Appendix

A. Comparison between From-Scratch Pruning and Transfer

We now present an experiment which supports our claim that from-scratch pruning and finetuning is inferior to transfer from ImageNet sparse models. For instance, image classification on the CIFAR-100 [38] dataset using a WideResNet [68] architecture, following [57], the AC/DC and GMP pruning methods at 90% sparsity reach 79.1% and 77.7% Top-1 validation accuracies, respectively. In contrast, finetuning from a ResNet50 backbone pruned on ImageNet using AC/DC and GMP at 90% sparsity reaches a validation accuracy of 83.9% and 84.4%, respectively (please see Table C.2 for the results). This example serves to illustrate the significant accuracy gains from using transfer learning with sparse models, as opposed to training sparse models from scratch.

B. Hyperparameters and Training Setup

Here we discuss the general hyperparameters and experimental setup used for the full and linear finetuning experiments. Regarding data loading image augmentation settings, we are careful to match them to the ones used in the original upstream training protocol. Specifically, this affects the choice of whether to use Bicubic or Bilinear image interpolation for image resizing; for example, RigL models were trained using Bicubic interpolation, whereas the other pruning methods considered used the Biliniar interpolation. All ResNet and MobileNet models considered were trained using standard ImageNet-specific values for the normalization mean and standard deviation. In the case of full finetuning, we used dataset-specific normalization values for the downstream tasks; these were obtained by loading the dataset once with standard data augmentations and computing the means and variances of the resulting data. For linear finetuning, we use center cropping of the images, followed by normalization using standard ImageNet values. For both full and linear finetuning, we use the same training hyperparameters as [61]; specifically, we train for 150 epochs, decreasing the initial learning rate by a factor of 10 every 50 epochs. We use 0.01 as the initial learning rate for all linear finetuning experiments; for full finetuning, we empirically found 0.001 to be the initial learning rate which gives comparable results for most datasets except Aircraft and Cars, for which we use 0.01. Our experiments were conducted using PyTorch 1.8.1 and NVIDIA GPUs. All full finetuning experiments on the ResNet50 backbone were repeated three times and all linear finetuning experiments five times.

C. Extended ResNet50 Results

In this section, we provide additional details, together with the complete results for our experiments for linear and full finetuning from ResNet50, presented in Sections 3.3 and 3.4. For each pruning method, we used a range of sparsity levels, and trained linear and full finetuning for each model and sparsity level, on all 12 downstream tasks; each experiment was repeated 5 times for linear and 3 times for full finetuning. Note that checkpoints for some pruning methods were not available for some of the higher sparsities.

C.1. Linear Finetuning Results

We provide the complete results for our linear finetuning experiments on each downstream task, for all pruning methods and sparsity levels considered. The results for the transfer accuracies for each pruning strategy, sparsity level, and downstream task are presented in Figure C.1 and Table C.1. We discussed in Section 3.3 that regularization methods match and even sometimes outperform the dense baseline transfer performance. Note that this fact is valid not only in aggregate, but also at the level of each individual dataset.

In table C.1, we also include the linear transfer results for LTH-T. We note that the generally poor performance of the method, especially for more specialized tasks and higher sparsity levels, should not be taken as a criticism of the method itself: this use case is clearly contrary to the method's design, and the spirit of the original Lottery Ticket Hypothesis (which aims to discover masks with the intent to retrain, rather than final weights). Rather, we include these results to provide quantitative justification for the omission of LTH-T from any further analyses, and supporting the original authors' point that additional finetuning is *necessary* in order to obtain a competitive lottery ticket for transfer learning.

Additionally, we also validate our linear finetuning results by training with a different optimizer than SGD with momentum; namely, we use L-BFGS [47] and L_2 regularization. We tested multiple values for the L_2 regularization strength and we report for each dataset and method the highest value for the test accuracy. The results of this experiment are presented in Table C.3. Despite the differences in test accuracy between models trained with SGD or L-BFGS, we can observe a very similar trend related to the performance of sparse models over the dense baseline: namely, regularization pruning methods,

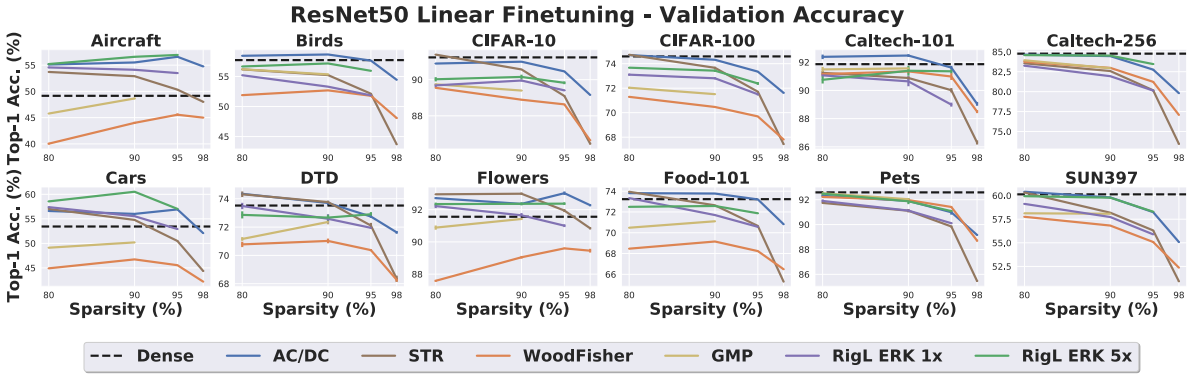

Figure C.1. (ResNet50) Per-dataset downstream validation accuracy for transfer learining with *linear finetuning*.

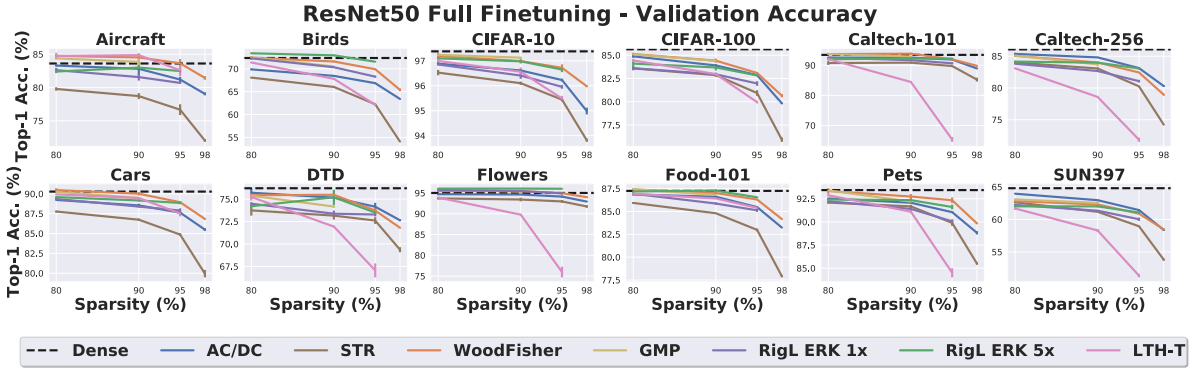

Figure C.2. (ResNet50) Per-dataset downstream validation accuracy for transfer learining with *full finetuning*.

such as AC/DC, STR or RigL, tend to be close to—or even outperform—the dense baseline, especially on fine-grained tasks (Aircraft, Cars).

C.2. Full Finetuning Results

Similarly to linear finetuning, we further provide complete results for full finetuning from sparse models. We present individual results per downstream task and pruning method, at different sparsity levels, in Figure C.2 and Table C.2; we report for each the average and standard deviation across 3 different trials. The results further support our conclusions from Section 3.4; namely, downstream task accuracy is correlated with the backbone sparsity, and progressive sparsification methods (GMP, WoodFisher) generally perform better than regularization methods.

D. Training Time Speed-up

The results presented in Section 3.6 are encouraging, suggesting that using features obtained from sparse models for linear finetuning can substantially reduce the training time, 2 and up to 3 times at 90% sparsity on the backbone model, with a small effect on the downstream accuracy, relative to the dense baseline. We additionally provide speed-up numbers for other sparsity levels. Table D.4 shows the average training time per epoch, as a fraction of the dense finetuning time, for the pruned models at different levels of sparsities. The numbers we show are computed for the Caltech-101 downstream task; however, they will most likely be very similar across all tasks, since the time required for inference on the backbone network dominates the time required for optimizing the linear classifier.

Furthermore, following [36], the accuracy for finetuning the linear classifier with pre-extracted features typically correlates very well with the accuracy of linear finetuning when data augmentation is used (as in this specific example). Therefore, based on the results from Figures C.1 and 4, we hypothesize that there are significant advantages when using sparse models for linear finetuning: firstly, potential improvements in transfer accuracy, compared to the dense backbone, and more notably, smaller memory footprint and training time savings.

