Supplementary Material for paper "Meta Batch-Instance Normalization for Generalizable Person Re-Identification"

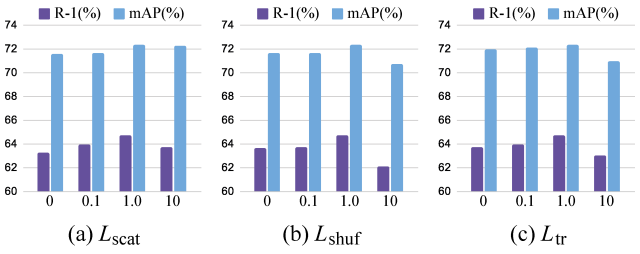
Seokeon Choi Taekyung Kim Minki Jeong Hyoungseob Park Changick Kim
Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{seokeon, tkkim93, rhm033, hyoungseob, changick}@kaist.ac.kr

A. More Analysis on Hyperparameters

Meta-train loss: The total meta-train loss can be reformulated as $\mathcal{L}_{\text{mtr}} = \lambda_{\text{scat}}\mathcal{L}_{\text{scat}} + \lambda_{\text{shuf}}\mathcal{L}_{\text{shuf}} + \lambda_{\text{tr}}\mathcal{L}_{\text{tr}}$. We compare the performance by changing each of these weights, as expressed in Fig. S1. We observed the highest performance when each weight parameter was 1.0. In addition, when we assign each weight parameter to 0 (*i.e.* remove the corresponding loss), its performance deteriorates, which proves that all loss components are essential. Note that when the weights for the inter-domain shuffle loss $\mathcal{L}_{\text{shuf}}$ and the triplet loss $\mathcal{L}_{\text{tr}}$ increase, their performances can be even lower. Thus, it is important to balance the weight parameters. For a more detailed analysis, we investigate the changes in the meta-train losses during the training process, as described in Fig. S2. At the beginning of training, whereas the intra-domain scatter loss $\mathcal{L}_{\text{scat}}$ and the inter-domain shuffle loss $\mathcal{L}_{\text{shuf}}$ increase rapidly, the triplet loss $\mathcal{L}_{\text{tr}}$ decreases dramatically. It shows that the model initially focuses on improving discrimination power. Unlike in the early stage of training, we observed a tendency for all three types of losses to decrease simultaneously. It means that our model becomes generalized as learning progresses.

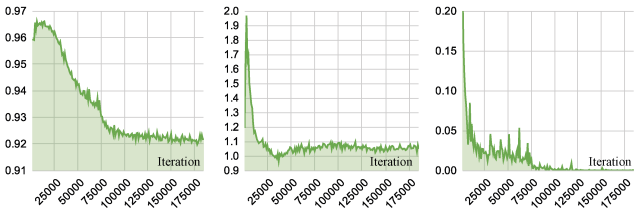
Cyclic inner-updates: We applied our cyclic inner-updating method to diversify virtual simulations. Figure S3 shows the performance differences over the cycle period. We observed the highest performance in the case of five epochs, so we set the cycle period to five epochs (*i.e.* 9,245 iterations). In other words, the meta-train step size β oscillates back and forth at every five epochs.

Step size in meta-optimization: While the step size β of inner-level optimization oscillates, the step size γ of final meta-optimization is assigned a fixed value. Figure S4 illustrates the final distribution ratios of balancing parameters according to the size of γ. As the step size increases, the balancing parameters were biased to the instance normalization. Thus, it is important to select an appropriate hyperparameter value considering the style variation between domains. We selected the step size γ as 0.1 and achieved the highest performance with it.



(a) L_{scat} (b) L_{shuf} (c) L_{tr}

Figure S1. Performance comparison according to the change of the weight parameters in the meta-train loss. We adjust each weight parameter while fixing the other two weight parameters to 1.0.



(a) L_{scat} (b) L_{shuf} (c) L_{tr}

Figure S2. The meta-train losses during the training process.

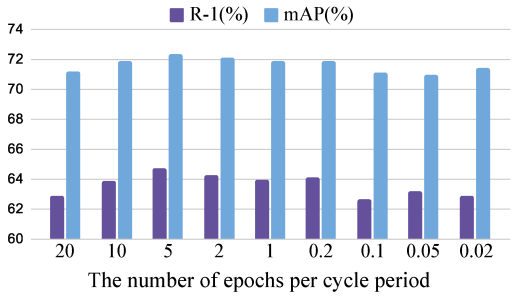


Figure S3. Analysis of a cycle period in inner-level optimization.

