A EXPERIMENT DETAILS

A.1 DATASETS

We conduct image classification experiments on the ImageNet dataset (Deng et al., 2009). To evaluate the generalization ability of pruned models, we also evaluate models on ImageNet-C (Hendrycks & Dietterich, 2019) and ImageNet-V2 (Recht et al., 2019) datasets. The former consists of the validation images from the ImageNet dataset, but with nineteen types of corruptions applied with five different levels of severity. The latter is a dataset created in the same manner as the original dataset, but consisting of different images. Therefore, it is intended to measure out of distribution performance of models. This ImageNet-V2 dataset includes three different sets, each one with a slightly different sampling strategy. We focus our experiments on the `MatchedFrequencies` dataset, but include evaluation all three datasets in Table 7 and see that our method consistently outperforms the baseline. Results do not use ASAM.

We conduct object detection experiments on the Pascal VOC dataset (Everingham et al., 2009). We additionally create a Pascal VOC-C dataset, in which we apply the ImageNet-C corruptions to the Pascal VOC test set. We only use one severity level.

A.2 ADDITIONAL ROBUSTNESS DATASETS

As described above, we choose the ImageNet V2 Matched Frequencies dataset to perform our main set of experiments, but there are two other ImageNet V2 datasets: Threshold 0.7 and Top Images. These three datasets vary in terms of selection criteria for included images. Matched Frequencies attempts to match the image selection frequency of MTurk workers in the original ImageNet validation set. Threshold 0.7 samples from among images with a selection frequency of at least 0.7. Top Images includes the images with the highest selection frequency within each class. We provide the pruned model robustness comparison on these the additional ImageNet V2 datasets in Table 7.

A.3 NETWORK ARCHITECTURES

In our experiments we prune three different networks for the classification task: ResNet50, MobileNet V1 and MobileNet V2 (He et al., 2016; Howard et al., 2017). We use a pretrained model trained for 90 epochs with a cosine learning rate as in HALP (Shen et al., 2022b) and EagleEye (Li et al., 2020). For object detection, we follow the experimental setup in HALP (Shen et al., 2022b) to prune an SSD512 model with ResNet50 as the backbone. We perform Distributed Data Parallel training across 8 V100 GPUs with batch size 128 for all experiments.

A.4 PRUNING SCHEDULE.

Given a pre-trained model, for any architecture, we run the warm up for 10 epochs, and then we follow the same pruning schedule as in de Jorge et al. (2020): we prune every 30 iterations and, in each iteration, we prune away a p_r fraction of neurons so that the final network is pruned by a fraction p (resulting in a network of size $1-p$). To determine the p_r fraction, we follow an exponential decay schedule. Let $k = 1-p$ and let k_r be the number of neurons remaining after r pruning iterations, where the total number of pruning iterations is R. Let m be the number of neurons in the dense network, and define $\alpha = \frac{r}{R}$. Then, $k_r = \exp\{\alpha \log k + (1-\alpha \log m)\}$. We fine-tune the pruned model for another 79 epochs (to reach 90 epochs total).

A.5 OPTIMIZATION HYPERPARAMETERS

The base optimizer is SGD with cosine annealing learning rate with a linear warmup over 8 epochs, a largest learning rate of 1.024, momentum of 0.875, and weight decay $3.05e-05$. Unless otherwise stated we use $\rho_{min} = 0.01$ and $\rho_{max} = 2.0$ for all experiments – we observe these values scale well across networks and tasks. For robustness encouragement we use $\rho = 2.0$ in line with prior work (Kwon et al., 2021). For some ablation experiments, the original SAM (Foret et al., 2020) optimizer is sufficient and in these cases we reduce $\rho_{max} = 0.1$ and use constant $\rho = 0.05$ for finetuning.

Table 7: Our AdaSAP method outperforms the Taylor pruning baseline on the additional ImageNet V2 datasets. Additionally, the AdaSAP model pruned to 76% outperforms the dense model across all three datasets. Results without ASAM.

Method	Matched Frequences	Threshold 0.7	Top Images
Dense	64.80	73.79	79.00
Taylor	60.56	69.74	75.53
AdaSAP$_P$	**62.03**	**70.83**	**76.62**
Taylor	63.51	72.54	77.83
AdaSAP$_P$	**64.62**	**73.75**	**78.87**
Taylor	64.53	73.32	78.72
AdaSAP$_P$	**66.00**	**74.70**	**79.66**

Table 8: **MobileNet-V1/V2 - Parameters.** Top1 Accuracy as a function of the pruning ratio.

