Appendix of "DTL: Parameter- and Memory-Efficient Disentangled Vision Learning"

Minghao Fu, Ke Zhu, Zonghao Ding and Jianxin Wu, *Member, IEEE*

1 Full Evaluation Settings

1.1 Datasets

Image Classification. The datasets used in our experiments are:

ImageNet-1K. ImageNet-1K [1] is a commonly used large-scale benchmark, containing about 1.28 million training images from 1,000 classes and 50,000 images for evaluation.

FGVC. Following [2], [3], we evaluated on 5 Fine-Grained Visual Classification (FGVC) datasets: CUB-200-2011 [4], NABirds [5], Oxford Flowers [6], Stanford Dogs [7] and Stanford Cars [8]. FGVC is a widely used benchmark to evaluate PETL methods.

VTAB-1K. VTAB-1K was introduced by [9] to evaluate the generalization ability of representation learning approaches. It contains diverse images from 19 different datasets. which is grouped as 1) *Natural* images captured by standard cameras; 2) *Specialized* images captured by specialist equipment; and 3) *Structured* images generated in simulated environments. These datasets vary in task-specific objectives (e.g., classic visual recognition, object counting or depth prediction) and there are only 1,000 images in each dataset for training. It is a challenging benchmark to evaluate PETL methods.

Fine-Grained Few-Shot Learning. We further evaluate on 5 fine-grained few-shot learning benchmark: Aircraft [10], Pets [11], Food-101 [12], Cars [8] and Flowers102 [6]. Following [13], we finetune the pretrained model with training set containing {1, 2, 4, 8, 16}-shot per class and report the average accuracy on the test set over 3 seeds.

Domain Generalization. We follow [14] to conduct experiments on domain generalization to evaluate the robustness of our method when domain shift is inevitable. In this scenario, the training set to finetune the pretrained ViT-B/16 model is sampled from the original training set of ImageNet-1K, with each class containing 16 shot of images. Apart from the validation set of ImageNet-1K, the model is evaluated on 4 datasets: 1) ImageNet-Sketch [15] composed of sketch images sharing the same label space with ImageNet-1K, 2) ImageNet-V2 [16] collected from different sources compared with ImageNet-1K, 3) ImageNet-A [17] consisting of adversarial examples, and 4) ImageNet-R [18] containing various artistic renditions of ImageNet-1K. The reported accuracy is average by 3 different random seeds.

Dense Prediction. We conducted experiments on MS-COCO [19], Pascal VOC [20] and ADE20K [21].

MS-COCO. The MS-COCO 2017 dataset [19] is one of the most widely used benchmarks for evaluating object detection and instance segmentation methods. It consists of 118K training images and 5K validation images. Following [22], we use Swin-Base [23] + Cascade MASK R-CNN [24] as the detector for all experiments on MS-COCO. Average Precision (AP) is used as the evaluation metric.

Pascal VOC. The Pascal VOC 0712 dataset [20], comprising 16K training images and 5K validation images, is commonly used for assessing object detection algorithms. In our experiments, we use the Swin-Large + RetinaNet [25] as the detector, and use $\mathrm{AP}_{\mathrm{Box}}$ as the evaluation metric.

ADE20K. The ADE20K dataset [21], widely recognized as the standard benchmark for semantic segmentation, comprises 20K training and 2K validation images. For our experiments, we use Swin-Large + UperNet [26] as the segmentation model and mIoU as the evaluation metric.

1.2 Data Augmentation

Image Classification. Following [3], [14], we resized input images directly to 224 $\times$ 224 for the VTAB-1K [9] benchmark. For other datasets, random resized cropping is applied during training. In ImageNet-1K, few-shot learning, and domain generalization experiments, we also applied color jitter and RandAugment during finetuning, as in [3], [14]. In the evaluation phase, input images were resized to 256 $\times$ 256, followed by a center crop to 224 $\times$ 224.

Dense Prediction. We followed [22], [27] to conduct the training and evaluation for dense prediction tasks. Specifically, for the MS-COCO dataset, AutoAugment was applied to enable the model to perceive more combinations of image resolutions to facilitate the training. During evaluation, the input image was scaled while maintaining its aspect ratio, with the shorter side resized to 800 pixels for MS-COCO, 600 for Pascal VOC, and 512 for ADE20K, respectively.