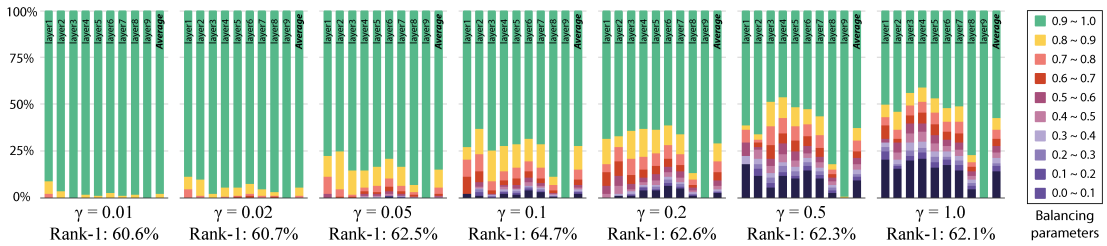


Figure S4. Analysis on the final distribution ratios of balancing parameters depending on the step size γ in meta-optimization.

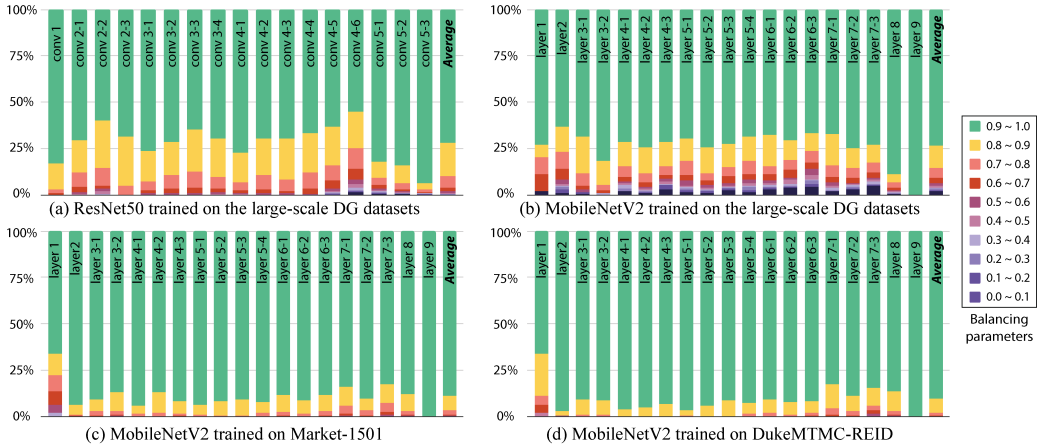
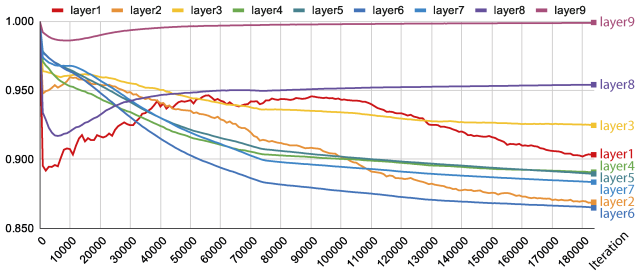


Figure S5. Analysis on the final distribution ratios of balancing parameters according to the network structures (*i.e.* MobileNetV2 and ResNet50) or training datasets (*i.e.* Market-1501, DukeMTMC-ReID, and large-scale domain generalization datasets).

