Supplementary Material: Hierarchical Proxy-based Loss for Deep Metric Learning

Zhibo Yang[1*], Muhammet Bastan[2], Xinliang Zhu[2], Doug Gray[2], Dimitris Samaras[1]
[1]Stony Brook University, [2]Visual Search & AR, Amazon

Contents

1. Implementation and Evaluation Details

In this paper, we adopt two evaluation protocols to evaluate our method. The first evaluation protocol is adopted in many previous metric learning papers [1, 6, 9, 12] where each dataset is split into a training set (first half of classes) and a testing set (second half of classes) and best results on the testing set are reported; The second evaluation protocol was recently proposed in [10] featuring a more standardized and fairer evaluation procedure. Based on the evaluation protocol, the implementation (i.e., hyper-parameters selection) is slightly different. Note that like other methods (e.g., Proxy Anchor [15]) embeddings and proxies are L2 normalized in our method. All code is implemented using PyTorch [11].

Under the first evaluation protocol, we test our hierarchical proxy-based loss (HPL) with two backbone networks—ResNet-50 [4] and BN-Inception [5] (see main paper for results with ResNet-50 and Sec. 2.1 for results with BN-Inception). Following [6], we append a max pooling layer in parallel with an average pooling layer to the penultimate layer of the backbone network. The outputs from the max pooling layer and the average pooling layer are added and then passed through a embedding layer (i.e., fully-connected layer) which projects the network output to a desired embedding space. The embeddings are 512-dimensional and L2 normalized before computing the loss, and during inference. Both backbone networks are pre-trained on ImageNet [2] for the classification task. The batch size is set to 128 during training. Each model is trained for 30 epochs with learning rate 10^{-4} using the AdamW optimizer [7]. We train the baseline models with both Proxy-NCA loss and Proxy Anchor loss by following the standard hyper-parameters settings in [9] and [6], respectively. Namely, the scaling factor and margin for Proxy Anchor loss is set as 32 and 0.1, respectively. For our HPL loss, the same hyper-parameters are used across all levels of proxies.

The second evaluation protocol [10] splits each dataset into a trainval set (first half of classes) and a testing set (second half of classes) as the first evaluation protocol. However, to mitigate the effect of hyper-parameter selections in different methods, the trainval set is further split into four equal-sized partitions for cross validation. Particularly, three partitions are used as the training set and the remaining partition is used as the validation set. Hyper-parameters are selected by optimizing the average validation performance (MAP@R) of four leave-one-out experiments. This hyper-parameter optimization is done by running M iterations of Bayesian Optimization ($M = 50$ for the CUB and Cars-196 datasets and $M = 10$ for the SOP dataset due to its high complexity). The hyper-parameters to be optimized in HPL include the learning rate of the proxies; the number of coarse proxies; the update frequency of the coarse proxies; and scaling factors and margins (if applicable) for each proxy level. The model parameters are trained using RMSprop with learning rate 10^{-6} and batch size 128. The training terminates when the validation accuracy plateaus. We refer the reader to [10] for further details of the evaluation protocol.

2. Additional Experimental Results

2.1. Full Results and Further Discussion

In this section, we present the full experimental results of Tab. 4 and Tab. 5 in the main paper which was partially presented due to the page limit. Table 1-2 and Table 3-4 present the full results of Tab. 4 and Tab. 5 in the main paper, respectively. One can see from Table 1 and Table 2 that our proposed HPL outperforms the baselines in all Recall@K. This further demonstrates the effectiveness of our HPL losses. We observe consistent improvement over the standard proxy-based losses across different datasets (i.e., In-Shop, SOP and iNaturalist). Moreover, the results in Table 2 also show that our HPL surpasses several

	In-Shop					
	R@1	R@10	R@20	R@30	R@40	R@50
Proxy-NCA	87.21	96.57	97.63	98.12	98.34	98.49
HPL-NCA	**88.70**	**96.83**	**97.97**	**98.38**	**98.61**	**98.79**
Proxy Anchor	89.85	97.14	97.94	98.33	98.52	98.73
HPL-PA	**92.46**	**97.97**	**98.57**	**98.92**	**99.11**	**99.18**

Table 1. **Recall@K (%) on the In-Shop dataset**. ResNet-50 is used as the backbone and we set $|P_1| = 500$.

	SOP			
	R@1	R@10	R@100	R@1000
HTL*	74.8	-	-	-
D&C*	75.9	-	-	-
MIC*	77.2	-	-	-
DiVA*	79.6	-	-	-
Proxy-NCA	77.6	89.3	94.6	97.7
HPL-NCA	**80.1**	**91.1**	**96.1**	**98.6**
Proxy Anchor	79.4	90.4	95.7	98.5
HPL-PA	**80.0**	**91.1**	**96.3**	**98.8**

Table 2. **Recall@K (%) on the SOP dataset**. ResNet-50 is used as the backbone and we set $|P_1| = 500$ for HPL-NCA and HPL-PA. * indicates that the results are directly taken from [8].

similar methods which either try to utilize the hierarchical data structure or model class-shared information including HTL[3], D&C[14], MIC [13] and DiVA [8] (a more detailed description of these methods can be found in Sec. 2 of the main paper).

