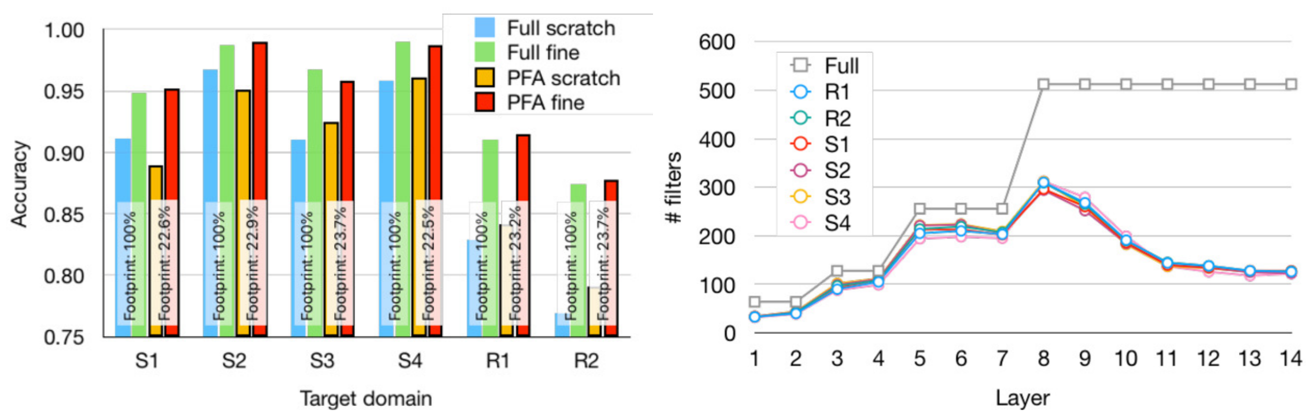
Appendices

A. Details on the domain adaptation experiments

In this section the exact classes used for the domain adaptation experiments shown in Sec. 4.3 are presented. In Sec. 4.3 the initial domain $\mathcal{D}_A$ is equal to the whole CIFAR-100 dataset. We then use different subsets of CIFAR-100 to generate target domains $\mathcal{D}_Z$. Here we are specifying exactly which classes are used for the different $\mathcal{D}_Z$s:

- **R1**: aquarium_fish, butterfly, cloud, elephant, mountain, palm_tree, poppy, snail, squirrel, wardrobe;
- **R2**: baby, camel, cockroach, flatfish, mouse, pear, porcupine, snake, sunflower, whale;
- **S1**: bowl, snail;
- **S2**: aquarium_fish, boy;
- **S3**: bee, raccoon;
- **S4**: train, tulip.

A.1. Domain adaptation from CIFAR-10



(a) Domain adaptation from CIFAR-10

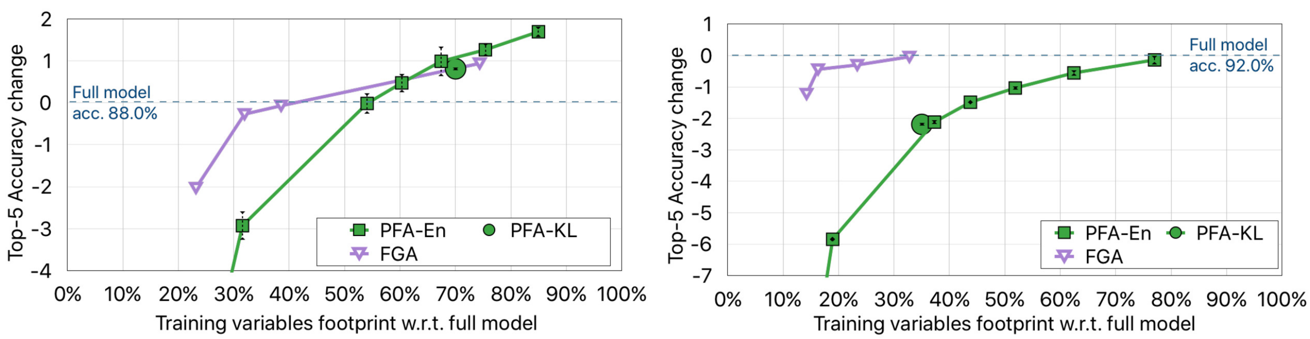
(b) PFA recipes starting from CIFAR-10

Figure 5. Domain adaption from CIFAR-10. (a) As when starting from CIFAR-100 (Fig 4), *PFA fine* matches the accuracy of *Full fine* while using architectures more than 4x smaller. *PFA fine* significantly outperforms the full model trained from scratch *Full scratch*. The vertical percentage labels show the PFA compression ratio. In (b) recipes for VGG-16 trained on CIFAR-100 using PFA-KL with data from different target domains. As opposed to Fig. 4, where the original network was trained on CIFAR-100, the recipes obtained when starting from CIFAR-10 are much more similar. This is due to the fact that the target domain is not included in the origin domain, and more re-learning is required.