Pruning Strategy	Dense	AC/DC	GMP	LTH-T	RigL ERK 1x	RigL ERK 5x	STR	WoodFisher
80% Sparsity								
Aircraft	49.2 ± 0.1	**55.1 ± 0.1**	45.8 ± 0.1	36.9 ± 0.1	54.6 ± 0.1	**55.2 ± 0.2**	53.7 ± 0.0	40.0 ± 0.2
Birds	57.7 ± 0.1	**58.4 ± 0.0**	56.2 ± 0.0	29.6 ± 0.1	55.2 ± 0.0	56.7 ± 0.1	56.2 ± 0.1	51.9 ± 0.1
CIFAR-10	91.2 ± 0.0	90.9 ± 0.0	89.7 ± 0.0	83.4 ± 0.1	89.7 ± 0.1	90.0 ± 0.1	**91.4 ± 0.0**	89.6 ± 0.0
CIFAR-100	**74.6 ± 0.1**	**74.7 ± 0.1**	72.0 ± 0.1	62.0 ± 0.1	73.1 ± 0.1	73.7 ± 0.0	**74.7 ± 0.0**	71.3 ± 0.0
Caltech-101	91.9 ± 0.1	**92.4 ± 0.2**	91.5 ± 0.2	75.4 ± 0.1	91.1 ± 0.1	90.8 ± 0.3	91.2 ± 0.1	91.2 ± 0.1
Caltech-256	**84.8 ± 0.1**	**84.6 ± 0.1**	83.9 ± 0.1	66.1 ± 0.1	83.3 ± 0.1	**84.6 ± 0.1**	83.6 ± 0.0	83.7 ± 0.1
Cars	53.4 ± 0.1	56.6 ± 0.0	49.1 ± 0.1	32.7 ± 0.1	57.4 ± 0.1	**58.6 ± 0.1**	57.0 ± 0.1	44.9 ± 0.1
DTD	73.5 ± 0.2	**74.4 ± 0.1**	71.2 ± 0.1	64.9 ± 0.2	73.5 ± 0.2	72.9 ± 0.3	**74.3 ± 0.2**	70.8 ± 0.2
Flowers	91.6 ± 0.1	92.7 ± 0.1	90.9 ± 0.1	85.6 ± 0.1	92.2 ± 0.1	92.3 ± 0.1	**93.0 ± 0.0**	87.6 ± 0.1
Food-101	73.2 ± 0.0	73.8 ± 0.0	70.5 ± 0.0	61.9 ± 0.0	73.3 ± 0.0	72.5 ± 0.1	**73.9 ± 0.0**	68.5 ± 0.0
Pets	**92.6 ± 0.1**	92.3 ± 0.1	**92.5 ± 0.1**	79.4 ± 0.1	91.9 ± 0.1	**92.5 ± 0.2**	91.7 ± 0.0	92.2 ± 0.1
SUN397	60.1 ± 0.0	**60.4 ± 0.0**	58.1 ± 0.0	47.4 ± 0.0	59.1 ± 0.1	59.9 ± 0.0	60.3 ± 0.0	57.8 ± 0.1
90% Sparsity								
Aircraft	49.2 ± 0.1	55.5 ± 0.1	48.7 ± 0.1	16.5 ± 0.2	54.1 ± 0.1	**56.6 ± 0.1**	52.9 ± 0.1	44.0 ± 0.2
Birds	57.7 ± 0.1	**58.7 ± 0.0**	55.4 ± 0.1	11.4 ± 0.1	53.3 ± 0.0	57.2 ± 0.1	55.2 ± 0.1	52.7 ± 0.1
CIFAR-10	**91.2 ± 0.0**	91.0 ± 0.0	89.4 ± 0.0	67.0 ± 0.1	90.0 ± 0.1	90.2 ± 0.1	90.6 ± 0.0	88.9 ± 0.0
CIFAR-100	**74.6 ± 0.1**	74.3 ± 0.0	71.5 ± 0.0	42.2 ± 0.1	72.8 ± 0.1	73.4 ± 0.1	73.7 ± 0.1	70.5 ± 0.0
Caltech-101	91.9 ± 0.1	**92.5 ± 0.1**	91.6 ± 0.1	49.0 ± 0.6	90.6 ± 0.3	91.4 ± 0.4	90.9 ± 0.1	91.3 ± 0.1
Caltech-256	**84.8 ± 0.1**	84.5 ± 0.0	82.9 ± 0.0	42.0 ± 0.1	81.9 ± 0.0	84.5 ± 0.1	82.6 ± 0.0	83.0 ± 0.1
Cars	53.4 ± 0.1	56.0 ± 0.1	50.2 ± 0.0	15.4 ± 0.1	55.5 ± 0.1	**60.5 ± 0.1**	54.8 ± 0.1	46.7 ± 0.0
DTD	**73.5 ± 0.2**	**73.7 ± 0.2**	72.4 ± 0.2	54.7 ± 0.1	72.6 ± 0.3	72.7 ± 0.2	**73.8 ± 0.1**	71.0 ± 0.2
Flowers	91.6 ± 0.1	92.4 ± 0.0	91.4 ± 0.1	67.7 ± 0.1	91.6 ± 0.1	92.4 ± 0.1	**93.0 ± 0.1**	89.0 ± 0.1
Food-101	73.2 ± 0.0	**73.8 ± 0.0**	71.1 ± 0.0	46.9 ± 0.0	71.7 ± 0.0	72.6 ± 0.0	72.6 ± 0.0	69.2 ± 0.0
Pets	**92.6 ± 0.1**	91.9 ± 0.1	92.0 ± 0.1	43.8 ± 0.2	91.1 ± 0.1	91.9 ± 0.2	91.1 ± 0.1	92.0 ± 0.1
SUN397	**60.1 ± 0.0**	59.8 ± 0.1	58.1 ± 0.0	31.7 ± 0.1	57.7 ± 0.0	59.8 ± 0.1	58.2 ± 0.0	56.8 ± 0.0
95% Sparsity								
Aircraft	49.2 ± 0.1	56.6 ± 0.1	N /A	4.5 ± 0.3	53.5 ± 0.1	**56.9 ± 0.1**	50.3 ± 0.1	45.6 ± 0.3
Birds	**57.7 ± 0.1**	**57.7 ± 0.0**	N /A	2.3 ± 0.1	51.9 ± 0.1	55.9 ± 0.0	52.1 ± 0.1	51.8 ± 0.1
CIFAR-10	**91.2 ± 0.0**	90.5 ± 0.0	N /A	39.9 ± 0.2	89.4 ± 0.0	89.8 ± 0.1	89.1 ± 0.0	88.6 ± 0.0
CIFAR-100	**74.6 ± 0.1**	73.4 ± 0.0	N /A	13.5 ± 0.2	71.5 ± 0.1	72.4 ± 0.1	71.7 ± 0.0	69.7 ± 0.0
Caltech-101	**91.9 ± 0.1**	91.6 ± 0.1	N /A	20.1 ± 0.5	89.0 ± 0.1	91.4 ± 0.1	90.0 ± 0.2	91.0 ± 0.2
Caltech-256	**84.8 ± 0.1**	82.8 ± 0.1	N /A	12.4 ± 0.3	80.1 ± 0.1	83.5 ± 0.1	80.2 ± 0.1	81.2 ± 0.1
Cars	53.4 ± 0.1	**56.9 ± 0.1**	N /A	3.9 ± 0.1	52.9 ± 0.0	**57.0 ± 0.1**	50.5 ± 0.1	45.5 ± 0.0
DTD	**73.5 ± 0.2**	72.7 ± 0.1	N /A	27.4 ± 0.2	71.9 ± 0.1	72.9 ± 0.2	72.1 ± 0.2	70.4 ± 0.1
Flowers	91.6 ± 0.1	**93.0 ± 0.1**	N /A	27.8 ± 0.6	91.0 ± 0.1	92.4 ± 0.1	91.9 ± 0.1	89.6 ± 0.0
Food-101	**73.2 ± 0.0**	**73.2 ± 0.0**	N /A	15.0 ± 0.1	70.6 ± 0.1	71.9 ± 0.0	70.7 ± 0.0	68.2 ± 0.0
Pets	**92.6 ± 0.1**	91.0 ± 0.2	N /A	15.9 ± 0.2	90.1 ± 0.1	91.1 ± 0.1	89.8 ± 0.1	91.4 ± 0.0
SUN397	**60.1 ± 0.0**	58.2 ± 0.0	N /A	8.4 ± 0.2	55.9 ± 0.1	58.3 ± 0.1	56.3 ± 0.0	55.1 ± 0.1
98% Sparsity								
Aircraft	49.2 ± 0.1	**54.8 ± 0.1**	N /A	N /A	N /A	N /A	48.0 ± 0.1	45.0 ± 0.1
Birds	**57.7 ± 0.1**	54.5 ± 0.0	N /A	N /A	N /A	N /A	43.7 ± 0.0	48.1 ± 0.1
CIFAR-10	**91.2 ± 0.0**	89.2 ± 0.0	N /A	N /A	N /A	N /A	86.5 ± 0.0	86.6 ± 0.0
CIFAR-100	**74.6 ± 0.1**	71.6 ± 0.0	N /A	N /A	N /A	N /A	67.4 ± 0.0	67.8 ± 0.0
Caltech-101	**91.9 ± 0.1**	89.0 ± 0.1	N /A	N /A	N /A	N /A	86.3 ± 0.1	88.5 ± 0.1
Caltech-256	**84.8 ± 0.1**	79.8 ± 0.0	N /A	N /A	N /A	N /A	73.4 ± 0.1	77.1 ± 0.0
Cars	**53.4 ± 0.1**	52.1 ± 0.0	N /A	N /A	N /A	N /A	44.4 ± 0.1	42.2 ± 0.0
DTD	**73.5 ± 0.2**	71.6 ± 0.1	N /A	N /A	N /A	N /A	68.4 ± 0.2	68.3 ± 0.1
Flowers	91.6 ± 0.1	**92.3 ± 0.1**	N /A	N /A	N /A	N /A	90.8 ± 0.1	89.5 ± 0.1
Food-101	**73.2 ± 0.0**	70.8 ± 0.0	N /A	N /A	N /A	N /A	65.3 ± 0.0	66.5 ± 0.0
Pets	**92.6 ± 0.1**	89.2 ± 0.1	N /A	N /A	N /A	N /A	85.5 ± 0.1	88.7 ± 0.1
SUN397	**60.1 ± 0.0**	55.1 ± 0.0	N /A	N /A	N /A	N /A	50.9 ± 0.0	52.4 ± 0.0

