10 EXTENDED APPENDIX

10.1 REDS performance on DNN and CNN architectures

In addition to the results on DS-CNN reported in the main paper, we show in Table 4 and Table 5 REDS performance on DNN and CNN architectures (with full fine-tuning) and compare to training model of each capacity from scratch and training REDS from scratch. Despite full fine-tuning, the results for S architecture show superior performance of the BU heuristic over TD.

MACs	Small (S) - Accuracy 83.82				Large (L) - Accuracy 86.87			
	Scratch	Knapsack BU	Knapsack TD	REDS training	Scratch	Knapsack BU	Knapsack TD	REDS training
100%	84.30 ±0.11	83.52 ±0.07	82.80 ±0.16	82.13 ±0.20	86.54 ±0.24	86.46 ±0.34	86.25±0.19	85.06 ±0.19
75%	83.77 ±0.23	82.29 ±0.35	81.88 ±0.30	81.23 ±0.21	85.96 ±0.13	86.38 ±0.65	86.09±0.03	84.93 ±0.20
50%	80.91 ±0.11	78.36 ±1.40	78.59 ±0.24	77.05 ±0.34	85.24 ±0.35	85.62 ±0.24	85.58±0.35	84.08 ±0.22
25%	69.77 ±0.67	64.42 ±1.99	61.43 ±3.35	63.69 ±3.26	84.22 ±0.13	82.61 ±0.61	83.00 ±0.62	82.03 ±0.59

Table 4: Test set accuracy [%] of training S and L fully-connected (DNN) architectures taken from [65]: training a network of each size from scratch ("Scratch"), conversion from a pre-trained network using two knapsack versions ("Knapsack BU" and "Knapsack TD"), and training REDS structure from scratch ("REDS training"). Reported results from three independent runs. The accuracy of each 100 % network reported in [65] is listed in the header row.

MACs	Small (S) - Accuracy 92.24				Large (L) - Accuracy 93.24			
	Scratch	Knapsack BU	Knapsack TD	REDS training	Scratch	Knapsack BU	Knapsack TD	REDS training
100%	91.10±0.23	91.60±0.39	91.20 ±0.35	88.89 ±0.26	92.94±0.20	92.83±0.26	92.97±0.15	90.97 ±0.21
75%	90.40±0.27	90.63±0.19	90.20 ±0.13	87.64 ±0.46	92.74±0.12	92.74±0.23	92.54 ±0.07	90.67 ±0.09
50%	89.07±0.24	88.39±0.31	88.39 ±0.52	85.99 ±0.19	92.44±0.30	91.95±0.22	91.93 ±0.28	90.36 ±0.30
25%	82.57±0.41	79.52±0.67	79.28 ±0.54	79.25 ±0.40	90.98±0.60	90.24±0.35	90.31 ±0.03	88.25 ±0.66

Table 5: The same as in Table 4 for S and L convolutional architectures (CNN) from [65].

Fig. 8 shows the impact of the architecture on REDS structure found by the knapsack BU solver. We present the results for DNN S, L and CNN S, L from left to right, respectively.

