-Anchor loss mentioned above. Specifically, as shown in Fig. 1, it assigns multiple sub-proxies p_k ($\forall k = 1, 2, \ldots, K$) and one main-proxy p_m to each class, where the sub-proxies represent the intra-class variance, and the main-proxy is served for the inter-class distinction. We describe DMA in detail as follows.

Giving a data sample x_i, the similarity $s\left(x_i, p_k\right)$ between x_i and a sub-proxy p_k of an arbitrary class can be calculated as

$$
s\left(x_i, p_k\right) = x_i^T p_k, \tag{2}
$$

where the subscript k denotes the intra-class variability. At this point, it is not feasible to treat each sub-proxy as an anchor directly, because it cannot reflect the relationship between classes. Therefore, we construct the main-proxy p_m for each class by the weighted sum of the similarity scores between the sub-proxies p_k and x_i. The similarity between x_i and the main-proxy p_m of this class is formulated as

$$
s\left(x_i, p_m\right) = \sum_k w_{ik} x_i^T p_k, \tag{3}
$$

where

$$
w_{ik} = \frac{\exp\left(\frac{1}{\gamma} x_i^T p_k\right)}{\sum_k \exp\left(\frac{1}{\gamma} x_i^T p_k\right)} \tag{4}
$$

is normalized similarity factor, and γ is the temperature. From Eq. (3) we can see that, the main-proxy is

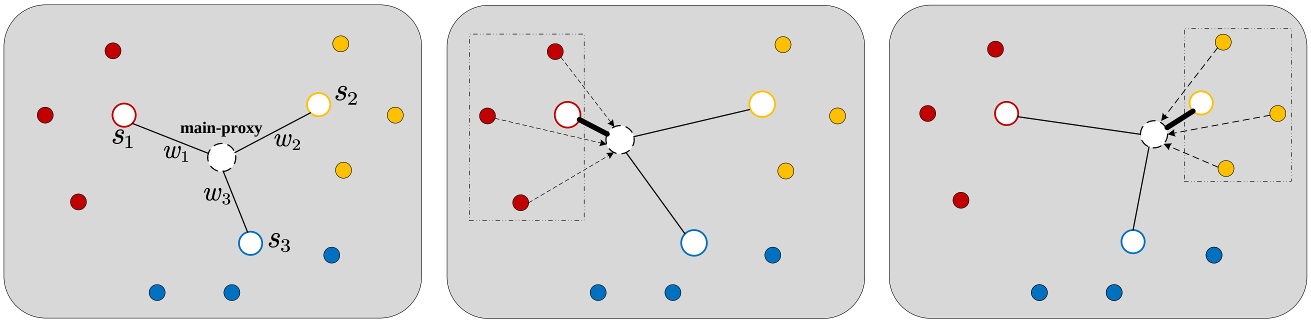
$$
p_m = \sum_k w_{ik} p_k, \tag{5}
$$

which is depended by both the sub-proxies and the sample x_i. Finally, substitute Eq. (3) into Eq. (1), the proposed DMA loss can be formulated as

$$
\begin{aligned}
L_m = & \frac{1}{|P_M^+|} \sum_{p_m \in P_M^+} \log \left(1 + \sum_{x_i \in \chi_{p_m}^+} e^{-\alpha\left(\sum_k w_{ik} x_i^T p_k - \delta\right)} \right) \\
& + \frac{1}{|P_M|} \sum_{p_m \in P_M} \log \left(1 + \sum_{x_i \in \chi_{p_m}^-} e^{\alpha\left(\sum_k w_{ik} x_i^T p_k + \delta\right)} \right),
\end{aligned} \tag{6}
$$

where P_M and P_M^+ denote the set of all proxies and the set of proxies in a batch respectively. The loss takes the main-proxy as an anchor, and uses it together with a sample from the same class to form a positive pair, and with a sample from a different class to form a negative pair. This formulation ensures the embedding vectors with similar inter-class features as close as possible in the embedding space.

From the above formulation, we can see that, unlike existing hierarchical structure method, such as [10], the main-proxy in our method is constructed dynamically for each sample, which not only explores the advantage of the Proxy-Anchor loss, but also reduces the computational complexity compared to [10].