Table C.1. Transfer accuracy for sparse ResNet50 transfer with *linear finetuning*.

Pruning Strategy	Dense	AC/DC	GMP	LTH-T	RigL ERK 1x	RigL ERK 5x	STR	WoodFisher
80% Sparsity								
Aircraft	83.6 ± 0.4	83.3 ± 0.1	**84.4 ± 0.2**	**84.7 ± 0.5**	82.6 ± 0.3	82.4 ± 0.2	79.8 ± 0.3	**84.8 ± 0.2**
Birds	72.4 ± 0.3	69.9 ± 0.2	72.5 ± 0.2	71.4 ± 0.1	72.3 ± 0.3	**73.4 ± 0.1**	68.1 ± 0.1	72.4 ± 0.4
CIFAR-10	**97.4 ± 0.0**	96.9 ± 0.1	97.2 ± 0.0	97.0 ± 0.0	96.9 ± 0.0	97.1 ± 0.0	96.5 ± 0.1	97.2 ± 0.1
CIFAR-100	**85.6 ± 0.2**	84.9 ± 0.2	85.1 ± 0.0	84.4 ± 0.2	83.6 ± 0.2	84.1 ± 0.4	83.6 ± 0.2	85.1 ± 0.1
Caltech-101	**93.5 ± 0.1**	92.5 ± 0.2	**93.7 ± 0.5**	92.1 ± 0.5	92.5 ± 0.1	92.0 ± 0.3	90.7 ± 0.6	**93.7 ± 0.1**
Caltech-256	**86.1 ± 0.1**	85.4 ± 0.2	85.1 ± 0.2	83.1 ± 0.1	83.8 ± 0.1	84.2 ± 0.2	84.0 ± 0.1	85.1 ± 0.1
Cars	**90.3 ± 0.2**	89.2 ± 0.1	**90.3 ± 0.1**	89.9 ± 0.0	89.4 ± 0.1	89.6 ± 0.1	87.8 ± 0.1	**90.5 ± 0.2**
DTD	**76.2 ± 0.3**	75.7 ± 0.5	75.4 ± 0.1	75.2 ± 0.4	74.5 ± 0.2	74.2 ± 0.2	73.7 ± 0.6	75.4 ± 0.3
Flowers	95.0 ± 0.1	94.7 ± 0.2	95.9 ± 0.2	93.9 ± 0.2	95.7 ± 0.2	**96.1 ± 0.1**	93.7 ± 0.2	95.5 ± 0.2
Food-101	**87.3 ± 0.1**	86.9 ± 0.1	**87.4 ± 0.1**	86.9 ± 0.1	86.9 ± 0.1	**87.2 ± 0.1**	85.9 ± 0.1	**87.4 ± 0.1**
Pets	**93.4 ± 0.1**	92.5 ± 0.0	**93.4 ± 0.1**	92.9 ± 0.1	92.2 ± 0.1	92.4 ± 0.1	92.1 ± 0.1	**93.3 ± 0.3**
SUN397	**64.8 ± 0.0**	64.0 ± 0.0	63.1 ± 0.1	61.7 ± 0.2	62.2 ± 0.2	62.0 ± 0.3	62.6 ± 0.1	62.8 ± 0.1
90% Sparsity								
Aircraft	83.6 ± 0.4	82.8 ± 1.0	**83.9 ± 0.7**	**84.9 ± 0.3**	81.6 ± 0.5	83.0 ± 0.4	78.7 ± 0.4	**84.5 ± 0.4**
Birds	**72.4 ± 0.3**	68.5 ± 0.1	70.5 ± 0.1	67.8 ± 0.2	70.3 ± 0.0	**72.9 ± 0.2**	66.0 ± 0.2	71.6 ± 0.2
CIFAR-10	**97.4 ± 0.0**	96.6 ± 0.1	97.1 ± 0.0	96.6 ± 0.2	96.4 ± 0.1	97.0 ± 0.1	96.1 ± 0.1	97.0 ± 0.1
CIFAR-100	**85.6 ± 0.2**	83.9 ± 0.1	84.4 ± 0.0	83.0 ± 0.1	83.0 ± 0.2	83.7 ± 0.3	82.9 ± 0.2	84.4 ± 0.2
Caltech-101	**93.5 ± 0.1**	92.6 ± 0.2	92.9 ± 0.2	84.5 ± 0.3	91.7 ± 0.3	92.3 ± 0.4	90.9 ± 0.3	**93.9 ± 0.3**
Caltech-256	**86.1 ± 0.1**	84.8 ± 0.1	83.7 ± 0.3	78.6 ± 0.1	82.7 ± 0.2	84.0 ± 0.1	83.1 ± 0.2	84.0 ± 0.1
Cars	**90.3 ± 0.2**	88.5 ± 0.2	89.5 ± 0.0	89.5 ± 0.1	88.4 ± 0.1	89.2 ± 0.1	86.7 ± 0.2	**90.0 ± 0.2**
DTD	**76.2 ± 0.3**	75.2 ± 0.1	74.2 ± 0.1	71.9 ± 0.1	73.4 ± 0.4	75.2 ± 0.8	73.2 ± 0.4	75.5 ± 0.4
Flowers	95.0 ± 0.1	94.6 ± 0.1	95.3 ± 0.1	89.8 ± 0.2	95.5 ± 0.1	**96.1 ± 0.1**	93.4 ± 0.4	95.5 ± 0.3
Food-101	**87.3 ± 0.1**	86.6 ± 0.1	86.8 ± 0.1	86.4 ± 0.1	85.9 ± 0.1	**87.3 ± 0.2**	84.8 ± 0.0	**87.0 ± 0.1**
Pets	**93.4 ± 0.1**	92.1 ± 0.1	92.2 ± 0.1	91.1 ± 0.2	91.4 ± 0.2	92.3 ± 0.1	91.7 ± 0.2	92.7 ± 0.3
SUN397	**64.8 ± 0.0**	63.0 ± 0.0	62.5 ± 0.2	58.3 ± 0.2	61.3 ± 0.1	62.0 ± 0.2	61.2 ± 0.0	62.3 ± 0.1
95% Sparsity								
Aircraft	**83.6 ± 0.4**	81.2 ± 0.4	N /A	82.6 ± 0.8	80.7 ± 0.1	82.5 ± 0.4	76.7 ± 0.8	**83.6 ± 0.6**
Birds	**72.4 ± 0.3**	66.9 ± 0.1	N /A	62.2 ± 0.1	68.3 ± 0.2	71.6 ± 0.1	62.3 ± 0.1	69.9 ± 0.1
CIFAR-10	**97.4 ± 0.0**	96.2 ± 0.1	N /A	95.5 ± 0.1	96.0 ± 0.1	96.6 ± 0.1	95.4 ± 0.1	96.7 ± 0.1
CIFAR-100	**85.6 ± 0.2**	82.9 ± 0.1	N /A	80.0 ± 0.1	82.0 ± 0.2	82.8 ± 0.0	80.9 ± 0.3	83.1 ± 0.1
Caltech-101	**93.5 ± 0.1**	91.9 ± 0.2	N /A	65.3 ± 0.8	90.7 ± 0.4	92.2 ± 0.3	89.8 ± 0.1	92.0 ± 0.3
Caltech-256	**86.1 ± 0.1**	83.1 ± 0.0	N /A	71.8 ± 0.3	81.1 ± 0.2	83.1 ± 0.2	80.3 ± 0.0	82.4 ± 0.1
Cars	**90.3 ± 0.2**	87.6 ± 0.1	N /A	87.5 ± 0.4	87.9 ± 0.3	88.9 ± 0.2	84.9 ± 0.2	88.9 ± 0.2
DTD	**76.2 ± 0.3**	74.1 ± 0.4	N /A	67.1 ± 0.8	73.3 ± 0.2	73.5 ± 0.2	72.6 ± 0.4	73.7 ± 0.3
Flowers	95.0 ± 0.1	94.1 ± 0.3	N /A	76.0 ± 1.3	94.9 ± 0.3	**96.0 ± 0.0**	93.0 ± 0.3	95.0 ± 0.3
Food-101	**87.3 ± 0.1**	85.5 ± 0.0	N /A	85.4 ± 0.1	85.1 ± 0.2	86.6 ± 0.0	83.0 ± 0.1	86.3 ± 0.1
Pets	**93.4 ± 0.1**	91.0 ± 0.1	N /A	84.5 ± 0.5	90.1 ± 0.2	91.6 ± 0.3	89.9 ± 0.3	92.3 ± 0.3
SUN397	**64.8 ± 0.0**	61.4 ± 0.2	N /A	51.4 ± 0.3	60.0 ± 0.3	61.1 ± 0.2	59.0 ± 0.1	60.9 ± 0.1
98% Sparsity								
Aircraft	**83.6 ± 0.4**	79.1 ± 0.2	N /A	N /A	N /A	N /A	72.0 ± 0.2	81.4 ± 0.3
Birds	**72.4 ± 0.3**	63.4 ± 0.1	N /A	N /A	N /A	N /A	54.1 ± 0.1	65.4 ± 0.3
CIFAR-10	**97.4 ± 0.0**	95.0 ± 0.1	N /A	N /A	N /A	N /A	93.8 ± 0.1	96.0 ± 0.0
CIFAR-100	**85.6 ± 0.2**	79.8 ± 0.1	N /A	N /A	N /A	N /A	75.9 ± 0.2	80.7 ± 0.2
Caltech-101	**93.5 ± 0.1**	88.9 ± 0.1	N /A	N /A	N /A	N /A	85.2 ± 0.6	89.8 ± 0.3
Caltech-256	**86.1 ± 0.1**	80.3 ± 0.1	N /A	N /A	N /A	N /A	74.2 ± 0.0	78.9 ± 0.1
Cars	**90.3 ± 0.2**	85.5 ± 0.2	N /A	N /A	N /A	N /A	79.9 ± 0.5	86.8 ± 0.1
DTD	**76.2 ± 0.3**	72.6 ± 0.1	N /A	N /A	N /A	N /A	69.4 ± 0.3	71.8 ± 0.1
Flowers	**95.0 ± 0.1**	92.9 ± 0.1	N /A	N /A	N /A	N /A	91.8 ± 0.3	94.0 ± 0.2
Food-101	**87.3 ± 0.1**	83.2 ± 0.0	N /A	N /A	N /A	N /A	77.9 ± 0.1	84.2 ± 0.1
Pets	**93.4 ± 0.1**	88.8 ± 0.2	N /A	N /A	N /A	N /A	85.5 ± 0.1	89.8 ± 0.1
SUN397	**64.8 ± 0.0**	58.4 ± 0.1	N /A	N /A	N /A	N /A	53.8 ± 0.2	58.5 ± 0.1