B. Compression Results

In this section for each pair architecture-dataset that we used in the experiments presented in Sec. 4.1 we report the accuracy achieved by the full model used for pruning, and for each PFA recipe we provide the change in the accuracy and the percentage of the trainable variables and FLOPs of the pruned model with respect to the full model. The number of trainable variables is computed by summing the products of the shape of all trainable variables returned by TensorFlow [15] 1.4.0 `trainable.variable()` API. The FLOPs are computed by running the TensorFlow 1.4.0 profiler. Furthermore, when available, we report the same details for state-of-the-art work that we used for comparison. Results for VGG-16 with CIFAR-10 and CIFAR-100 are in Tabs. 1 and 2. Results for ResNet-56 with CIFAR-10 and CIFAR-100 are in Tabs. 3 and 4. Results for VGG-16 with ImageNet (top-1 and top-5) are in Tabs. 5 and 6. Results for ResNet-18 with ImageNet (top-1 and top-5) are in Tabs. 7 and 8.

In Sec. 4.1 we presented, among others, the Top-1 accuracy of PFA applied to VGG-16 and ResNet-34 on ImageNet (Fig. 3). Here, in Fig. 6 we also provide the Top-5 results for the same experiment.



(a) VGG-16 on ImageNet

(b) ResNet-34 on ImageNet

Figure 6. Top-5 results on ImageNet. Accuracy change in the y axis is reported in percentage points.

Table 1. VGG-16 with CIFAR-10.

VGG-16	Accuracy	% Trainable Var.	% FLOPs
Full Model	92.07%	100%	100%
	Δ Accuracy	% Trainable Var.	% FLOPs
PFA-En 0.99	0.31	18.25%	51.98%
PFA-En 0.98	0.35	12.70%	42.27%
PFA-En 0.97	0.14	10.03%	36.29%
PFA-En 0.96	-0.01	8.33%	31.67%
PFA-En 0.95	-0.05	7.12%	28.21%
PFA-En 0.93	-0.27	5.45%	22.56%
PFA-En 0.90	-0.97	3.87%	16.61%
PFA-En 0.80	-3.97	1.38%	6.10%
PFA-KL	0.244	19.63%	41.70%
FP [29]	0.15	36.00%	65.81%
NS [30]	0.14	11.48%	49.06%
VIB [11]	0.81	5.79%	N.A.

Table 2. VGG-16 with CIFAR-100.

VGG-16	Accuracy	% Trainable Var.	% FLOPs
Full Model	68.36%	100%	100%
	Δ Accuracy	% Trainable Var.	% FLOPs
PFA-En 0.99	1.68	43.15%	70.97%
PFA-En 0.98	1.41	33.14%	59.97%
PFA-En 0.97	1.21	27.13%	52.13%
PFA-En 0.96	0.78	22.72%	45.91%
PFA-En 0.95	0.50	19.29%	40.31%
PFA-En 0.93	-0.27	14.39%	32.06%
PFA-En 0.90	-2.37	9.66%	23.16%
PFA-En 0.80	-8.71	3.01%	8.12%
PFA-KL	1.40	41.91%	55.29%
NS [30]	0.22	24.90%	62.86%
VIB [11]	1.17	15.08%	N.A.
FGA/A [39]	0.39	39.67%	59.97%
FGA/B [39]	-0.02	18.54%	26.60%
FGA/C [39]	-0.57	13.20%	15.18%
FGA/D [39]	-1.93	11.31%	11.42%

Table 3. ResNet-56 with CIFAR-10.

ResNet-56	Accuracy	% Trainable Var.	% FLOPs
Full Model	93.15%	100%	100%
	Δ Accuracy	% Trainable Var.	% FLOPs
PFA-En 0.99	-0.20	72.60%	80.69%
PFA-En 0.98	-0.18	61.89%	70.04%
PFA-En 0.97	-0.80	54.10%	61.85%
PFA-En 0.96	-0.70	48.27%	55.46%
PFA-En 0.95	-1.01	43.36%	49.51%
PFA-En 0.93	-1.25	36.11%	41.61%
PFA-En 0.90	-1.81	27.42%	28.08%
PFA-En 0.80	-4.10	12.57%	14.56%
PFA-KL	-0.65	59.97%	61.57%
FP/A [29]	0.06	90.59%	89.60%
FP/B [29]	0.02	85.88%	72.72%

Table 4. ResNet-56 with CIFAR-100.

