Supplementary Material
# Budget-Aware Adapters for Multi-Domain Learning

Rodrigo Berriel[1*] Stéphane Lathuilière[2] Moin Nabi[3] Tassilo Klein[3]
Thiago Oliveira-Santos[1] Nicu Sebe[2] Elisa Ricci[2,4]
[1]LCAD, UFES [2]DISI, University of Trento [3]SAP ML Research [4]Fondazione Bruno Kessler
berriel@lcad.inf.ufes.br

In this supplementary materials, we first provide more details about the training and evaluation protocols used in our experiments (Section 1). Then, in Section 2, we report additional experiments on the ImageNet-to-Sketch benchmark.

## 1. Additional training and testing details

In order to draw a fair comparison with other methods, we employed the same training schedule and hyperparameters of previous works. Here, we provide more details about the training and test procedures used in the three experiments: (i) Visual Decathlon Challenge, (ii) ImageNet-to-Sketch benchmark, and (iii) Single-Domain Classification.

**Visual Decathlon challenge** Following previous works [3, 5, 6, 7], we employ the Wide ResNet WRN-28-4-*B*(3,3), i.e., 28 convolutional layers with widening factor of 4 and the original "basic" block (2 convolutions using $3 \times 3$ kernel size). As in [6, 5], random crops of $64 \times 64$ pixels are used to feed the network during training. The optimizer parameters are set following [3, 5], where SGD with momentum is employed for the classifier and Adam for the rest of the architecture with initial learning rates of 0.001 and 0.0001, respectively. The model is trained with batch size of 32 for 60 epochs; after 45 epochs, the learning rates are decayed by a factor of 10. The real-valued switches ($\tilde{s}_c$) are initialized with a value of 0.001. In addition to random cropping, in regards to training data augmentation, horizontal mirroring is also applied with a 50% probability, except for four datasets (DTD, Omniglot, SVHN, and GTSR) on which it could be either harmful or useless. During training, the models for all the domains are initialized using the ImageNet pre-trained weights and these weights are kept fixed, i.e., only the domain-specific parameters ($\tilde{s}_c$, Batch Normalization and classifiers) are learned. At test time, we follow the procedure proposed in [2] and used in [5]. We employ a ten-crop strategy for the datasets in which horizontal mirroring was applied and five-crop otherwise. In the five-crop strategy, a crop is performed on each corner of the image in addition to a central one; the ten-crop adds horizontally mirrored versions of each one of the five crops. The final prediction is based on the average of the predictions over all the crops.

*Work done when with MHUG (DISI, University of Trento).

**ImageNet-to-Sketch benchmark** We use the ResNet-50 as in [3, 5] feeding a random crop of $224 \times 224$ pixels after resizing the images to $256 \times 256$ pixels. In addition to random cropping, horizontal mirroring is applied to all datasets during training. The networks are initially trained using the same schedule as in [3, 5], i.e., 30 epochs with a learning rate drop (with a factor of 10) after 15 epochs. For the lowest budget ($\beta = 0.25$), we add another learning rate drop at 30 epochs and train for additional 15 epochs in order to fully satisfy the constraints. All the other steps are performed as in the Visual Decathlon Challenge.

**Single-domain classification** In these experiments, we use the same variant of the ResNet proposed in the original Residual Networks [1] paper for the CIFAR-10 dataset with $n = 9$, which results in a ResNet with 56 layers. First, we train the baseline model, which is the model without *switches*. Since we are interested in single-domain learning for this experiment, we do not need to freeze the weights as in the multi-domain experiments. Therefore, we jointly train the switches and the weights using the baseline pre-trained weights as initialization. In these cases, we use SGD with momentum (initial learning rate of 0.1) for both the baseline model and the joint training. Concerning data augmentation, we use random crops of $32 \times 32$ pixels with 4 pixel padding and horizontal mirroring with 50% probability. The same setting is used for the CIFAR-100.