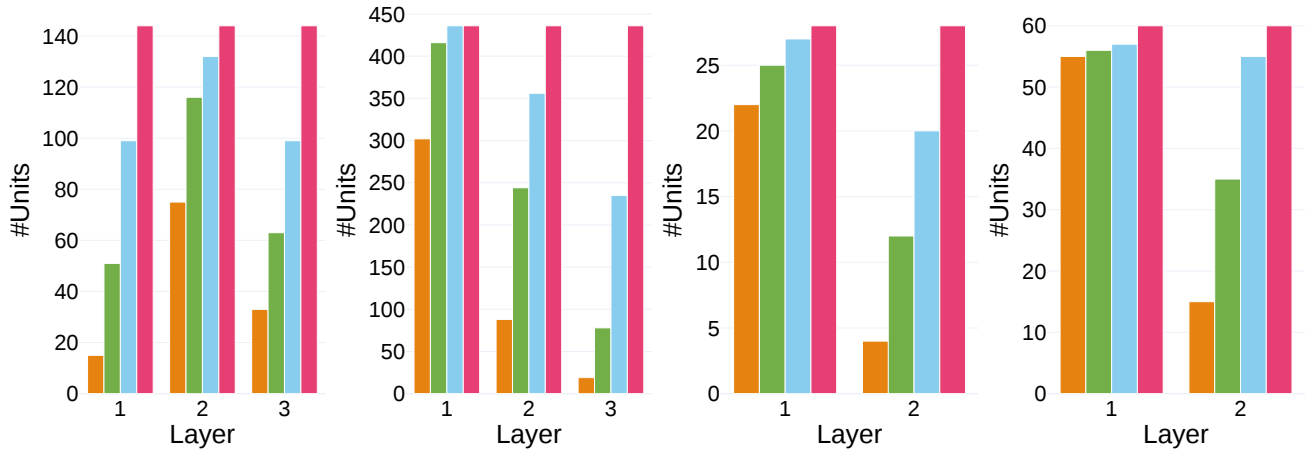

Figure 8: Analysis of the subnetwork architecture obtained by the knapsack BU heuristics. From left to right: DNN S, DNN L, CNN S and CNN L on Google Speech Commands. The patterns as to which computational units constitute a child subnetwork are architecture-specific.

10.2 REDS performance with 10 nested subnetworks

Table 6 and Fig. 9 show the performance of DNN, CNN and DS-CNN on Google Speech Commands, when REDS structure comprises 10 subnetworks, compared to 4 subnetworks in the main paper. A larger number of subnetworks does not degrade model accuracy.

MACs	DNN		CNN		DS-CNN	
	Small	Large	Small	Large	Small	Large
100%	83.07 ±0.35	86.19 ±0.26	91.16 ±0.45	93.1 ±0.1	93.5 ±0.15	94.34 ±0.07
90%	82.93 ±0.4	86.17 ±0.36	90.4 ±0.47	92.84 ±0.27	93.33 ±0.11	94.32 ±0.1
80%	82.67 ±0.53	86.1 ±0.08	90.1 ±0.07	92.77 ±0.06	92.84 ±0.15	94.31 ±0.1
70%	81.67 ±0.53	85.78 ±0.11	89.43 ±0.41	92.25 ±0.47	92.64 ±0.24	94.21 ±0.14
60%	80.37 ±0.57	85.66 ±0.25	88.84 ±0.25	92.28 ±0.11	92.27 ±0.31	94.08 ±0.03
50%	78.26 ±0.53	85.33 ±0.09	88.4 ±0.2	92.03 ±0.23	91.04 ±0.51	93.97 ±0.04
40%	75.37 ±1.26	84.8 ±0.45	85.76 ±0.18	91.78 ±0.04	89.42 ±0.66	93.83 ±0.22
30%	67.76 ±1.59	82.66 ±0.18	81.92 ±0.49	90.52 ±0.54	87.63 ±1.03	93.59 ±0.14
20%	52.36 ±6.99	80.19 ±0.39	73.74 ±0.04	88.61 ±0.51	84.14 ±2.57	93.35 ±0.15
10%	23.93 ±5.88	50.7 ±7.5	58.87 ±5.27	81.15 ±1.39	58.46 ±3.35	90.38 ±0.17

Table 6: Test set accuracy [%] from Small (S) and Large (L) pretrained fully-connected (DNN), convolutional (CNN), and depth-wise separable convolutional (DS-CNN) networks taken from [65]. For each pre-trained architecture, REDS can support ten subnetworks obtained from the Knapsack BU formulation. The accuracies of the DS-CNN and CNN subnetworks do not degrade drastically until the lowest percentage of MACs considered. In contrast, the accuracies in the DNN subnetworks show a more pronounced drop from 30% MACs.