In addition, we include additional results when $|P_1| = 500$ for the SOP dataset Table 3. One can see from Table 3 and Table 4 that both HPL-NCA and HPL-NCA-GT (i.e., HPL-NCA with ground-truth hierarchy) outperform Proxy-NCA, and surprisingly, HPL-NCA surpasses HPL-NCA-GT even when given the same number of coarse proxies. This shows that the hierarchy of categories learned by our methods via online clustering performs better than directly using the human-curated hierarchy of categories in terms of image retrieval accuracy. We think this might be caused by the fact that the human-curated category hierarchy may not fully reflect the visual similarity among classes, whereas, our method automatically learns the hierarchy based on visual similarity between classes, making it more favorable for metric learning. In the iNaturalist dataset, the composition of a genus (i.e., coarse classes) is not fully determined by the appearance similarity among species (i.e., fine classes), and the species within the same genus can look very different (e.g., a fennec fox and a arctic fox in Fig. 1). Therefore, directly using the human-curated hierarchy could make the class-shared signals weak and noisy.

Fennec Fox Arctic Fox

Figure 1. Examples of the fennec fox (left) and the arctic fox (right) in the iNaturalist Dataset.

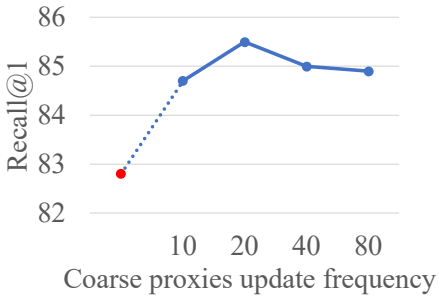

Figure 2. **Impact of the updating frequency of coarse proxy**. We train the models with HPL-NCA loss and report the Recall@1 on the Cars-196 dataset of models that are trained with different values of the hyperparameter T. Red dot represents the traditional Proxy-NCA.

In contrast, our method groups the species based on their visual similarities (i.e., proxies) in an unsupervised manner. This makes the commonalities among species stronger and easier to capture using the loss on the coarse proxies.

In Fig. 2, we further evaluate the impact of the hyper-

	SOP				
	$\|P_1\|$	R@1	R@10	R@100	R@1000
Proxy-NCA	-	77.63	89.29	94.58	97.67
HPL-NCA-GT	12	78.69	90.44	95.72	98.47
HPL-NCA	12	79.33	90.71	95.63	98.35
HPL-NCA	100	79.83	91.06	96.07	**98.63**
HPL-NCA	500	**80.13**	**91.07**	**96.11**	98.62

Table 3. **Learned hierarchy vs human-curated hierarchy on SOP dataset**. HPL-NCA-GT denotes PL-NCA with ground-truth class hierarchy.

	iNaturalist				
	$\|P_1\|$	R@1	R@2	R@4	R@8
Proxy-NCA	-	51.32	62.56	72.53	80.80
HPL-NCA-GT	48	51.63	63.00	72.77	81.12
HPL-NCA	48	51.95	63.18	73.04	81.42
HPL-NCA	500	**52.26**	**63.53**	**73.47**	**81.62**

Table 4. **Learned hierarchy vs human-curated hierarchy on iNaturalist dataset**. HPL-NCA-GT denotes PL-NCA with ground-truth class hierarchy.

parameter T—the updating frequency of coarse proxy T in our method. In particular, we evaluate HPL-NCA with 10 coarse proxies using different coarse proxies update frequencies $T \in \{10, 20, 40, 80\}$ on Cars-196. The Recall@1 are 84.7%, 85.5%, 85.0% and 84.9%, respectively. As a baseline, traditional Proxy-NCA has a Recall@1 of 82.8% in our experiments. This shows that our method is robust to different choices of T and outperforms the baseline consistently.

2.2. Qualitative Results

Fig. 3 showcases some image retrieval results of the models trained with our HPL-NCA loss and standard Proxy-NCA loss. One can see that the quality of the retrieval results returned by our model is better than that of the model trained with standard Proxy-NCA loss. Noticeably, our top-4 matches all share the same categories as the queries, while standard Proxy-NCA model sometimes returns out-of-categories matches despite that the matches sharing some similarities as the query images. This demonstrates the effectiveness of having a hierarchical structure in the proxies where features shared among similar classes can be learned.

