Towards Interpretable Deep Metric Learning with Structural Matching Supplementary Material

A. Implementation of DIML

A.1. The Sinkhorn Algorithm

The Sinkhorn algorithm [1] modifies the original optimal transport problem (Eq.4) into the following one:

$$T^* = \underset{T\geq 0}{\arg\min}\, \mathrm{tr}(CT^\top) + \lambda \mathrm{tr}\left(T(\log(T) - \mathbf{1}\mathbf{1}^\top)^\top\right), \\ \text{subject to} \quad T\mathbf{1} = \mu^{\mathrm{s}}, \quad T^\top\mathbf{1} = \mu^{\mathrm{t}}, \tag{A.1}$$

where λ is a non-negative regularization parameter. By adding the entropic regularizer, the Equation (A.1) becomes a convex problem, which can be solved with Sinkhorn-Knopp algorithm [12]. Starting from an initial matrix $K = \exp(-C/\lambda)$, the problem can be solved by iteratively projecting onto the marginal constraints until convergence:

$$\boldsymbol{a} \leftarrow \mu^{\mathrm{s}}/K\boldsymbol{b}, \quad \boldsymbol{b} \leftarrow \mu^{\mathrm{t}}/K^\top\boldsymbol{a}. \tag{A.2}$$

After converged, we can obtain the optimal transport plan:

$$T^* = \mathrm{diag}(\boldsymbol{a})K\mathrm{diag}(\boldsymbol{b}). \tag{A.3}$$

A.2. Testing

In all of our experiments, we use ResNet50 [6] as our backbone. Therefore, the size of the feature map before the pooling layer is 7×7. To reduce computational costs, we first use ROI Align [5] to pool the feature map to $G \times G$ and $G = 4$ in most of our experiments unless otherwise noted. According to the multi-scale matching algorithm, for each image as a query, we first sort the images in the gallery using the standard cosine similarity to obtain the indices of top-K candidates $\mathcal{I}_K$ (we use $K = 100$ in most of the experiments). We then calculate the proposed structural similarity of all the images in $\mathcal{I}_K$. To combine both global and structural information, we use the sum of the cosine similarity and the structural similarity for the top-K images to compute their ranks. The regularization parameter λ in Equation (6) is set to 0.05.

A.3. Training

Incorporating DIML into the training objectives is quite straightforward. Generally, the loss functions in metric learning can be roughly categorized into distance-based methods (*e.g.*, Contrastive [4], Triplet [3], Margin [16]) and similarity-based methods (*e.g.*, Multi-Similarity [15], Arcface [2], N-Pair [13]) For distance-based methods, we replace the original distance function d with the average of d and our structural distance d_{struct}; For similarity-based methods, we replace the original similarity function s with the average of s and our structural similarity s_{struct}. In this section, we will use several loss functions as examples to demonstrate how to apply DIML during training.

Margin [16] The Margin loss [16] is defined as

$$\mathcal{L}_{\text{margin}}(k, l) = \left(\sigma + (-1)^{I(y^k \neq y^l)}\left(D_{k,l} - \beta\right)\right)_+, \tag{A.4}$$

where σ and β are learnable parameters, and D_{kl} is used to measure the distance between image k and l:

$$D_{k,l} = \frac{1}{2}\left(d_{\text{struct}}(z^k, z^l) + d(\bar{z}^k, \bar{z}^l)\right), \tag{A.5}$$

where d is Euclid distance and d_{struct} is derived from d using Equation (10).

Multi-Similarity [15] The original Multi-Similarity is defined as:

$$s^*(k,l) = \begin{cases} s\left(k,l\right), & s\left(k,l\right) > \min_{p\in\mathcal{P}_k} s\left(k,p\right) - \epsilon \\ s\left(k,l\right), & s\left(k,l\right) < \max_{n\in\mathcal{N}_k} s\left(k,n\right) + \epsilon \\ 0, & \text{otherwise} \end{cases}, \tag{A.6}$$

