One-Shot Knowledge Transfer for Scalable Person Re-Identification —Supplementary Material—

Longhua Li[1,2] Lei Qi[1,2*] Xin Geng[1,2*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
{lhli, qilei, xgeng}@seu.edu.cn

A. Teacher Models and Student Models

A.1. CNN Architecture

For the CNN architecture, we utilize the pre-trained ResNet50 from LUPerson [2], which consists of four stages with inplanes values of 64, 128, 256, and 512, respectively. The output channels of each stage are four times its inplanes value. To construct student models of varying sizes, we introduce four adjustable coefficients: $inplanes$ and $multipliers[3]$. The architecture of these models is illustrated in Table A1. Here, $inplanes$ corresponds to the inplanes of the first stage, while $multipliers[3]$ represents the scaling factors of the inplanes for the subsequent three stages relative to the first stage. Based on these coefficients, we design six student models with progressively increasing sizes, where each consecutive model doubles the parameters of its predecessor. The student models are designated as Res-50-S1 through Res-50-S6. The specific attributes of these models are detailed in Table A2. While our approach supports variable widths for all layers, we use these coefficients for simplicity.

A.2. ViT Architecture

For the ViT architecture, we employ pre-trained ViT-B and ViT-S models from PASS [10]. For both ViT-B and ViT-S, we construct two student models each, adhering to the same parameter doubling scheme. These student models are denoted as ViT-B-S1, ViT-B-S2, ViT-S-S1, and ViT-S-S2. The student models differ from the teacher models only in dimensionality while sharing the patch embedding layer. To enable dimensionality reduction while preserving the shared patch embedding layer, we introduce a fully connected layer between the patch embedding layer and the backbone. The attributes of the teacher and student models are detailed in Table A3.

*Corresponding authors.

B. Weight Chains, Rows, and Columns

For CNNs, considering that the sizes of the student models of ResNet50 in Table A2 span a relatively wide range, in order to better inherit the knowledge of the teacher model, we refined three weight chains denoted as WTC-1, WTC-2, and WTC-3. Their $inplanes$ are 8, 16, and 32, respectively, which correspond to $1/8$, $1/4$, and $1/2$ of the width of the teacher model. In the experiments, WTC-1 is used to expand and obtain Res-50-S1 and Res-50-S2, WTC-2 is used to expand and obtain Res-50-S3 and Res-50-S4, and WTC-3 is used to expand and obtain Res-50-S5 and Res-50-S6. For ViT, we refined one weight chain for each of them, with dimensions of 192 and 96 respectively, which are one-fourth of the corresponding teacher models.

We use "rows" to denote both the weight matrix rows in FC layers and filters in Conv layers, while "columns" refer to the weight matrix columns in FC layers and channels in Conv layers, with each row outputting a feature dimension and each column processing a input feature dimension.

C. Experimental Details

C.1. Clustering Method

During the initialization of the weight chain, we employ AgglomerativeClustering [5] for clustering, setting $n\_clusters$ to the target dimension count while using default values for other parameters. In the ablation study presented in Table 6(b) (rand cluster), we first ensure that no cluster is empty by randomly selecting a weight row from the teacher model for each cluster. Subsequently, the remaining weight rows are assigned random cluster labels. In Table 6(c) (inverse cluster), we begin by modeling the relationships between weight rows and computing a distance matrix. This matrix is then inverted, and AgglomerativeClustering is applied to achieve clusters with larger internal distances.

Table A1. Model architectures of ResNet50 and its width-reduced student models. $\boldsymbol{p}$: *inplanes*. $[\boldsymbol{m_1}, \boldsymbol{m_2}, \boldsymbol{m_3}]$: *multipliers*.

Layer Name	Teacher: ResNet50	Students: Res-50-Sx
conv1	7×7, 64, stride 2	7×7, $\boldsymbol{p}$, stride 2
conv2_x	3×3 max pool, stride 2	3×3 max pool, stride 2
	$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, \boldsymbol{p} \\ 3\times 3, \boldsymbol{p} \\ 1\times 1, \boldsymbol{p}\times\boldsymbol{4} \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, \boldsymbol{p}\times\boldsymbol{m_1} \\ 3\times 3, \boldsymbol{p}\times\boldsymbol{m_1} \\ 1\times 1, \boldsymbol{p}\times\boldsymbol{m_1}\times\boldsymbol{4} \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, \boldsymbol{p}\times\boldsymbol{m_2} \\ 3\times 3, \boldsymbol{p}\times\boldsymbol{m_2} \\ 1\times 1, \boldsymbol{p}\times\boldsymbol{m_2}\times\boldsymbol{4} \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, \boldsymbol{p}\times\boldsymbol{m_3} \\ 3\times 3, \boldsymbol{p}\times\boldsymbol{m_3} \\ 1\times 1, \boldsymbol{p}\times\boldsymbol{m_3}\times\boldsymbol{4} \end{bmatrix} \times 3$
classifier	average pool, fc, softmax	average pool, fc, softmax

Table A2. Model attributes of ResNet50 and its student models. The teacher model is highlighted in gray.

Model	*Inplanes*	*Multipliers*	Params	FLPOs
Res-50-S1	8	[2,4,8]	0.37M	0.05G
Res-50-S2	16	[2,3,4]	0.61M	0.14G
Res-50-S3	16	[2,4,8]	1.48M	0.19G
Res-50-S4	32	[2,3,4]	2.42M	0.51G
Res-50-S5	32	[2,4,8]	5.89M	0.70G
Res-50-S6	64	[2,3,4]	9.64M	1.92G
ResNet50	64	[2,4,8]	23.5M	2.70G

Table A3. Model attributes of ViT-B and ViT-S, along with their student models. The teacher model is highlighted in gray.