1.3 Finetuning Details

Image Classification. Following [3], [14], we used AdamW [28] as the optimizer, with a weight decay of 0.05. The learning rate was decayed using a cosine scheduler. For experiments on ImageNet-1K, the pretrained models were

• *All authors are with the National Key Laboratory for Novel Software Technology, Nanjing University, China and the School of Artificial Intelligence, Nanjing University, China. J. Wu is the corresponding author. E-mail: {fumh, zhuk, dingzh}@lamda.nju.edu.cn, wujx2001@gmail.com*

finetuned for 30 epochs with a learning rate of 1e-3, using a local batch size of 64 on 4 GPUs (a total batch size of 256).

For other experiments, the pretrained models were finetuned for 100 epochs. Specifically, for experiments on VTAB-1K, fine-grained few-shot learning, and domain generalization, we finetuned using a batch size of 32 on a single GPU. For experiments on the five FGVC datasets, we finetuned with a local batch size of 32 across two GPUs (total batch size of 64). As in [3], the learning rate was selected based on validation accuracy for each dataset. Unlike previous methods [3], [14], in this case, we did *not* employ additional techniques such as mixup [29], cutmix [30], or label smoothing [31].

Dense Prediction. Following [22], [27], we used AdamW [28] as the optimizer. For the MS-COCO dataset, we finetuned the pretrained models with a learning rate of 1e-4 over 36 epochs, with the learning rate decreasing at the 27th and 33rd epochs. For the Pascal VOC dataset, we finetuned the models with a learning rate of 1e-4 over 12 epochs, reducing the learning rate at the 10th epoch. For the ADE20K dataset, we finetuned the models with a learning rate of 0.001 over 160,000 iterations. In all experiments, the batch size per GPU was set to 2. We conducted our experiments using 4 RTX 3090 GPUs for MS-COCO, 1 for Pascal VOC, and 8 for ADE20K.

1.4 Detailed Architectures of DTL

In Table 1, we provide a detailed overview of the DTL architectures across different backbones used for image classification and dense prediction tasks. Notably, we adhere to the principle of adapting the output of approximately *half of the late blocks* in the backbone with CSN for downstream tasks. Consequently, the 'g' and 'Sparse G' terms typically start from the intermediate block in the backbone. Interestingly, for ViT-g/14, adapting only about 30% of the late blocks is sufficient to achieve strong downstream accuracy, highlighting the effectiveness of our DTL approach, particularly for large-scale models. The architecture of Swin-L for the ADE20K case differs slightly from other Swin Transformer cases, as it has been further optimized to achieve a better balance between parameter efficiency and segmentation mIoU. However, its design still adheres to the principles described in the main paper.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. 1

[2] M. Jia, L. Tang, B.-C. Chen *et al.*, "Visual prompt tuning," in *European Conference on Computer Vision*, ser. LNCS, vol. 13693, 2022, pp. 709–727. 1

[3] D. Lian, D. Zhou, J. Feng, and X. Wang, "Scaling & shifting your features: A new baseline for efficient model tuning," in *Advances in Neural Information Processing Systems*, 2022, pp. 109–123. 1, 2

[4] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011. 1

[5] G. Van Horn, S. Branson, R. Farrell *et al.*, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 595–604. 1

[6] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008, pp. 722–729. 1

[7] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1

[8] T. Gebru, J. Krause, Y. Wang, D. Chen, J. Deng, and L. Fei-Fei, "Fine-grained car detection for visual census estimation," in *AAAI Conference on Artificial Intelligence*, 2017, p. 4502–4508. 1

[9] X. Zhai, J. Puigcerver, A. Kolesnikov *et al.*, "A large-scale study of representation learning with the Visual Task Adaptation Benchmark," *arXiv:1910.04867*, 2019. 1

[10] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *arXiv:1306.5151*, 2013. 1

[11] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3498–3505. 1

[12] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with Random Forests," in *European Conference on Computer Vision*, ser. LNCS, vol. 8694, 2014, pp. 446–461. 1