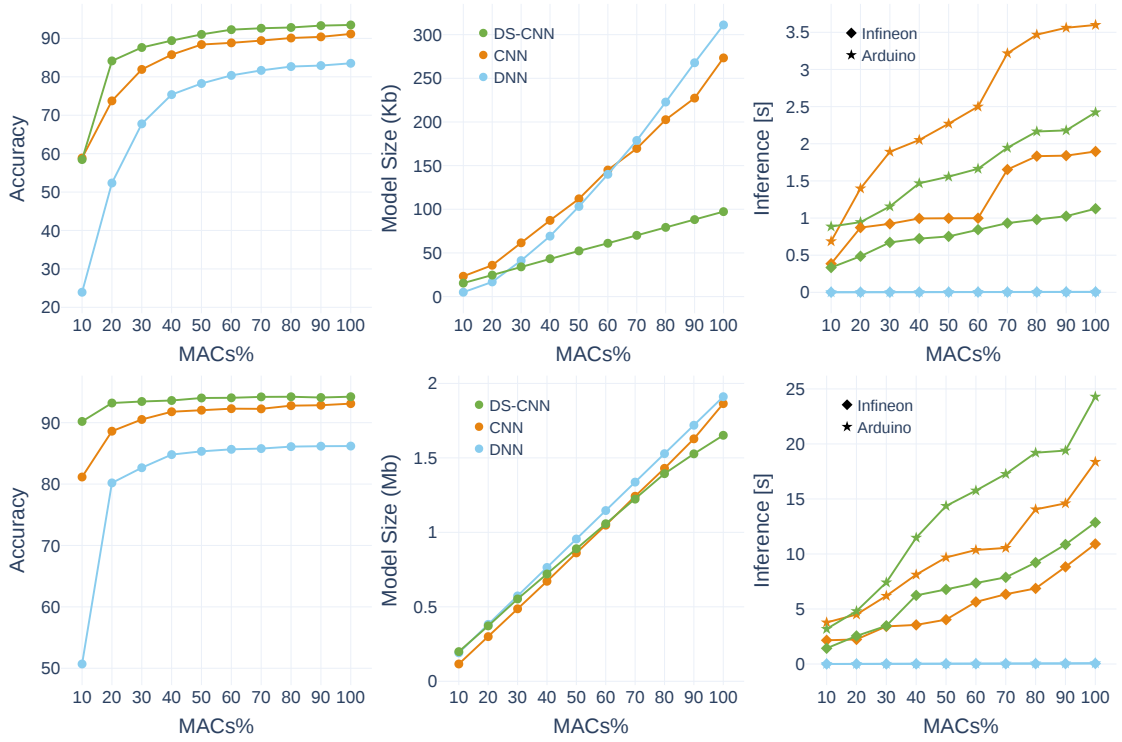

Figure 9: REDS size S (top row) and L (bottom row) architectures analysis finetuned on Google Speech Commands [60] with ten subnetworks. The plots from left to right show the subnetworks size, the subnetworks accuracy and the subnetworks inference time as a function of MAC percentage.

10.3 REDS on FMNIST and CIFAR10

The results in Table 7 and Table 8 show REDS performance using DS-CNN architecture of size S on FMNIST and CIFAR10. BU heuristic was used to obtain the results. REDS supports a different data domain without degrading the accuracy of the pre-trained model, reported in the header row. Compared to the state-of-the-art such as μNAS [44], REDS demonstrates a faster architecture search time for both FMNIST and CIFAR10. In the former, REDS takes 19 minutes as opposed to 3 days; in the latter, REDS takes 90 minutes as opposed to 39 days while requiring less memory for model storage for both datasets. After finding and freezing the 25% MACs subnetwork architecture, the BU heuristic takes only a few seconds to find the other 50% and 75% MACs subnetworks architectures.

MACs	Acc (%) - Pre-trained 90.28	Model Size (Kb)	Time Taken (m)
100%	91.06 ±0.18	128.54	–
75%	90.56 ±0.1	107.73	1.58 [s]
50%	90 ±0.1	87.4	5.83 [s]
25%	87.86 ±0.39	66.63	19 [m]

Table 7: Analysis of the BU knapsack subnetworks obtained from a depth-wise separable convolutional (DS-CNN) S network, pre-trained on FMNIST [63]. REDS supports a different data domain without degrading the accuracy of the pre-trained model, reported in the header row.

MACs	Acc (%) - Pre-trained 78.28	Model Size (Kb)	Time Taken
100%	80.72 ±0.42	128.54	–
75%	78.8 ±0.75	109.41	2.89 [s]
50%	75.4 ±3.09	88.01	10.59 [s]
25%	62.34 ±1.22	69.63	90 [m]

Table 8: The same evaluation as in Table 7 for CIFAR10 [38].

10.4 REDS energy efficiency

Table 9 reports the energy consumption of inference time by different network architectures measured by Power Profiler Kit (PPK2) on Nordic nRF52840 (Arduino Nano 33 BLE Sense). In comparison, the model adaptation time takes less than 0.01 mJ.

MACs	DNN [mJ]	CNN [mJ]	DS-CNN [mJ]
100%	0.18	90.61	61
75%	0.15	86.01	44.03
50%	0.07	50.3	33.81
25%	0.05	44.81	20.44

Table 9: Energy consumption of REDS size S architectures measured by the Power Profiler Kit (PPK2) on Nordic nRF52840. The results are obtained by performing an inference pass for each subnetwork model and recording the inference current.