(a) (b) (c)

Fig. 1. Illustration of the proposed DMA loss. Different colors represent different intra-class features of one class. Solid circles are data samples, while hollow circles are proxies. (a) The dynamic main-proxy is determined by all the sub-proxies s_1-s_3 and weight factors w_1-w_3. (b) For intra-class feature "red", as the weight factor w_1 increases, the main-proxy moves closer to the direction of sub-proxy s_1. At this point, the main-proxy, as an anchor, will pull these samples closer to s_1. (c) Similarly, the main-proxy pulls the data samples with intra-class feature "yellow" to s2.

2.3. Regularization on Sub-proxies

The sub-proxies, which serve as local cluster centers in each class, represent the intra-class variability. To ensure the proximity between similar sub-proxies and the distance between dissimilar ones, we apply a constraint on the sub-proxies. Specifically, we regard each sub-proxy as a sample, and establish positive/negative pairs with the main-proxy anchors from the same/different classes. To reduce the computational complexity, we designate the main-proxy in the regularization term as the center of all sub-proxies in the same class:

$$p_{m2} = \sum_k \mu p_k, \tag{7}$$

where μ is a scalar. Similarly, the constraint can be formulated as

$$\begin{aligned} L_p = & \frac{1}{|P^+_{M2}|} \sum_{p_{m2} \in P^+_{M2}} \log \left(1 + \sum_{x_i \in \chi^+_{p_{m2}}} e^{-\alpha\left(p_k^T p_{m2} - \delta\right)} \right) \\ & + \frac{1}{|P_{M2}|} \sum_{p_{m2} \in P_{M2}} \log \left(1 + \sum_{x_i \in \chi^-_{p_{m2}}} e^{\alpha\left(p_k^T p_{m2} + \delta\right)} \right). \end{aligned} \tag{8}$$

Finally, the overall objective of DMA becomes

$$L = L_m + \lambda L_p, \tag{9}$$

where $\lambda > 0$ is a trade-off hyper-parameter.

3. EXPERIMENTS

3.1. Datasets and Experiment Setting

We conducted experiments on three standard datasets. The CUB-200-2011 (CUB) [11] dataset consists of 11,788 images of 200 bird species. We used the first 100 classes for training and the remaining 100 classes for testing. The Cars196 (Cars) [12] dataset consists of 16185 images of 196 bird species. We used the first 98 classes for training and the remaining 98 classes for testing. The Stanford Online Products (SOP) [5] dataset consists of 120,053 images of 22,634 online products. We used the first 11,318 classes for training and the remaining 11,316 classes for testing.

We leveraged the commonly used Resnet50 [13] model pre-trained on ImageNet [14] as our backbone. All input images were resized to 224×224. The model was optimized by Adam with 50 epochs. The batch size was set to 180. The learning rate for the network parameters was set to 10^{-4} on the CUB-200-2011 and Cars-196, and 6×10^{-4} on the SOP. To accelerate convergence, the learning rate for proxies was scaled up 100 times. The input batches were randomly sampled during training. The number of sub-proxies N was set to 10 for the CUB-200-2011 and Cars-196, and 2 for the SOP. The temperature γ was set to 0.1.

We conducted a broad comparison on the tasks of image retrieval which is evaluated by the Recall@k metric [5], as well as clustering which is evaluated by the Normalized Mutual Information (NMI) [15] respectively.

3.2. Comparison with Other Methods

We compare the proposed method with two categories of methods: classical deep metric learning methods [4, 7, 8, 16–22] and some recently published methods [23–26].

Table 1 shows the results of the comparison methods on image retrieval. From the table, we can see that the proposed method achieves the top performance in various R@k metrics on all three datasets. Specifically, the proposed method achieves the best scores on both the CUB and Cars datasets,

Table 1. Recall@K(%) performance on CUB, Cars and SOP in image retrieval. Some recent methods are marked by the superscript "$*$". The top two methods are highlighted in red and blue colors, respectively.