$$\mathcal{L}_{\text{MS}} = \frac{1}{B}\sum_{k\in\mathcal{B}}\left[\frac{1}{\alpha}\log\left[1 + \sum_{p\in\mathcal{P}_k}\exp\left(-\alpha\left(s^*\left(k,p\right) - \lambda\right)\right)\right]\right. \\ \left. + \frac{1}{\beta}\log\left[1 + \sum_{n\in\mathcal{N}_k}\exp\left(\beta\left(s^*\left(k,n\right) - \lambda\right)\right)\right]\right], \tag{A.7}$$

where $s(k,l) = s(\psi^k, \psi^l)$ is the cosine similarity of the embeddings ψ^k, ψ^l of the two images. To utilize DIML, we can replace s with

$$s(k,l) \leftarrow \frac{1}{2}\left(s(\bar{z}^k, \bar{z}^l) + s_{\,\text{struct}}(z^k, z^l)\right). \tag{A.8}$$

Note that in our notation both ψ^k and $\bar{z}^k$ represent the same embedding in $\mathbb{R}^D$.

ProxyNCA [8] It is also worth mentioning there are slight difference when applying DIML to proxy-based methods during training. Taking ProxyNCA [8] as example, the original objective is

$$\mathcal{L}_{\text{proxy}} = -\frac{1}{B}\sum_{k\in\mathcal{B}} \log\left(\frac{\exp\left(-d\left(\psi^k, \eta^{y^k}\right)\right)}{\sum_{c\in\mathcal{C}\setminus\{y^k\}} \exp\left(-d\left(\psi^k, \eta^c\right)\right)}\right), \tag{A.9}$$

where d is Euclid distance and $\eta^c \in \mathbb{R}^D$ is the proxy for the c-th class. To use DIML, we need to use proxies with the size $\mathbb{R}^{H\times W\times D}$, denoted as $\{\rho^c, c\in\mathcal{C}\}$. Then, we can replace the $d(\psi^k, \eta^c)$ with

$$d(\psi^k, \eta^c) \leftarrow \frac{1}{2}\left(d(\psi^k, \eta^c) + d_{\,\text{struct}}(z^k, \rho^c)\right), \tag{A.10}$$

where we also note that $\text{GAP}(\rho^c) = \eta^c$.

B. Experimental Details

B.1. Evaluation Metrics

We implement the same evaluation metrics as [9], including Precision at 1 (P@1), R-Precision (RP), and Mean Average Precision at R (MAP@R).

P@1 is also known as Recall@1 in metric learning. Given a sample x^q and feature encoder $\phi(\cdot)$, the set of k nearest neighbors of x^q is calculated as the precision of k nearest neighbors:

$$\mathcal{N}_q^k = \underset{\mathcal{N}\subset\mathcal{X}_{\text{test}}, |\mathcal{N}|=k}{\arg\min} \sum_{x^f\in\mathcal{N}} d_e(\phi(x^q), \phi(x^f)) \tag{B.1}$$

where $d_e(\cdot,\cdot)$ is the euclidean distance. Then P@k can be measured as

$$\text{P@}k = \frac{1}{|\mathcal{X}_{\text{test}}|}\sum_{x_q\in\mathcal{X}_{\text{test}}} \frac{1}{k}\sum_{x^i\in\mathcal{N}_q^k} \begin{cases} 1, & y^i = y^q, \\ 0, & \text{otherwise} \end{cases}, \tag{B.2}$$

where y^i is the class label of sample x^i. We only report P@1 in our experiments, i.e. $k=1$.

R-precision is defined in [9]. Specifically, for each sample x^q, let R be the number of images that are the same class with x^q and R-precision is simply defined as P@R (see Equation B.2). However, R-precision does not consider the ranking of correct retrievals, so it is not informative enough. To tackle this problem, [9] introduced Mean Average Precision at R.

MAP@R is similar to mean average precision, but limit the number of nearest neighbors to R. So it replaces *precision* in MAP calculation with *R-precision*:

$$\text{MAP@}R = \frac{1}{R}\sum_{i=1}^{R} P(i), \tag{B.3}$$

where

$$P(i) = \begin{cases} \text{P@}i, & \text{if the } i\text{-th retrieval is correct;} \\ 0, & \text{otherwise.} \end{cases} \tag{B.4}$$

MAP@R is more informative than P@1 and it can be computed directly from the embedding space without clustering as post-processing.