Method	Size ↓	Val	R_{V2}	R_{C}	IN-V2	IN-C
			MobileNet-V1			
Dense	1	72.63	0.82	0.45	59.30	32.79
Taylor	0.40	69.61	0.80	0.42	55.90	29.51
AdaSAP$_P$	**0.39**	**71.05**	**0.82**	**0.44**	**58.08**	**31.06**
Taylor	0.52	71.21	0.81	0.43	57.65	30.92
AdaSAP$_P$	**0.52**	**71.58**	**0.82**	**0.44**	**58.75**	**31.77**
EagleEye	0.56	70.86	0.80	0.42	56.88	29.98
AdaSAP$_P$	**0.56**	**72.05**	**0.82**	**0.45**	**58.94**	**32.24**
			MobileNet-V2			
Dense	1	72.10	0.81	0.45	58.50	32.40
Taylor	0.72	70.49	0.81	0.43	56.87	30.22
AdaSAP$_P$	**0.72**	**72.06**	**0.82**	**0.45**	**58.73**	**32.63**
PolarReg	0.87	71.72	0.82	0.45	58.46	32.04
Taylor	0.88	71.93	0.82	0.45	58.75	32.21
AdaSAP$_P$	0.88	**72.34**	**0.82**	**0.45**	**59.28**	**32.80**

B Additional Results

B.1 MobileNet Results

We include results on MobileNet V1 and MobileNet V2 in Tables 8 and 9. These parallel our main results in the main paper, in which we evaluated AdaSAP$_P$ and AdaSAP$_L$ on ResNet50 and compared them against other pruning methods. Here, we perform a similar comparison, where Table 8 includes our results on AdaSAP$_P$ and Table 9 includes our results on AdaSAP$_L$. We compare against Taylor importance (Molchanov et al., 2019), EagleEye (Li et al., 2020), and PolarReg (Zhuang et al., 2020) for AdaSAP$_P$ and against HALP (Shen et al., 2022b), SMCP (Humble et al., 2022), MetaPruning (Liu et al., 2019), AutoSlim (Yu & Huang, 2019), AMC (He et al., 2018), and EagleEye (Li et al., 2020) for AdaSAP$_L$. Results here do not use ASAM. We can observe that AdaSAP performs strongly compared to baselines, particularly in the parameter-based setting.

B.2 Margin of Improvement in Classification

Figure 4 shows the margin of performance on various corruption types for the classification task. AdaSAP outperforms a Taylor pruned model on all corruptions, across three different sparsities.

Table 9: **MobileNet-V1/V2 - Latency.** Top1 accuracy for latency constrained pruning for various speedup ratios. "–" indicates that we could not evaluate the model due to unavailable code or models.

Method	Speedup ↑	Val	R_{V2}	R_C	IN-V2	IN-C
		MobileNet-V1				
Dense	1	72.63	0.82	0.45	59.30	32.79
MetaPruning	2.06	66.1	–	–	–	–
AutoSlim	2.27	67.9	–	–	–	–
HALP	2.32	68.30	0.80	0.41	54.95	28.15
SMCP	**2.39**	68.34	0.80	**0.42**	54.38	**28.68**
AdaSAP$_L$	2.33	**68.45**	**0.81**	0.41	**55.42**	28.29
0.75 MobileNetV1	1.37	68.4	–	–	–	–
AMC	1.42	70.5	–	–	–	–
MetaPruning	1.42	70.9	–	–	–	–
EagleEye	1.47	70.86	0.80	0.42	56.88	29.98
HALP	1.68	71.31	0.81	0.43	57.38	30.77
SMCP	**1.72**	71.00	0.81	0.44	57.20	31.02
AdaSAP$_L$	1.70	**71.48**	**0.82**	**0.44**	**58.23**	**31.35**
		MobileNet-V2				
Dense	1	72.10	0.81	0.45	58.50	32.40
HALP	**1.84**	70.42	0.81	0.45	57.21	31.69
AdaSAP$_L$	1.81	**71.35**	0.81	**0.46**	**57.85**	**32.63**
HALP	1.33	72.16	0.81	0.46	58.53	**33.04**
AdaSAP$_L$	**1.39**	**72.19**	**0.82**	0.46	**59.36**	32.91

We can see that it tends to particularly improve performance on several corruptions that may be important for the autonomous driving application, such as pixelated images, fog, and snow.

B.3 Margin of Improvement in Object Detection

Similarly to our result on classification, we include margins of performance improvement on various corruption types for the object detection task in Figure 5.