## 2. Results

**ImageNet-to-Sketch** In the paper, we report the results on the ImageNet-to-Sketch benchmark using the ResNet-

| | FLOP | Params | ImageNet | CUBS | Cars | Flowers | WikiArt | Sketch | Score | $S_O$ | $S_P$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Classifier Only [3] | 1 | 1 | 74.4 | 73.5 | 56.8 | 83.4 | 54.9 | 53.1 | 328 | 328 | 328 |
| Individual Networks [3] | 1 | 6 | 74.4 | 81.7 | 91.4 | 96.5 | 76.4 | 80.5 | 1500 | 1500 | 250 |
| PackNet $\rightarrow$ [4] | 1 | **1.11** | 74.4 | 80.7 | 84.7 | 91.1 | 66.3 | 74.7 | 691 | 691 | 623 |
| PackNet $\leftarrow$ [4] | 1 | **1.11** | 74.4 | 69.6 | 77.9 | 91.5 | 69.2 | 78.9 | 610 | 610 | 550 |
| Piggyback [3] | 1 | 1.15 | 74.4 | 79.7 | 87.2 | 94.3 | 72.0 | **80.0** | 951 | 951 | 827 |
| Piggyback+BN [3] | 1 | 1.21 | 74.4 | 81.4 | 90.1 | 95.5 | <u>73.9</u> | 79.1 | 1215 | 1215 | 1004 |
| WTPB [5] | 1 | 1.21 | 74.4 | <u>81.7</u> | 91.6 | **96.9** | **75.7** | 79.8 | **1540** | 1540 | **1268** |
| *BA*$^2$ (Ours) ($\beta = 1.00$) | 0.687 | <u>1.17</u> | 74.4 | **82.4** | **92.9** | <u>96.0</u> | 71.5 | <u>79.9</u> | <u>1440</u> | <u>2096</u> | <u>1230</u> |
| *BA*$^2$ (Ours) ($\beta = 0.75$) | 0.578 | <u>1.17</u> | 74.4 | 81.2 | <u>91.9</u> | 94.9 | 68.9 | <u>79.9</u> | 1193 | 2064 | 1019 |
| *BA*$^2$ (Ours) ($\beta = 0.50$) | 0.543 | <u>1.17</u> | 74.4 | 78.2 | 89.2 | 95.0 | 66.2 | 78.8 | 925 | 1703 | 790 |
| *BA*$^2$ (Ours) ($\beta = 0.25$) | **0.375** | <u>1.17</u> | 74.4 | 76.2* | 88.4* | 94.7* | 67.9* | 78.4 | 840 | **2240** | 717 |

Table 1: State-of-the-art comparison on the ImageNet-to-Sketch benchmark using DenseNet-121 architecture. (*) Even though the average sparsities are greater than 75%, these models did not satisfy the constraint for every single layer.

50 architecture. In addition to this model, results using the DenseNet-121 are also reported in several works [3, 4, 5]. For that reason, we also provide these results in Table 1. First, we see that our method achieves the best scores in two domains and the second best in three other domains. Interestingly, in the *Cars* domain, our method with $\beta = 0.75$ is the second best model (only after our method with $\beta = 1.00$), achieving results better than all the other methods using only 57.8% of the FLOP, on average. Concerning the number parameters, our approach is the second best in terms of Params. Indeed the number of batch normalization and $1 \times 1$ convolutions parameters in the DenseNet-121 model with respect to the total number of parameters is higher than in the ResNet-50 model. Nevertheless, *BA*$^2$ is still the only one with FLOP less than 1. Finally, it can be noted that our method with $\beta = 1.00$ still achieves a good trade-off between performance and complexity.

## References


[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[3] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights. In *European Conference on Computer Vision (ECCV)*, 2018.

[4] Arun Mallya and Svetlana Lazebnik. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Massimiliano Mancini, Elisa Ricci, Barbara Caputo, and Samuel Rota Bulò. Adding New Tasks to a Single Network with Weight Transformations using Binary Masks. In *European Conference on Computer Vision Workshops (ECCVW)*, 2018.

[6] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[7] Amir Rosenfeld and John K. Tsotsos. Incremental Learning Through Deep Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. Early Access.