Table C.2. Transfer accuracy for sparse ResNet50 transfer with *full finetuning*.

Pruning Strategy	Dense	AC/DC	GMP	RigL ERK 1x	RigL ERK 5x	STR	WoodFisher
80% Sparsity							
Aircraft	50.3	**56.7**	46.9	55.4	55.6	54.6	43.1
Birds	56.7	**57.7**	54.6	55.1	56.2	55.8	50.7
Caltech-101	91.8	**92.0**	91.2	91.5	91.2	91.4	91.3
Caltech-256	84.3	**84.6**	83.2	83.3	**84.6**	83.3	83.0
Cars	56.2	59.5	50.1	58.9	**60.4**	60.0	46.5
CIFAR-10	88.5	88.3	87.5	86.9	88.1	**88.9**	86.3
CIFAR-100	72.3	72.4	69.1	70.7	71.8	**72.9**	68.1
DTD	73.2	72.8	69.9	72.9	73.1	**73.3**	70.0
Flowers	92.9	**93.9**	92.0	93.3	93.3	**93.9**	89.0
Food-101	67.7	**68.6**	65.3	67.2	68.1	68.1	62.8
Pets	**92.5**	91.9	92.2	91.3	92.2	91.5	91.4
SUN397	58.5	59.3	56.4	58.0	**59.4**	**59.4**	55.8
90% Sparsity							
Aircraft	50.3	56.7	49.6	55.3	**57.4**	54.6	45.0
Birds	56.7	**57.7**	54.3	52.8	56.9	54.7	51.5
Caltech-101	91.8	**92.3**	91.0	90.5	91.5	90.4	91.2
Caltech-256	**84.3**	84.1	82.6	81.7	**84.7**	82.3	82.5
Cars	56.2	59.0	52.4	57.3	**62.0**	57.8	48.4
CIFAR-10	**88.5**	**88.5**	86.7	87.1	87.5	87.4	86.2
CIFAR-100	**72.3**	71.6	68.9	70.1	72.0	72.0	67.6
DTD	**73.2**	72.8	71.8	71.5	71.6	72.2	69.3
Flowers	92.9	93.4	92.7	92.6	93.3	**94.1**	90.2
Food-101	**67.7**	**67.7**	65.9	65.0	67.5	67.3	63.6
Pets	**92.5**	91.6	91.8	91.3	91.5	90.5	91.1
SUN397	58.5	58.2	56.3	56.9	**59.0**	57.2	54.6
95% Sparsity							
Aircraft	50.3	57.2	N /A	54.3	**57.4**	51.5	45.7
Birds	**56.7**	56.4	N /A	50.8	55.5	51.1	49.9
Caltech-101	**91.8**	91.6	N /A	89.4	91.7	89.9	90.6
Caltech-256	**84.3**	82.4	N /A	80.1	83.5	80.0	80.8
Cars	56.2	**59.4**	N /A	55.1	58.8	53.0	46.8
CIFAR-10	**88.5**	87.9	N /A	86.7	86.9	86.4	86.3
CIFAR-100	**72.3**	69.6	N /A	68.8	70.0	69.6	66.4
DTD	**73.2**	71.3	N /A	71.1	72.8	70.3	70.1
Flowers	92.9	**94.2**	N /A	92.3	93.5	93.1	90.8
Food-101	**67.7**	66.6	N /A	63.6	66.1	64.8	63.0
Pets	**92.5**	90.4	N /A	89.7	90.9	89.2	90.5
SUN397	**58.5**	56.8	N /A	54.9	57.7	54.8	52.7

Table C.3. Validation accuracy for sparse ResNet50 transfer with linear finetuning using the L-BFGS optimizer

Sparsity	STR	GMP	WoodFisher	AC/DC	RigL 5x
80%	0.44×	0.50×	0.53×	0.60×	0.71×
90%	0.28×	0.36×	0.37×	0.43×	0.50×
95%	0.22×	N /A	0.28×	0.32×	0.36×

Table D.4. Average training time per epoch for linear finetuning using sparse models, as a fraction of the time per epoch required for the dense backbone. The numbers shown are computed on the Caltech-101 dataset.

E. Sparse Convolutional Filters

In this section we illustrate the percentage of convolutional filters that are pruned during the training phase of the sparse ResNet50 models on ImageNet. The numbers presented in Table E.5 show that models pruned using AC/DC have considerably more sparse filters, compared to other sparse models. Similarly, RigL 5x models also have a significant number of sparse filters at high sparsity (95% sparsity).