ResNet-56	Accuracy	% Trainable Var.	% FLOPs
Full Model	70.92	100%	100%
	Δ Accuracy	% Trainable Var.	% FLOPs
PFA-En 0.99	-1.86	90.28%	89.46%
PFA-En 0.98	-1.68	81.95%	79.36%
PFA-En 0.97	-2.33	74.92%	71.54%
PFA-En 0.96	-2.75	68.50%	64.80%
PFA-En 0.95	-3.50	62.76%	59.20%
PFA-En 0.93	-4.29	35.87%	41.61%
PFA-En 0.90	-5.16	27.24%	29.18%
PFA-En 0.80	-9.70	19.58%	16.76%
PFA-KL	-2.97	74.00%	65.60%
–			

Table 5. VGG-16-Conv Top-1 with ImageNet.

VGG-16 Top-1	Accuracy	% Model Size	% FLOPs
Full Model	67.73%	100%	100%
	Δ Accuracy	% Model Size	% FLOPs
PFA-En 0.99	2.80	84.98%	77.31%
PFA-En 0.98	1.97	75.40%	66.90%
PFA-En 0.97	1.59	67.43%	58.75%
PFA-En 0.96	0.90	60.32%	52.03%
PFA-En 0.95	0.02	54.06%	46.14%
PFA-En 0.90	-5.02	31.58%	26.46%
PFA-En 0.85	-20.94	10.32%	8.72%
PFA-KL	1.06	70.03%	53.57%
FGA/A [39]	0.82	74.37%	43.01%
FGA/B [39]	-0.28	38.55%	22.14%
FGA/C [39]	-1.06	31.95%	19.67%
FGA/D [39]	-3.49	23.18%	15.16%

Table 6. VGG-16-Conv Top-5 with ImageNet.

VGG-16 Top-5	Accuracy	% Model Size	% FLOPs
Full Model	88.03%	100%	100%
	Δ Accuracy	% Model Size	% FLOPs
PFA-En 0.99	1.70	84.98%	77.31%
PFA-En 0.98	1.27	75.40%	66.90%
PFA-En 0.97	0.99	67.43%	58.75%
PFA-En 0.96	0.47	60.32%	52.03%
PFA-En 0.95	-0.02	54.06%	46.14%
PFA-En 0.90	-2.93	31.58%	26.46%
PFA-En 0.85	-15.19	10.32%	8.72%
PFA-KL	0.81	70.03%	53.57%
FGA/A [39]	0.94	74.37%	43.01%
FGA/B [39]	-0.07	38.55%	22.14%
FGA/C [39]	-0.27	31.95%	19.67%
FGA/D [39]	-2.03	23.18%	15.16%

Table 7. ResNet-34 Top-1 with ImageNet.

ResNet-34 Top-1	Accuracy	% Model Size	% FLOPs
Full Model	74.49%	100%	100%
	Δ Accuracy	% Model Size	% FLOPs
PFA-En 0.99	-0.27	77.02%	76.00%
PFA-En 0.98	-1.08	62.46%	65.39%
PFA-En 0.97	-1.85	51.88%	57.14%
PFA-En 0.96	-2.71	43.79%	50.47%
PFA-En 0.95	-3.83	37.30%	44.73%
PFA-En 0.90	-9.78	18.94%	26.02%
PFA-En 0.85	-29.05	6.37%	9.45%
PFA-KL	-4.04	35.02%	48.21%
FP/A [29]	-0.67	92.13%	84.62%
FP/B [29]	-1.06	89.35%	75.82%
FP/C [29]	-0.75	93.06%	92.58%
FGA/A [39]	-0.35	32.78%	54.37%
FGA/B [39]	-1.02	23.37%	35.25%
FGA/C [39]	-1.70	16.30%	19.67%
FGA/D [39]	-3.03	14.23%	15.16%

Table 8. ResNet-34 Top-5 with ImageNet.

ResNet-34 Top-5	Accuracy	% Model Size	% FLOPs
Full Model	91.99%	100%	100%
	Δ Accuracy	% Model Size	% FLOPs
PFA-En 0.99	-0.14	77.02%	76.00%
PFA-En 0.98	-0.55	62.46%	65.39%
PFA-En 0.97	-1.03	51.88%	57.14%
PFA-En 0.96	-1.49	43.79%	50.47%
PFA-En 0.95	-2.12	37.30%	44.73%
PFA-En 0.90	-5.84	18.94%	26.02%
PFA-En 0.85	-20.27	6.37%	9.45%
PFA-KL	-2.19	35.02%	48.21%
FGA/A [39]	-0.04	32.78%	54.37%
FGA/B [39]	-0.30	23.37%	35.25%
FGA/C [39]	-0.44	16.30%	19.67%
FGA/D [39]	-1.22	14.23%	15.16%

C. Architectures and Training Details

This appendix is meant to provide all details needed to reproduce the results presented in the paper.