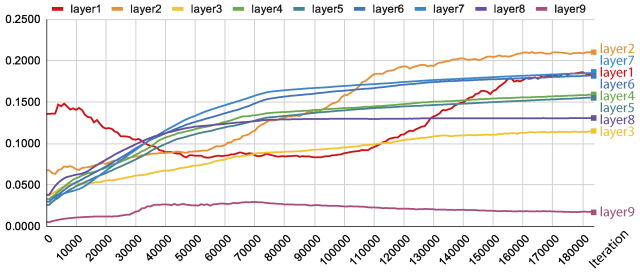
B. More Analysis on Balancing Parameters

Different network structures and training datasets: We compare the final distribution ratios of balancing parameters under different situations. Figure S5 (a) and (b) show the results corresponding to network structures. While the existing BIN method [2] normalizes instance-specific styles only in the shallow and deep layers (Fig. 4 in the manuscript), our MetaBIN method focuses on normalizing styles in the overall layers excluding the last layer, regardless of the network architecture. Figure S5 (c) and (d) show the final distributions of balancing parameters trained on Market-1501 [4] and DukeMTMC-ReID [5], respectively. At this time, we consider camera domains within a single-source dataset as multiple-source domains. We observed that the balancing parameters are hardly biased towards IN. The reason is that the camera-domain discrepancy within the single-source dataset is relatively smaller than the domain discrepancy between multiple-source datasets in the large-scale DG benchmark.

Parameter changes during training: We analyze how the balancing parameters are updated during the training process, as shown in Fig. S6. Interestingly, the mean value of balancing parameters in layers 1, 2, 8, and 9 dropped sharply at the beginning, but soon rebounded. It can be explained in line with the situation in Fig. S2. At the beginning of training, useless style information is removed from layers 1, 2, 8, and 9 to improve the discrimination ability rather than the generalization capability. After that, the balancing parameters are updated to overcome the unsuccessful generalization scenarios caused by the three types of meta train losses. Eventually, the model gradually becomes generalized through the training process.



(a) The mean value of balancing parameters for each layer



(a) The standard deviation of balancing parameters for each layer

Figure S6. Analysis on statistical characteristics of balancing parameters for each layer during the training process.

Table S1. Performance (%) and specification comparison under different network structures, where 'Mem' is the training memory usage in our MetaBIN framework and 'Time' is the inference time per image when the mini-batch size is 64.

Method	Performance										Specification					
	Average		VIPeR		PRID		GRID		i-LIDS		Dim	Norm Layers	Balancing Params	All Params	Mem (MiB)	Time (ms)
	R-1	*m*AP	R-1	*m*AP	R-1	*m*AP	R-1	*m*AP	R-1	*m*AP						
MobileNetV2 (w1.0)	61.9	70.1	54.8	64.0	70.2	77.8	44.2	53.2	78.5	85.3	1,280	53	17,088	2.26M	5,907	1.34
MobileNetV2 (w1.4)	64.7	72.3	56.9	66.0	72.5	79.8	49.7	58.1	79.7	85.5	1,792	53	23,822	4.34M	7,883	1.89
ResNet18	59.5	67.5	54.2	63.0	65.0	74.2	42.1	49.8	76.7	83.0	512	20	4,800	11.19M	2,473	0.80
ResNet34	62.8	71.0	57.1	66.0	73.0	79.5	43.7	54.1	77.3	84.4	512	36	8,512	21.30M	3,097	1.38
ResNet50	66.0	73.6	59.9	68.6	74.2	81.0	48.4	57.9	81.3	87.0	2,048	53	26,560	23.56M	6,885	2.63
ResNet101	68.1	75.9	61.5	70.2	77.1	83.3	52.7	62.8	81.2	87.2	2,048	104	52,672	42.61M	9,207	4.26
ResNet152	68.3	75.8	62.4	70.7	74.1	81.9	53.3	61.9	83.5	88.8	2,048	155	75,712	58.30M	12,665	5.92

C. Various architecture designs

Domain generalizable person re-identification aims to learn a robust model for obtaining good performance on the unseen target domain without additional updates. This task is more useful for real-world applications since it does not require any target images to train a model. Therefore, we share experimental results on various network architectures for practical use. Especially, we cover the variants of MobileNetV2 [3] and ResNet [1]. Table S1 shows the performance and specification corresponding to the different network structures. We employed a single NVIDIA Titan Xp GPU and set the input image size to 256×128. We also measured the maximum memory requirement for training the MetaBIN framework and calculated the inference time per image when a mini-batch was 64. We expect our MetaBIN method to be actively utilized in the real-world environment.

References

[1] 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 (CVPR)*, pages 770–778, 2016.

[2] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2558–2567, 2018.

[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[4] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, 2015.

[5] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3754–3762, 2017.