Method	CUB				Cars				SOP			
	R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8	R@1	R@10	R@100	R@1000
Triplet [4]	42.5	55	66.4	77.2	51.5	63.8	73.5	82.4	66.7	82.4	91.9	-
Npairs [16]	51.9	64.3	74.9	83.2	68.9	78.9	85.8	90.9	66.4	82.9	92.1	-
Angular Loss [17]	54.7	66.3	76.0	83.9	71.4	81.4	87.5	92.1	70.9	85.0	93.5	98.0
Proxy-NCA [7]	49.2	61.9	67.9	72.4	73.2	82.4	86.4	88.7	73.7	-	-	-
Normalized Softmax [18]	59.6	72.0	81.2	88.4	81.7	88.9	93.4	96.0	73.8	88.1	95.0	-
RLL-H [19]	57.4	69.7	79.2	86.9	74.0	83.6	90.1	94.1	76.1	89.1	95.4	-
Multi-similarity [20]	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5	78.2	90.5	96.0	98.7
SoftTriple [21]	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9	78.3	90.3	95.9	-
Proxy Anchor [8]	68.4	79.2	86.8	91.6	86.1	91.7	94.5	96.9	79.1	90.8	96.2	98.7
Proxy-GML [22]	66.6	77.6	86.4	-	85.5	91.8	95.3	-	78.0	90.6	96.2	-
DANML* [23]	67.6	79.1	86.4	91.2	85.6	92.1	94.1	97.7	79.9	92.1	96.4	98.9
RS-Topnk-MS* [24]	67.8	78.7	86.8	92.1	85.2	90.9	94.5	96.9	79.0	91.3	96.8	-
MS + DAS* [25]	69.2	79.2	87.1	92.6	87.8	93.1	95.6	97.8	80.5	91.8	96.7	98.9
MHP + Proxy Anchor* [26]	69.8	79.8	87.1	92.1	87.4	92.5	95.4	97.7	79.7	91.2	96.4	98.9
DMA (ours)	70.3	80.4	87.7	92.8	88.2	93.0	95.8	97.8	80.4	91.8	96.8	98.9

Table 2. NMI performance on CUB, Cars and SOP in clustering. The recent method is marked by the superscript "$*$". The top two methods are highlighted in red and blue colors, respectively.

Method	NMI		
	CUB	Cars	SOP
Triplet [4]	55.3	53.4	89.5
Npairs [16]	60.2	62.7	87.9
Angular Loss [17]	66.1	63.2	88.6
Proxy-NCA [7]	59.5	64.9	90.6
Normalized Softmax [18]	66.2	70.5	89.8
RLL-H [19]	63.6	65.4	89.7
DCES [27]	69.6	70.3	90.2
MIC [28]	69.7	68.4	90.0
Proxy-GML [22]	69.8	72.4	90.2
MS + DAS* [25]	69.1	70.8	90.4
DMA (ours)	72.8	74.1	90.6

except for the R@2 on the Cars dataset, which is inferior only to MS+DAS [25]. The proposed method obtains the runner-up performances in terms of R@1 and R@10 on the SOP, falling behind methods MS+DAS [25] and DANML [23] respectively. The main reason for this suboptimal performance is that the dataset contains a large number of classes (11318 classes), and has a low intra-class variance (each class only has an average of 5 images), which contradicts the benefits of the multi-proxy strategy. However, the gap between them is not obvious, and the proposed method can still be competitive with the mainstream methods in R@100 and R@1000.

Table 2 lists the clustering results of the comparison methods. From the table we see that, the proposed method achieves the highest NMI in all three datasets. For example, for CUB and Cars, the proposed method gets a score of 72.8 and 74.1, which is 3 percentage points (pp) and 1.7pp better than the runner-up methods respectively. For SOP, the proposed method achieves the best score of 90.6, which is the same as Proxy-NCA [7] and is 0.2pp higher than the runner-up method.

Table 3. Impact of regularization term

Datasets	L_p	R@1	R@2	R@4	R@8	NMI
CUB	✗	69.5	79.7	87.2	92.5	72.2
	✓	70.3	80.4	87.7	92.8	72.8
Cars	✗	87.5	92.7	95.4	97.5	73.3
	✓	88.2	93.0	95.8	97.8	74.1

3.3. Ablation Studies

To verify the effectiveness of the regularization term in the proposed loss function, we conducted an ablation study on CUB and Cars. Experimental results in Table 3 show that the regularization term L_p can improve the overall performance in both R@k and NMI metrics, which implies that it can regularize the global geometry among sub-proxies, thereby enabling the learning of more discriminative sub-proxies.