Pruned Filters (%)	AC/DC	WoodFisher	GMP	STR	RigL ERK 1x	RigL ERK 5x
80%	**2.9%**	0.9%	1.6%	0.5%	0.2%	0.6%
90%	**8.5%**	2.0%	2.8%	2.0%	1.2%	2.7%
95%	**18%**	3.0%	N/A	6.0%	4.3%	9.1%

Table E.5. Percentage of convolutional filters that are completely masked out, for different pruning methods on ResNet50, at different sparsity levels. AC/DC has significantly more pruned filters.

F. Experiments on ResNet18 and ResNet34

In this section, we further validate our findings for linear finetuning from ResNet50 on two additional smaller architectures, namely ResNet18 and ResNet34. Specifically, we test whether regularization pruning methods generally have better transfer potential than progressive sparsification methods, and whether regularization pruning methods improve over dense models for fine-grained classification tasks. For this purpose, we trained AC/DC and GMP on ImageNet using ResNet18 and ResNet34 models, for 80% and 90% sparsity, using the same hyperparameters as for ResNet50. For both ResNet18 and ResNet34, there was a fairly large gap in ImageNet validation accuracy between GMP and AC/DC for both 80% and 90% sparsity, in favor of GMP, which almost recovered the baseline accuracy at 80% sparsity.

We show the results for linear finetuning using AC/DC and GMP in Table F.6 for ResNet18, respectively Table F.7 for ResNet34. Interestingly, despite the larger gap in ImageNet validation accuracy between GMP and AC/DC (with GMP being closer to the dense baseline), AC/DC tends to outperform GMP in terms of transfer performance, on most of the downstream tasks. Furthermore, we observe that AC/DC tends to transfer better than the dense baseline, especially for specialized or fine-grained downstream tasks. These observations confirm our findings for linear finetuning on ResNet50.

Pruning Strategy Task	Dense	GMP 80%	GMP 90%	AC/DC 80%	AC/DC 90%
Aircraft	47.7 ± 0.1	45.5 ± 0.1	45.6 ± 0.1	48.0 ± 0.1	$\mathbf{48.1 \pm 0.1}$
Birds	49.4 ± 0.1	49.3 ± 0.1	48.1 ± 0.0	$\mathbf{50.2 \pm 0.0}$	48.7 ± 0.1
CIFAR-10	87.2 ± 0.0	$\mathbf{87.4 \pm 0.0}$	87.2 ± 0.0	$\mathbf{87.4 \pm 0.0}$	87.2 ± 0.1
CIFAR-100	68.9 ± 0.0	68.1 ± 0.0	69.1 ± 0.0	$\mathbf{69.6 \pm 0.1}$	68.9 ± 0.0
Caltech-101	89.4 ± 0.3	$\mathbf{89.8 \pm 0.3}$	88.6 ± 0.2	89.0 ± 0.2	88.2 ± 0.4
Caltech-256	$\mathbf{79.4 \pm 0.1}$	78.3 ± 0.1	77.3 ± 0.1	78.8 ± 0.1	77.3 ± 0.1
Cars	45.6 ± 0.1	45.0 ± 0.1	44.4 ± 0.1	46.2 ± 0.1	$\mathbf{46.7 \pm 0.1}$
DTD	68.1 ± 0.1	68.2 ± 0.3	66.9 ± 0.2	$\mathbf{68.6 \pm 0.2}$	68.4 ± 0.2
Flowers	89.0 ± 0.1	89.3 ± 0.1	89.3 ± 0.1	89.9 ± 0.1	$\mathbf{90.2 \pm 0.1}$
Food-101	64.9 ± 0.0	65.0 ± 0.0	64.6 ± 0.0	$\mathbf{65.6 \pm 0.0}$	65.3 ± 0.0
Pets	$\mathbf{90.1 \pm 0.1}$	89.8 ± 0.1	89.4 ± 0.2	89.7 ± 0.1	89.4 ± 0.1
SUN397	$\mathbf{54.8 \pm 0.1}$	53.8 ± 0.1	52.9 ± 0.1	$\mathbf{54.8 \pm 0.1}$	53.5 ± 0.1

Table F.6. Transfer accuracy for different pruning methods for *linear finetuning* on ResNet18

G. Experiments on MobileNetV1

The MobileNet [31] architecture is a natural choice for devices with limited computational resources. We measure the results of sparse transfer with full and linear finetuning on the same downstream tasks starting from dense ImageNet models pruned using regularization-based and progressive sparsification methods. Specifically, we use AC/DC, STR for regularization methods and M-FAC [16] for the progressive sparsification category.

Pruning Strategy Task	Dense	GMP 80%	GMP 90%	AC/DC 80%	AC/DC 90%
Aircraft	45.8 ± 0.2	43.5 ± 0.2	44.9 ± 0.1	48.7 ± 0.1	**50.7 ± 0.2**
Birds	52.9 ± 0.0	53.0 ± 0.1	53.0 ± 0.1	**54.5 ± 0.1**	54.2 ± 0.1
CIFAR-10	89.5 ± 0.0	89.1 ± 0.0	88.5 ± 0.0	**89.6 ± 0.0**	89.0 ± 0.0
CIFAR-100	71.0 ± 0.0	70.4 ± 0.1	70.2 ± 0.1	**72.0 ± 0.0**	**72.0 ± 0.0**
Caltech-101	**92.5 ± 0.2**	91.8 ± 0.3	90.9 ± 0.2	92.0 ± 0.3	91.8 ± 0.4
Caltech-256	82.2 ± 0.1	81.8 ± 0.0	81.4 ± 0.1	**82.3 ± 0.1**	81.2 ± 0.1
Cars	47.3 ± 0.1	46.0 ± 0.1	45.6 ± 0.1	48.5 ± 0.1	**49.0 ± 0.1**
DTD	69.5 ± 0.1	68.6 ± 0.5	68.6 ± 0.2	**70.4 ± 0.3**	69.6 ± 0.2
Flowers	88.1 ± 0.1	88.5 ± 0.1	89.0 ± 0.1	90.0 ± 0.1	**91.1 ± 0.1**
Food-101	66.8 ± 0.0	66.7 ± 0.0	67.4 ± 0.0	68.2 ± 0.0	**68.8 ± 0.0**
Pets	92.0 ± 0.1	**92.5 ± 0.1**	91.4 ± 0.1	91.7 ± 0.1	91.1 ± 0.2
SUN397	55.9 ± 0.1	55.4 ± 0.1	55.0 ± 0.1	**56.8 ± 0.1**	55.6 ± 0.1

Table F.7. Transfer accuracy for different pruning methods for *linear finetuning* on ResNet34

M-FAC is a framework for efficiently computing high-dimensional inverse-Hessian vector products, which can be applied to different scenarios that use second-order information. In particular, one such instance is pruning, where M-FAC aims to solve the same optimization problem as WoodFisher, and thus from this point of view these methods are very similar. In particular, it has been shown [16] that M-FAC outperforms WoodFisher on ImageNet models, in terms of accuracy at a given sparsity level. Specifically, for MobileNet, M-FAC surpasses all existing methods at 90% sparsity, reaching 67.2% validation accuracy. For this reason, we included M-FAC, in favor of WoodFisher, to our list of progressive sparsification methods for MobileNetV1.

Due to the smaller size of the MobileNetV1 architecture, we additionally test the effect that lower sparsity levels have on the transfer performance, by training on ImageNet AC/DC models at 30%, 40% and 50% sparsity; these models fully recover the dense baseline accuracy on ImageNet.

The results on MobileNet are presented in Figure G.3 and Table G.8 for linear finetuning and Figure G.4 and Table G.9 for full finetuning. The results for linear finetuning are obtained after running from five different random seeds, and the mean and standard deviation are reported. However, the experiments for full finetuning were each run once. For both linear and full finetuning, we observe that generally the performance decays faster with increased sparsity, compared to ResNet50; this is expected, given the lower parameter count for MobileNet and the larger gap in ImageNet validation accuracy between dense and sparse models.

For linear finetuning, we observe AC/DC outperforms STR at both 75% and 90% sparsity. Furthermore, AC/DC tends to be close to M-FAC at 75% sparsity, while at 90% sparsity M-FAC performs better on almost half of the tasks. Differently from ResNet50, for MobileNet neither regularization based nor progressive sparsification models outperform the dense baseline, at higher sparsity (75% and 90%). We observe at lower sparsity (30% and 50%) a few instances where sparse models slightly outperform the dense baseline (Birds, Cars, DTD), but generally the differences are not significant.

In the case of full finetuning, we observe that the performance of sparse models decays more quickly than for ResNet50, and even at lower sparsity (30-50%) there is a gap in transfer performance compared to the dense baseline. Furthermore, AC/DC outperforms STR and M-FAC at both 75% and 90% sparsity on all downstream tasks. Overall, the results for MobileNet indicate that the transfer performance is significantly affected by the sparsity of the backbone model, for both linear and full finetuning. Moreover, the experiments on MobileNet seem to suggest that although some of the conclusions derived from the ResNet experiments are confirmed (e.g. sparse models usually have similar or slightly better performance to the dense baseline for linear finetuning), the guidelines for the preferred sparsity method in a given scenario might be specific to the choice of the backbone architecture.