[13] S. Jie and Z.-H. Deng, "FacT: Factor-Tuning for lightweight adaptation on Vision Transformer," in *AAAI Conference on Artificial Intelligence*, 2023, pp. 1060–1068. 1

[14] Y. Zhang, K. Zhou, and Z. Liu, "Neural prompt search," *arXiv:2206.04673*, 2022. 1, 2

[15] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 506–10 518. 1

[16] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?" in *International Conference on Machine Learning*, 2019, pp. 5389–5400. 1

[17] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 257–15 266. 1

[18] D. Hendrycks, S. Basart, N. Mu *et al.*, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8320–8329. 1

[19] T.-Y. Lin, M. Maire, S. Belongie *et al.*, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, ser. LNCS, vol. 8693, 2014, pp. 740–755. 1

[20] M. Everingham, S. M. Eslami, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015. 1

[21] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5122–5130. 1

[22] D. Yin, Y. Yang, Z. Wang, H. Yu, K. Wei, and X. Sun, "1% vs 100%: Parameter-efficient low rank Adapter for dense predictions," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 116–20 126. 1, 2

[23] Z. Liu, Y. Lin, Y. Cao *et al.*, "Swin Transformer: Hierarchical Vision Transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9992–10 002. 1

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988. 1

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 2999–3007. 1

[26] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *European Conference on Computer Vision*, ser. LNCS, vol. 11209, 2018, pp. 432–448. 1

[27] D. Yin, L. Hu, B. Li, Y. Zhang, and X. Yang, "5% >100%: Breaking performance shackles of full fine-tuning on visual recognition tasks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 20 071–20 081. 1, 2

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019, pp. 1–18. 1, 2

[29] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "*mixup*: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018, pp. 1–13. 2

[30] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6022–6031. 2

Table 1: Detailed architectures of DTL. '#Blocks' indicates the number of blocks in the backbone. d' specifies d' of the low-rank linear transformation in CSN. 'w' and 'Sparse w' indicate the indices at which the input of the backbone blocks is extracted to serve as the input for CSN. Similarly, 'g' and 'Sparse G' represent the indices where the output of the backbone blocks is adapted by CSN.

Backbone	Pretrained Weights	#Blocks	d'	DTLv1		DTLv2 & DTLv2-LS	
				w	g	Sparse w	Sparse G
Image Classification							
ViT-B/16	checkpoint[1]	12	2	1-12	7-12	{1,3,5,7,9,11,12}	{7,9,11,12}
Swin-B	checkpoint[2]	24	4	5-24	13-24	{2,4,5,9,13,17,22,24}	{13,15,17,19,22,24}
ConvNeXt-B	checkpoint[3]	36	4	7-36	19-36	{3,6,7,11,15,19,23,27,31,33,36}	{19,21,23,25,27,29,31,33,36}
ViT-g/14	checkpoint[4]	40	4	1-40	29-40	{1,5,9,13,17,21,25,29,33,37,40}	{29,33,37,40}
Dense Prediction							
Swin-B (COCO)	checkpoint[2]	24	256	5-24	13-24	{2,4,5,9,13,17,22,24}	{13,15,17,19,22,24}
Swin-L (Pascal VOC)	checkpoint[5]	24	128	5-24	13-24	{2,4,5,9,13,17,22,24}	{13,15,17,19,22,24}
Swin-L (ADE20K)	checkpoint[5]	24	192	5-24	11-24	{5,9,13,17,21,23,24}	{11,13,15,17,19,21,23,24}

[1] https://storage.googleapis.com/vit_models/imagenet21K/ViT-B_16.npz
[2] https://github.com/SwinTransformer/storage/releases/download/v1.0.0/swin_base_patch4_window7_224_22k.pth
[3] https://dl.fbaipublicfiles.com/convnext/convnext_base_22k_224.pth
[4] https://dl.fbaipublicfiles.com/dinov2/dinov2_vitg14/dinov2_vitg14_pretrain.pth
[5] https://github.com/SwinTransformer/storage/releases/download/v1.0.0/swin_large_patch4_window7_224_22k.pth

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826. 2