References

[1] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *European Conference on Computer Vision*, pages 677–694. Springer, 2020.

[2] 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*, pages 248–255. Ieee, 2009.

[3] Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R Scott. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision*, pages 269–285, 2018.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[6] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3238–3247, 2020.

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[8] 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 *European Conference on Computer Vision*, pages 590–607. Springer, 2020.

[9] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

[10] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020.

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[12] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6450–6458, 2019.

[13] Karsten Roth, Biagio Brattoli, and Bjorn Ommer. Mic: Mining interclass characteristics for improved metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8000–8009, 2019.

[14] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 471–480, 2019.

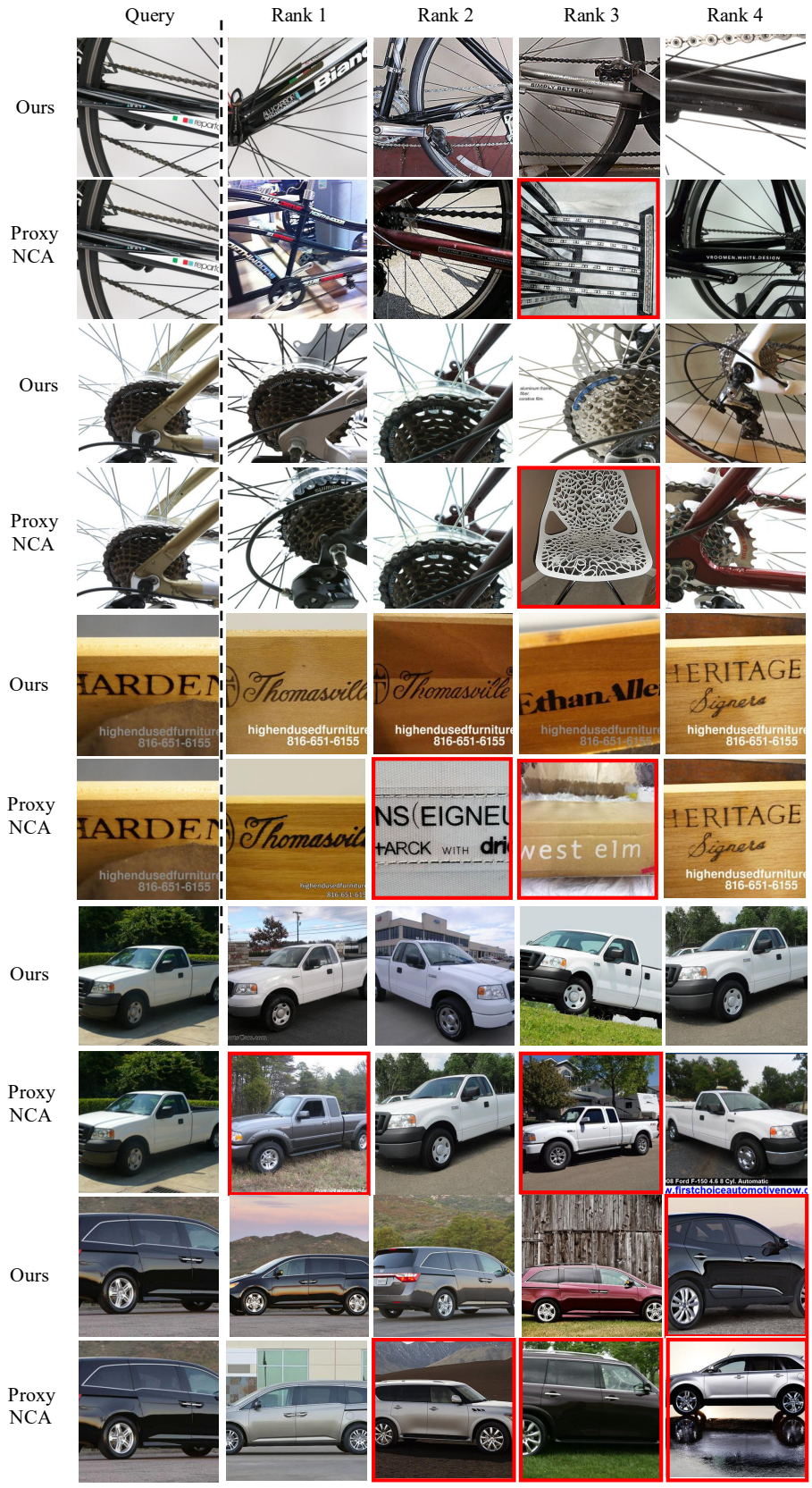

Figure 3. Additional qualitative results on SOP and Cars-196. We present the query images on the leftmost, and the top-4 matches on the right. Odd rows are the results of our HPL-NCA loss; even rows are the results from the Proxy-NCA loss. Red box indicates incorrect matches.