Finally, we consider the accuracy tradeoff of using a smaller network such as MobileNet (4.2M trainable weights) versus a larger model, ResNet50 (25.5M trainable weights), but pruned to 90% sparsity. Below, we present linear and full finetuning accuracy results for these two scenarios for an easier comparison. We use the overall best pruning strategy for each type of transfer on ResNet50: AC/DC for linear finetuning and WoodFisher for full finetuning. Note that these same results are also presented in Tables C.2 and C.1, G.9, and G.8.

We observe that generally, pruning ResNet50 to 80 or even 90% sparsity results in higher accuracy than MobileNet, for

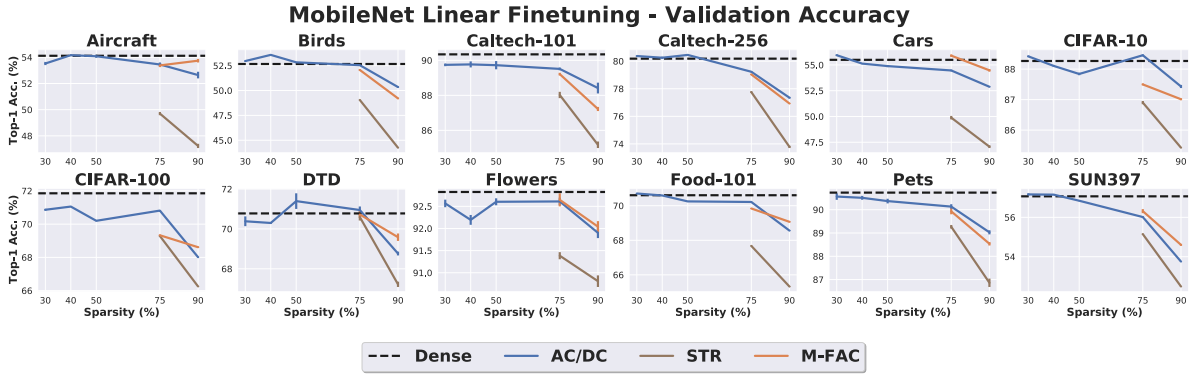

Figure G.3. (MobileNetV1) Per-dataset downstream validation accuracy for transfer learining with linear finetuning.

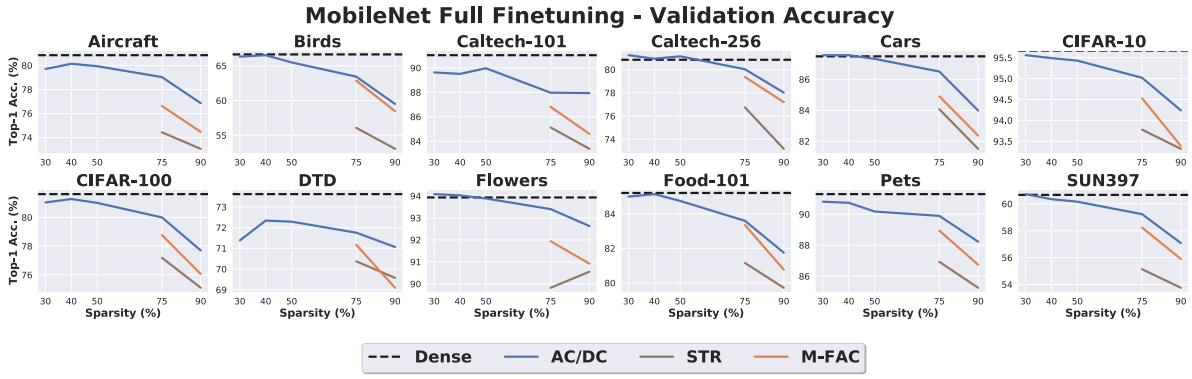

Figure G.4. (MobileNetV1) Per-dataset downstream validation accuracy for transfer learining with full finetuning.

Pruning Strategy Task	Dense	AC/DC 30%	AC/DC 40%	AC/DC 50%	AC/DC 75%	AC/DC 90%	M-FAC 75%	M-FAC 89%	STR 75%	STR 90%
Aircraft	54.1 ± 0.2	53.5 ± 0.1	**54.2 ± 0.1**	54.1 ± 0.1	53.5 ± 0.2	52.6 ± 0.3	53.4 ± 0.1	53.7 ± 0.1	49.7 ± 0.1	47.2 ± 0.2
Birds	52.7 ± 0.1	53.0 ± 0.1	**53.6 ± 0.1**	52.8 ± 0.0	52.6 ± 0.1	50.3 ± 0.1	52.1 ± 0.1	49.2 ± 0.1	49.0 ± 0.0	44.2 ± 0.0
CIFAR-10	88.3 ± 0.1	88.4 ± 0.0	88.1 ± 0.0	87.8 ± 0.0	**88.5 ± 0.0**	87.4 ± 0.1	87.5 ± 0.0	87.0 ± 0.0	86.9 ± 0.1	85.4 ± 0.0
CIFAR-100	**71.9 ± 0.0**	70.9 ± 0.1	71.1 ± 0.0	70.2 ± 0.0	70.8 ± 0.0	68.0 ± 0.0	69.3 ± 0.0	68.6 ± 0.0	69.3 ± 0.0	66.3 ± 0.0
Caltech-101	**90.3 ± 0.1**	89.7 ± 0.1	89.8 ± 0.2	89.7 ± 0.2	89.5 ± 0.1	88.4 ± 0.3	89.2 ± 0.1	87.2 ± 0.1	88.0 ± 0.2	85.2 ± 0.2
Caltech-256	80.2 ± 0.1	80.4 ± 0.0	80.2 ± 0.1	**80.5 ± 0.0**	79.2 ± 0.0	77.3 ± 0.1	79.0 ± 0.0	76.9 ± 0.0	77.8 ± 0.1	73.8 ± 0.1
Cars	55.5 ± 0.0	**55.9 ± 0.1**	55.1 ± 0.1	54.9 ± 0.1	54.5 ± 0.1	52.9 ± 0.0	**55.9 ± 0.1**	54.5 ± 0.1	49.9 ± 0.2	47.1 ± 0.1
DTD	70.8 ± 0.2	70.4 ± 0.2	70.3 ± 0.0	**71.4 ± 0.4**	70.9 ± 0.2	68.8 ± 0.1	70.7 ± 0.2	69.6 ± 0.2	70.6 ± 0.2	67.2 ± 0.1
Flowers	**92.8 ± 0.1**	92.6 ± 0.1	92.2 ± 0.1	92.6 ± 0.1	92.6 ± 0.1	91.9 ± 0.1	92.6 ± 0.1	92.0 ± 0.1	91.4 ± 0.1	90.8 ± 0.1
Food-101	70.6 ± 0.0	**70.7 ± 0.0**	70.6 ± 0.0	70.3 ± 0.0	70.2 ± 0.0	68.6 ± 0.0	69.8 ± 0.0	69.1 ± 0.0	67.7 ± 0.0	65.3 ± 0.0
Pets	**90.7 ± 0.1**	90.6 ± 0.1	90.5 ± 0.1	90.4 ± 0.1	90.1 ± 0.1	89.0 ± 0.1	89.9 ± 0.1	88.5 ± 0.1	89.3 ± 0.1	86.9 ± 0.2
SUN397	57.1 ± 0.0	**57.2 ± 0.1**	57.2 ± 0.0	56.8 ± 0.0	56.0 ± 0.0	53.8 ± 0.0	56.3 ± 0.1	54.6 ± 0.0	55.1 ± 0.0	52.5 ± 0.0

Table G.8. Transfer accuracy for *linear finetuning* using sparse MobileNet models

both linear and full finetuning. However, in almost all cases, the gap is below 5%. This finding confirms conventional wisdom that training and pruning large networks generally results in higher accuracy than training dense small networks from scratch.

H. Impact of fully connected layer bias on full finetuning transfer accuracy

In our experiments, we used the original architectures used to train the upstream ImageNet models when performing transfer with full-finetuning, only resizing the final layer to match the number of output classes in the downstream task. This choice was necessitated partially by ensuring that the weights were applied correctly. For example, the RigL models were trained using TensorFlow, which uses slightly different Convolution and MaxPooling padding conventions than PyTorch.