C.1. SimpleCNN

The SimpleCNN network is used in Sec. 4.1 in order to perform the experiments related to the upper-bound and the ablation studies.

SimpleCNN is composed of the following layers: 3 conv layers of size 96x3x3, a drop-out layer, 3 conv layers of size 192x3x3, drop-out layer, 1 conv layer of size 192x3x3, 1 conv layer of size 192x1x1, 1 conv layer of size [number of classes]x1x1, and finally an average pooling before the softmax layer. We use batchnorm and ReLU activations after every convolutional layer.

In the following table we list the details for the training of the full and compressed architectures.

	SimpleCNN on CIFAR-10 and CIFAR-100
Optimizer	Nesterov mom. 0.9, no decay
Learning rate	0.1
Learning rate decay factor	0.1
Epochs	50
Epochs per decay	30
Weights decay	0.0001
Batch size	512
Batch-norm	moving average decay 0.99, epsilon: 0.001
Drop-out	0.5
Augmentation	–

C.2. Training VGG-16 and ResNet-56 on CIFAR-10 and CIFAR-100

In the following tables we list the details for the training of the full and compressed architectures used for the results presented in Sec. 4.1, Fig. 2.

	VGG-16 on CIFAR-10 and CIFAR-100
Optimizer	Nesterov mom. 0.9, no decay
Learning rate	0.1
Learning rate decay factor	0.1
Epochs	160
Epochs per decay	90
Weights decay	0.0001
Batch size	256
Batch-norm	moving average decay 0.99, epsilon: 0.001
Drop-out	0.5
Augmentation	we pad the image with 4 pixels around the boarder and randomly crop a patch of size 32x32 and randomly flip the image

	ResNet-56 on CIFAR-10 and CIFAR-100
Optimizer	Nesterov mom. 0.9, no decay
Learning rate	0.1
Learning rate decay factor	0.1
Epochs	160
Epochs per decay	90
Weights decay	0.0005
Batch size	256
Batch-norm	moving average decay 0.99, epsilon: 0.001
Drop-out	0.5
Augmentation	we pad the image with 4 pixels around the boarder and randomly crop a patch of size 32x32 and randomly flip the image

After compression it is possible that some ResNet blocks provide features maps with smaller number of channels than those forwarded by the respective skip-connections. In those cases, instead of padding the skip-connection we pad the output of the block before the combination with the skip-connection. This let us arbitrarily compress each block while ensuring the correct depth of the feature maps.

C.3. Training VGG-16 and ResNet-34 on ImageNet

In the following table we list the details for the training of the full and compressed architectures used for the results presented in Sec. 4.1, Fig. 3 and in the App. B, Fig. 6. For training VGG-16 with ImageNet we use distributed training [34] with 8 machines and 8 GPUs each.

In the following tables we list the details for the training of the full and compressed architectures.

	VGG-16 on ImageNet
Optimizer	Momentum
Initial learning rate	0.1
Max. learning rate	1.6
Learning rate decay factor	0.9
Warming up epochs	5
Epochs	90
Epochs per decay	2
Weights decay	0.00004
Batch size	64
Batch-norm	moving average decay 0.99, epsilon: 0.001
Drop-out	0.5
Augmentation	We resize the shortest side of each image to 256 then we randomly crop an area of size 224x224 and randomly flip, finally we remove the average values for each RGB channel: 123.68, 116.78, 103.94.

	ResNet-34-16 on ImageNet
Optimizer	Momentum
Initial learning rate	0.1
Max. learning rate	1.6
Learning rate decay factor	0.85
Warming up epochs	2
Epochs	180
Epochs per decay	4
Weights decay	0.0001
Std weights in conv.	0.1
Batch size	64
Batch-norm	moving average decay 0.99, epsilon: 0.0001
Drop-out	0.5
Loss label smoothing	0.1
Augmentation	We resize the shortest side of each image to 256 then we randomly crop an area of size 224x224 and randomly flip, finally we remove the average values for each RGB channel: 123.68, 116.78, 103.94.

When using skip-connections with projections the application of PFA becomes easier than with padding. We analyze the output of the combination between the two branches (skip-connection and ResNet block) and reflect the compression back to the convolutions used for projection and that used as last step of the ResNet block.