4. CONCLUSION

In this paper, we propose DMA loss to overcome the limitations of traditional single-proxy methods in capturing the intra-class features. The DMA is a hierarchical multi-proxy structure which allocates multiple sub-proxies to acquire diverse intra-class characteristics before employing the dynamically constructed main-proxy to handle class-related attributes. The experiment results on CUB, Cars and SOP demonstrate its superior performance to the state-of-the-art methods in the tasks of image retrieval and clustering.

5. REFERENCES

[1] Sumit Chopra, Raia Hadsell, and Raia Hadsell, “Learning a similarity metric discriminatively, with application to face verification,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, 2005, vol. 1, pp. 539–546.

[2] Raia Hadsell, Sumit Chopra, and Yann LeCun, “Dimensionality reduction by learning an invariant mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, 2006, vol. 2, pp. 1735–1742.

[3] Elad Hoffer and Nir Ailon, “Deep metric learning using triplet network,” in *Lecture Notes in Computer Science*, 2015, vol. 9370, pp. 84–92.

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[5] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese, “Deep metric learning via lifted structured feature embedding,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4004–4012.

[6] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl, “Sampling matters in deep embedding learning,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2859–2867.

[7] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh, “No fuss distance metric learning using proxies,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 360–368.

[8] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak, “Proxy anchor loss for deep metric learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3235–3244.

[9] Carlos Roig, David Varas, Issey Masuda, Juan Carlos Riveiro, and Elisenda Bou, “Smooth proxy-anchor loss for noisy metric learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Mingda Wang, Canqian Yang, and Yi Xu, “Hierarchical multiple proxy loss for deep metric learning,” *Digital Signal Processing*, vol. 133, pp. 103826, 2023.

[11] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” 2011.

[12] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei, “3D Object Representations for Fine-Grained Categorization,” in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 554–561.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[15] Christopher, Prabhakar Manning, Hinrich Raghavan, and Schütze, *Introduction to information retrieval*, Natural Language Engineering, 2010.

[16] Kihyuk Sohn, “Improved deep metric learning with multi-class n-pair loss objective,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, p. 1857–1865.

[17] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin, “Deep metric learning with angular loss,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2612–2620.

[18] Andrew Zhai and Hao-Yu Wu, “Classification is a strong baseline for deep metric learning,” in *British Machine Vision Conference*, 2018.

[19] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M. Robertson, “Ranked list loss for deep metric learning,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5202–5211.

[20] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott, “Multi-similarity loss with general pair weighting for deep metric learning,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5017–5025.

[21] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Tacoma Tacoma, Hao Li, and Rong Jin, “Softtriple loss: Deep metric learning without triplet sampling,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6449–6457.

[22] Yuehua Zhu, Muli Yang, Cheng Deng, and Wei Liu, “Fewer is more: A deep graph metric learning perspective using fewer proxies,” 2020, NIPS’20.

[23] Kun Song, Junwei Han, Gong Cheng, Jiwen Lu, and Feiping Nie, “Adaptive neighborhood metric learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4591–4604, 2022.

[24] Jian Wang, Xinyue Li, Zhichao Zhang, Wei Song, and Weiqi Guo, “Ranked similarity weighting and top-nk sampling in deep metric learning,” *IEEE Transactions on Multimedia*, pp. 1–10, 2022.

[25] Lizhao Liu, Shangxin Huang, Zhuangwei Zhuang, Ran Yang, Mingkui Tan, and Yaowei Wang, “Das: Densely-anchored sampling for deep metric learning,” in *ECCV*, 2022, pp. 399–417.

[26] Jian Wang, Xinyue Li, Wei Song, Zhichao Zhang, and Weiqi Guo, “Multi-hierarchy proxy structure for deep metric learning,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1645–1649.

[27] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Büchler, and Björn Ommer, “Divide and conquer the embedding space for metric learning,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 471–480.

[28] Biagio Brattoli, Karsten Roth, and Bjorn Ommer, “Mic: Mining interclass characteristics for improved metric learning,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7999–8008.