B.2. Experimental Setups

For most of the baseline methods, we follow the implementation and the hyper-parameters in [14]. For Proxy Anchor [7], we use their original implementation but set the hyper-parameters as [14] (batch size 112, embedding size 128, *etc*.). Besides various loss functions, we also experiment with different sampling methods. In Table 1 of the original paper, we use suffixes to represent the sampling methods (-R: Random; -D: Distance [16]; -S Semihard [11]; -H: Softhard [10]).

C. Detailed Results

In the original paper, we have demonstrated the effects of truncation number K and feature map size G using charts. In this section, we provide the original numerical results that were used to plot those charts in Table 1 and Table 2.

References

[1] Marco Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *NIPS*, volume 2, page 4, 2013. 1

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 1

[3] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *ECCV*, pages 269–285, 2018. 1

[4] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742, 2006. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

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

Table 1: **Comparisons of different truncation numbers.** We test for different truncation number K ranging from 0 to 500. Experimental results show that a small K can already bring considerable performance improvement.

Baseline	K	**CUB-200**			**Cars196**			**SOP**		
		P@1	**RP**	**M@R**	**P@1**	**RP**	**M@R**	**P@1**	**RP**	**M@R**
Margin[16]	0	62.47	34.12	23.14	72.18	32.00	20.82	78.39	45.64	42.34
	10	**65.16**	34.56	23.87	**76.65**	32.52	21.72	**79.26**	**46.44**	**43.20**
	50	65.16	35.43	**24.54**	76.65	33.64	22.83	79.26	46.44	43.19
	100	65.16	**35.48**	24.54	76.65	**33.93**	**22.95**	79.26	46.44	43.19
	500	65.16	35.48	24.54	76.65	33.93	22.95	79.26	46.44	43.19
Multi-Similarity[15]	0	62.56	32.74	21.99	74.81	32.72	21.60	77.90	44.97	41.54
	10	**64.89**	33.21	22.73	**78.50**	33.26	22.50	**78.53**	**45.60**	**42.24**
	50	64.89	34.04	23.37	78.50	34.46	23.72	78.53	45.60	42.23
	100	64.89	**34.12**	**23.38**	78.50	**34.70**	**23.81**	78.53	45.60	42.23
	500	64.89	34.12	23.38	78.50	34.70	23.81	78.53	45.60	42.23

Table 2: **Effects of the size of feature map.** Generally, the performance of our DIMLis better with higher G. DIML with $G = 4$ yields good results within relatively low computational costs.

Baseline	G	**CUB-200**			**Cars196**			**SOP**		
		P@1	**RP**	**M@R**	**P@1**	**RP**	**M@R**	**P@1**	**RP**	**M@R**
Margin[16]	1	62.47	34.12	23.14	72.18	32.00	20.82	78.39	45.64	42.34
	2	64.15	34.79	23.83	75.04	32.59	21.85	79.06	46.29	43.03
	4	65.16	35.48	24.54	76.65	**33.93**	**22.95**	79.26	46.44	43.19
	7	**65.58**	**35.58**	**24.79**	**76.96**	32.93	22.66	**79.59**	**46.83**	**43.62**
Multi-Similarity [15]	1	62.56	32.74	21.99	74.81	32.72	21.60	77.90	44.97	41.54
	2	63.77	33.33	22.60	77.45	33.25	22.60	78.39	45.56	42.15
	4	64.89	34.12	23.38	78.50	**34.70**	**23.81**	78.53	45.60	42.23
	7	**65.45**	**34.15**	**23.55**	**78.93**	33.64	23.50	**78.76**	**45.90**	**42.57**

[8] 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. 2

[9] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, pages 681–699. Springer, 2020. 2

[10] 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. PMLR, 2020. 2

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

[12] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967. 1

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

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 2

[15] 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. 1, 3

[16] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, pages 2840–2848, 2017. 1, 2, 3