Model	Type	Dimension	Params	MACs
ViT-S-S1	Student	120	2.20M	0.29G
ViT-S-S2	Student	168	4.42M	0.56G
ViT-S	Teacher	384	21.74M	2.87G
ViT-B-S1	Student	264	10.35M	1.36G
ViT-B-S2	Student	336	16.68M	2.20G
ViT-B	Teacher	768	86.24M	11.37G

C.2. Comparison Methods

DepGraph for model pruning. For model pruning, we assess DepGraph [1], an advanced method that models inter-layer dependencies to form groups, implements group sparsity learning prior to pruning, and subsequently fine-tunes the model. In the experiments of this paper, the settings recommended by the authors are uniformly adopted. For CNN, the combination of GroupNormImportance and GroupNormPruner is utilized. For ViT, the combination of GroupTaylorImportance and MetaPruner is employed.

KD++ for knowledge distillation. For knowledge distillation, we utilize KD++ [6], a state-of-the-art technique that aligns student features with teacher class means and enhances feature norms to achieve superior results. In the experiments of this paper, the parameters recommended by the authors are all used. The coefficients of the loss functions are set as follows: the kd loss factor is 1.0, the nd loss factor is 1.0, and the temperature coefficient is set as $t = 1.0$.

C.3. Combined with Modern Light-weight ReID Architectures

We employ the original-sized OSNet [9] and MSINet [3] as teacher models and perform weight chain refinement over 50 epochs to obtain smaller-scale models with width multipliers of 0.25 and 0.5. Following the same pre-training protocol, we utilize the MSMT17 [7] dataset for pre-training and subsequently fine-tune the resulting lightweight models on Market1501 [8] and CUHK03 [4] to evaluate their performance. Table C1 presents the results of direct pre-training on MSMT17, as well as the outcomes of models trained on MSMT17 using our proposed method.

Table C1. Results of modern ReID models initialized with various methods on MSMT17. β: width multiplier.

	β	Params	Method	mAP (%)	Rank-1 (%)
OSNet [9]	0.25	0.15M	Scratch	33.1	56.9
			OSKT	**40.6**	**65.7**
	0.5	0.56M	Scratch	45.3	69.7
			OSKT	**49.3**	**73.2**
MSINet [3]	0.25	0.17M	Scratch	23.0	50.0
			OSKT	**28.2**	**57.9**
	0.5	0.63M	Scratch	46.1	70.1
			OSKT	**50.3**	**73.7**

Table C2. The width of each weight chain in section 4.4.2. Each weight chain is configured with $multipliers = [2, 4, 8]$.

Method	$Inplanes$
# weight chain = 3	[8,16,32]
# weight chain = 4	[8,13,23,38]
# weight chain = 5	[8,12,18,28,42]
# weight chain = 6	[8,11,16,23,,32,45]

C.4. Scalability Analysis of Weight Chains

As described in Section 4.4.2, the width of each weight chain is determined using a geometric progression. Table C2 provides the specific widths of each weight chain.

To further enhance performance, we employ a progressive framework where a S-Student constructed with a larger weight chain serves as the teacher to guide the training of the next, smaller weight chain. Since the models built with smaller weight chains are significantly more compact than the original teacher model, this approach simultaneously improves both performance and efficiency.

D. Limitations and Future Directions

OSKT is limited to generating models with varying widths that share the same depth as the teacher model. When expanding to larger models using the weight chain, the performance is constrained by the limited knowledge that can be accommodated within the smaller parameter space of the weight chain. In the future, we aim to preserve more knowledge from the teacher model by increasing the capacity of the weight chain and designing more sophisticated mechanisms for knowledge inheritance.

References

[1] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. DepGraph: Towards Any Structural Pruning. In *CVPR*, 2023. 2

[2] Dengpan Fu, Dongdong Chen, Jianmin Bao, Hao Yang, Lu Yuan, Lei Zhang, Houqiang Li, and Dong Chen. Unsupervised pre-training for person re-identification. In *CVPR*, 2021. 1

[3] Jianyang Gu, Kai Wang, Hao Luo, Chen Chen, Wei Jiang, Yuqiang Fang, Shanghang Zhang, Yang You, and Jian Zhao. MSINet: Twins Contrastive Search of Multi-Scale Interaction for Object ReID. In *CVPR*, 2023. 2, 3

[4] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. DeepReID: Deep Filter Pairing Neural Network for Person Re-identification. In *CVPR*, 2014. 2

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 2011. 1

[6] Yuzhu Wang, Lechao Cheng, Manni Duan, Yongheng Wang, Zunlei Feng, and Shu Kong. Improving knowledge distillation via regularizing feature norm and direction. *arXiv:2305.17007*, 2023. 2

[7] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person Transfer GAN to Bridge Domain Gap for Person Re-Identification. *arXiv:1711.08565*, 2018. 2

[8] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable Person Re-identification: A Benchmark. In *ICCV*, 2015. 2

[9] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-Scale Feature Learning for Person Re-Identification. In *CVPR*, 2019. 2, 3

[10] Kuan Zhu, Haiyun Guo, Tianyi Yan, Yousong Zhu, Jinqiao Wang, and Ming Tang. Pass: Part-aware self-supervised pre-training for person re-identification. In *ECCV*, 2022. 1