Pruning Strategy	Dense	AC/DC 30%	AC/DC 40%	AC/DC 50%	AC/DC 75%	AC/DC 90%	M-FAC 75%	M-FAC 89%	STR 75%	STR 90%
Aircraft	**80.9**	79.7	80.1	79.9	79.0	76.9	76.6	74.5	74.4	73.0
Birds	**66.6**	66.3	66.5	65.5	63.4	59.5	62.9	58.5	56.1	53.1
CIFAR-10	**95.7**	95.6	95.5	95.4	95.0	94.2	94.5	93.4	93.8	93.3
CIFAR-100	**81.6**	81.0	81.3	81.0	80.0	77.7	78.8	76.1	77.2	75.1
Caltech-101	**91.0**	89.6	89.5	90.0	88.0	87.9	86.8	84.6	85.1	83.4
Caltech-256	80.9	**81.2**	80.9	81.1	80.0	78.0	79.4	77.2	76.7	73.2
Cars	87.5	**87.6**	**87.6**	87.3	86.5	84.0	84.9	82.4	84.1	81.5
DTD	**73.6**	71.4	72.3	72.3	71.8	71.1	71.2	69.1	70.4	69.6
Flowers	93.9	**94.1**	94.0	93.9	93.4	92.6	91.9	90.9	89.8	90.6
Food-101	**85.2**	85.0	85.1	84.7	83.6	81.8	83.4	80.8	81.2	79.7
Pets	**91.3**	90.8	90.7	90.2	89.9	88.2	88.9	86.7	86.9	85.3
SUN397	**60.7**	**60.7**	60.3	60.2	59.2	57.1	58.2	55.9	55.1	53.8

Table G.9. Transfer accuracy for *full finetuning* using sparse MobileNet models

Model	MobileNet Dense	ResNet50 AC/DC 80%	ResNet50 AC/DC 90%
Aircraft	54.1 ± 0.2	55.1 ± 0.1	55.5 ± 0.1
Birds	52.7 ± 0.1	58.4 ± 0.0	58.7 ± 0.0
CIFAR-10	88.3 ± 0.1	90.9 ± 0.0	91.0 ± 0.0
CIFAR-100	71.9 ± 0.0	74.7 ± 0.1	74.3 ± 0.0
Caltech-101	90.3 ± 0.1	92.4 ± 0.2	92.5 ± 0.1
Caltech-256	80.2 ± 0.1	84.6 ± 0.1	84.5 ± 0.0
Cars	55.5 ± 0.0	56.6 ± 0.0	56.0 ± 0.1
DTD	70.8 ± 0.2	74.4 ± 0.1	73.7± 0.2
Flowers	92.8 ± 0.1	92.7 ± 0.1	92.4 ± 0.0
Food-101	70.6 ± 0.0	73.8 ± 0.0	73.8 ± 0.0
Pets	90.7 ± 0.1	92.3 ± 0.1	91.9 ± 0.1
SUN397	57.1 ± 0.0	60.4 ± 0.0	59.8 ± 0.1

Table G.10. Comparison of MobileNet dense versus ResNet50 sparse models when transferring with *linear finetuning*

Model	MobileNet Dense	ResNet50 WoodFisher 80%	ResNet50 WoodFisher 90%
Aircraft	80.9	84.8 ± 0.2	84.5 ± 0.4
Birds	66.6	72.4 ± 0.4	71.6 ± 0.2
CIFAR-10	95.7	97.2 ± 0.1	97.0 ± 0.1
CIFAR-100	81.6	85.1 ± 0.1	84.4 ± 0.2
Caltech-101	91.0	93.7 ± 0.1	93.9 ± 0.3
Caltech-256	80.9	85.1 ± 0.1	84.0± 0.1
Cars	87.5	90.5 ± 0.2	90.0 ± 0.2
DTD	73.6	75.4 ± 0.3	75.5 ± 0.4
Flowers	93.9	95.5 ± 0.2	95.5 ± 0.3
Food-101	85.2	87.4 ± 0.1	87.0 ± 0.1
Pets	91.3	93.3 ± 0.3	92.7 ± 0.3
SUN397	60.7	62.8 ± 0.1	62.3 ± 0.1

Table G.11. Comparison of MobileNet dense versus ResNet50 sparse models when transferring with *full finetuning*

Likewise, STR models were trained using a slightly nonstandard PyTorch implementation of ResNet50, which did not use a bias term in the final Fully-Connected (FC) layer. We investigate the possibility that the latter difference could have an effect on downstream transfer accuracy. To do so, we transferred a set of 80% sparse ResNet50 STR models to all downstream tasks, using a bias term in the FC layer. The results are shown in Table H.12. Additionally, we perform a similar comparison on MobileNetV1, for STR models at 75% sparsity. As in the case of ResNet50, the version of MobileNet used by the STR models does not use bias in the final classification layer. The results illustrating the bias effect on full finetuning for MobileNet are presented in Table H.13. We observe that the presence of a bias term in the final layer can, in some cases, have a small positive effect on the resulting model, and so we caution that these effects be considered when choosing a transfer architecture.

I. Impact of label smoothing on transfer accuracy

We take advantage of the fact that we have STR checkpoints trained with and without label smoothing (LS) to investigate the effect of LS on dense and sparse transfer accuracy in the context of linear transfer. As Table I.14 shows, label smoothing tends to have a negative effect on transfer accuracy (confirming the results in [36]). However, our experiments suggest that this effect is more pronounced on the Aircraft and Cars datasets in the case of sparse STR models, and generally for most specialized datasets for the dense models. Furthermore, we observe that the performance gap tends to narrow with increased sparsity. We also note that even with label smoothing, at 80% sparsity STR matches or outperforms GMP on all datasets, although the effect largely reverses at 90% sparsity.

Dataset	With FC Bias	Without FC Bias
Aircraft	79.8 ± 0.6	79.8 ± 0.3
Birds	67.9 ± 0.2	68.1 ± 0.1
CIFAR-10	96.5 ± 0	96.5 ± 0.1
CIFAR-100	83.7 ± 0.2	83.6 ± 0.2
Caltech-101	91.2 ± 0.2	90.7 ± 0.6
Caltech-256	84.4 ± 0.1	84.0 ± 0.1
Cars	87.7 ± 0.1	87.8 ± 0.1
DTD	74.4 ± 0.2	73.7 ± 0.6
Flowers	94 ± 0.1	93.7 ± 0.2
Food-101	86 ± 0.1	85.9 ± 0.1
Pets	92.1 ± 0.1	92.1 ± 0.1
SUN397	63.2 ± 0.1	62.6 ± 0.1

Table H.12. Top-1 validation accuracy on ResNet50 trained using STR on ImageNet, with using bias in the FC layer versus without. The original model architecture does not use bias in the FC layer.

Dataset	With FC Bias	Without FC Bias
Aircraft	74.2	74.4
Birds	56.4	56.1
CIFAR-10	93.9	93.8
CIFAR-100	77.5	77.2
Caltech-101	86.3	85.1
Caltech-256	76.9	76.7
Cars	83.8	84.1
DTD	71.8	70.4
Flowers	90.4	89.8
Food-101	80.9	81.2
Pets	87.9	86.9
SUN397	56.1	55.1

Table H.13. Top-1 validation accuracy on MobileNetV1 trained using STR on ImageNet, with bias in the FC layer versus without. The original model architecture does not use bias in the FC layer.

Overall, these data can be taken as a preliminary confirmation of the importance of controlling for variation in hyperparameters when comparing the transfer performance of various training and pruning methods.

Dataset	Dense	Dense LS	STR 80%	STR LS 80%	STR 90%	STR LS 90%	STR 95%	STR LS 95%	STR 98%	STR LS 98%
Aircraft	49.2 ± 0.1	38.2 ± 0.1	53.7 ± 0.0	47.0 ± 0.0	52.9 ± 0.1	46.4 ± 0.1	50.3 ± 0.1	46.6 ± 0.1	48.0 ± 0.1	45.2 ± 0.1
Birds	57.7 ± 0.1	52.4 ± 0.0	56.2 ± 0.1	56.4 ± 0.0	55.2 ± 0.1	56.0 ± 0.0	52.1 ± 0.1	51.7 ± 0.1	43.7 ± 0.0	45.6 ± 0.0
CIFAR-10	91.2 ± 0.0	89.6 ± 0.0	91.4 ± 0.0	90.1 ± 0.0	90.6 ± 0.0	89.4 ± 0.0	89.1 ± 0.0	88.6 ± 0.0	86.5 ± 0.0	86.0 ± 0.0
CIFAR-100	74.6 ± 0.1	71.6 ± 0.0	74.7 ± 0.0	73.3 ± 0.0	73.7 ± 0.1	72.2 ± 0.1	71.7 ± 0.0	70.1 ± 0.0	67.4 ± 0.0	66.3 ± 0.0
Caltech-101	91.9 ± 0.1	91.6 ± 0.1	91.2 ± 0.1	92.6 ± 0.1	90.9 ± 0.1	91.1 ± 0.2	90.0 ± 0.2	89.8 ± 0.1	86.3 ± 0.1	85.4 ± 0.1
Caltech-256	84.8 ± 0.1	84.6 ± 0.1	83.6 ± 0.0	84.3 ± 0.0	82.6 ± 0.0	82.6 ± 0.1	80.2 ± 0.1	79.7 ± 0.0	73.4 ± 0.1	73.8 ± 0.0
Cars	53.4 ± 0.1	44.9 ± 0.1	57.0 ± 0.1	50.9 ± 0.0	54.8 ± 0.1	49.8 ± 0.1	50.5 ± 0.1	46.9 ± 0.1	44.4 ± 0.1	42.5 ± 0.1
DTD	73.5 ± 0.2	72.3 ± 0.1	74.3 ± 0.2	73.9 ± 0.3	73.8 ± 0.1	73.7 ± 0.2	72.1 ± 0.2	71.9 ± 0.1	68.4 ± 0.2	68.3 ± 0.1
Flowers	91.6 ± 0.1	86.7 ± 0.1	93.0 ± 0.0	91.2 ± 0.0	93.0 ± 0.1	92.1 ± 0.1	91.9 ± 0.1	91.0 ± 0.1	90.8 ± 0.1	90.4 ± 0.1
Food-101	73.2 ± 0.0	69.5 ± 0.0	73.9 ± 0.0	72.2 ± 0.0	72.6 ± 0.0	71.1 ± 0.0	70.7 ± 0.0	68.8 ± 0.0	65.3 ± 0.0	64.3 ± 0.0
Pets	92.6 ± 0.1	92.9 ± 0.1	91.7 ± 0.0	92.4 ± 0.1	91.1 ± 0.1	91.7 ± 0.1	89.8 ± 0.1	90.1 ± 0.1	85.5 ± 0.1	86.6 ± 0.1
SUN397	60.1 ± 0.0	59.3 ± 0.1	60.3 ± 0.0	60.0 ± 0.1	58.2 ± 0.0	58.5 ± 0.1	56.3 ± 0.0	55.8 ± 0.0	50.9 ± 0.0	51.0 ± 0.0

Table I.14. Linear Finetuning Validation Accuracy of STR-pruned and dense models with and without label smoothing.

J. Finetuning with structured sparsity

In this section, we examine the transfer properties of models that were sparsified using structured pruning methods, which remove entire convolutional filters. Specifically, we use both ResNet50 and MobileNetV1 models trained on ImageNet and we do full finetuning on all twelve downstream tasks.

J.1. ResNet50 with structured sparsity

We consider a ResNet50 model that was pruned with progressive sparsification, using the L_1 magnitude of the convolutional filters as a pruning criterion. The resulting model has an ImageNet validation accuracy of 75.7% and results in 2.2x inference speed-up compared to the dense baseline, when evaluated on a single sample; this makes it comparable to unstructured 90% sparse models that achieve a similar inference speed-up (please see Table D.4). The results for full finetuning with the structured sparse model, together with the best results for dense and unstructured 80% and 90% models are presented in Table J.15. We observe that models with structured sparsity transfer similarly to or worse than unstructured 90% sparse models. Note that the unstructured ResNet50 model has higher ImageNet accuracy compared to 90% sparse models, at a similar inference speed-up. These results align with the observations made in Section 3.5, that having fewer filters in the structured sparse models limits their capability of expressing features.

Dataset	Dense	Structured	Best 80%	Best 90%
Aircraft	83.6 ± 0.4	81.8 ± 0.5	**84.8± 0.2**	**84.9 ± 0.3**
Birds	72.4 ± 0.3	70.7 ± 0.1	**73.4± 0.1**	72.9 ± 0.2
Caltech101	**93.5± 0.1**	92.8 ± 0.1	**93.7± 0.1**	**93.9 ± 0.3**
Caltech256	**86.1 ± 0.1**	84.6 ± 0.1	85.4 ± 0.2	84.8 ± 0.1
Cars	90.3 ± 0.2	89.4 ± 0.0	**90.5 ± 0.2**	90.0 ± 0.2
CIFAR-10	**97.4 ± 0.**	97.1 ± 0.1	97.2 ± 0.1	$97.1 \pm 0.$
CIFAR-100	**85.6 ± 0.2**	84.7 ± 0.2	85.1 ± 0.1	84.4 ± 0.2
DTD	**76.2 ± 0.3**	75.2 ± 0.2	**75.7 ± 0.5**	75.5 ± 0.4
Flowers	95.0 ± 0.1	95.2 ± 0.0	**96.1 ± 0.1**	**96.1 ± 0.1**
Food-101	**87.3 ± 0.1**	86.3 ± 0.1	**87.4 ± 0.1**	**87.3 ± 0.2**
Pets	**93.4 ± 0.1**	92.5 ± 0.1	**93.4 ± 0.2**	92.7 ± 0.3
SUN397	**64.8 ± 0.**	63.4 ± 0.1	$64.0 \pm 0.$	$63.0 \pm 0.$

Table J.15. (ResNet50) Comparison on full finetuning between dense baseline, models with structured sparsity, and best results for unstructured 80% and 90% sparsity.

Dataset	Dense	50% Time	50% FLOPs
Aircraft	80.9	**82.9**	**83.0**
Birds	**66.6**	66.1	66.1
Caltech101	**91.0**	88.6	88.9
Caltech256	**80.9**	78.6	78.4
Cars	87.5	**88.4**	**88.3**
CIFAR-10	**95.7**	95.2	95.3
CIFAR-100	**81.6**	79.9	80.2
DTD	**73.6**	71.1	72.2
Flowers	93.9	**94.1**	**94.1**
Food-101	**85.2**	84.6	84.5
Pets	**91.3**	91.0	91.0
SUN397	**60.7**	59.4	59.1

Table J.16. (MobileNet) Full finetuning validation accuracy for MobileNet models with structured sparsity, at 50% inference time or 50% inference FLOPs.

Type	all	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
box	32.62	54.05	51.96	48.72	44.81	40.84	34.72	26.89	17.33	6.33	0.55
mask	30.74	50.28	47.66	44.57	41.02	36.39	31.47	25.53	18.55	10.03	1.91

Table K.17. Mean average precision for dense transfer on Pascal, at various thresholds.

Type	all	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
box	33.55	54.15	51.79	49.2	45.57	41.51	35.95	29.2	19.66	7.78	0.74
mask	31.5	50.66	47.89	45.04	41.67	37.32	32.39	26.35	19.98	11.22	2.5

Table K.18. Mean average precision for *sparse* transfer on Pascal, at various thresholds. Notice the similar or slightly improved accuracy.

J.2. MobileNet with structured sparsity

We additionally perform full finetuning using MobileNet models pruned for structured sparsity. For these experiments, we use the upstream models provided in [27]; specifically, we use the MobileNet models that achieve 50% of the inference time or have 50% of the dense FLOPs. These models achieve 70.2% and 70.5% ImageNet validation accuracy, respectively. The results presented in Table J.16 show that in general models with structured sparsity perform similar to or worse than their dense counterparts, with the exception of Aircraft and Cars where these models significantly outperform the dense baseline.

K. Sparse Transfer Learning for Segmentation

To complement the experiments for object detection, we executed transfer learning for a YOLACT model [3] using a ResNet-101 backbone, that has been trained and sparsified on the segmentation version of the COCO dataset. The average sparsity of the model is $\sim$ 87%, obtained via gradual magnitude pruning (GMP). The model has mAP@0.5 values 49.36 (bounding box), and 46.37 (mask), versus 50.16 (bounding box), 46.57 (mask) for the dense model on COCO. We transfer the pruned trained weights onto the Pascal dataset. The prediction heads get initialized as dense, and kept dense for transfer. The results are presented in Tables K.17 and K.18, and show that indeed sparse transfer is competitive against the dense variant in this case as well.

L. Distillation from Sparse Teachers

Our linear finetuning experiments suggest that sparse models may provide superior representations relative to dense ones. To further test this hypothesis, we employ sparse models as teachers in a standard knowledge distillation (KD) setting, i.e. training a ResNet34 student model with distillation from a ResNet50 teacher, which may be dense or sparse. The accuracy of the resulting models is provided in Table L.19.

Results suggest that differences in accuracy between the sparse and dense teachers do not affect distillation. Sparse teachers will also reduce distillation overhead due to faster inference.

Baseline	Dense KD	AC/DC 80%	WoodFisher 80%	AC/DC 90%	WoodFisher 90%
73.83%	74.42%	74.64%	74.63%	74.19%	74.44%

Table L.19. Top-1 validation accuracy on ResNet34 trained on ImageNet, when distilling from dense or sparse teachers.