B.4 Relative Robustness

In Figure 6 we show that AdaSAP outperforms baselines on each of the constituent elements of the relative robustness metric. Recall that relative robustness is robust accuracy divided by standard validation accuracy. In addition to outperforming baselines on relative robustness, AdaSAP also outperforms on ImageNet validation accuracy and ImageNet C robust accuracy.

C Additional Ablations

C.1 Performance change after pruning

In Table 12 we show how the loss and accuracy change over the course of pruning. Our hypothesis is that our method sets up the network for better pruning, so that the performance drop over the course of pruning is minimized. In most cases, our method has the best validation loss and accuracy both before and after pruning. This indicates that our method sets up the model to be pruned well, and also preserves performance well throughout the pruning process.

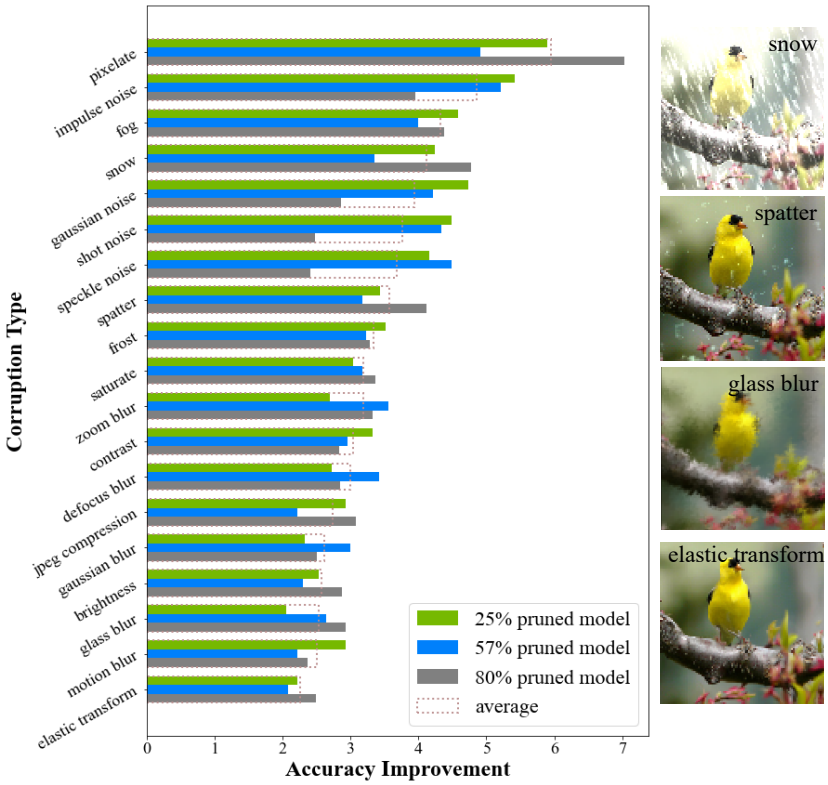

Figure 4: Performance difference on various ImageNet C corruption types on models of varying sparsity. Accuracy improvement is the Top1 accuracy on a model trained with AdaSAP minus that of a Taylor pruned model.

Table 10: **Sensitivity to pruning criteria.** AdaSAP performs best when combined with magnitude pruning, but is flexible enough to be used with other criteria, such as Taylor importance. Here we see that AdaSAP with Taylor pruning matches or outperforms SGD with Taylor pruning. Results do not use ASAM.

Method	Size ↓	Val	R_{V2}	R_{C}	IN-V2	IN-C
Taylor + SGD	0.42	75.85	0.84	0.50	63.51	37.84
AdaSAP$_{P,\text{Taylor}}$	0.43	76.26	0.84	0.50	63.77	38.07
AdaSAP$_P$	**0.41**	**76.93**	**0.84**	**0.52**	**64.49**	**39.64**
Taylor + SGD	0.76	77.05	0.84	0.52	64.53	39.68
AdaSAP$_{P,\text{Taylor}}$	0.76	77.42	0.84	0.52	65.24	40.27
AdaSAP$_P$	**0.76**	**77.86**	**0.85**	**0.53**	**66.00**	**41.30**

C.2 Varying the length of the adaptive weight perturbation step.

Throughout the main set of experiments we set to 10 the number of epochs used for the adaptive weight perturbation step. This delineates how long we use the AdaSAP optimizer for, as well as how soon into the procedure we begin pruning. In this experiment, we analyze the sensitivity of the approach to this parameter. We report results for this experiment in Table 13. We can observe that as we make this period longer, validation accuracy and ImageNet C accuracy both drop slightly, while ImageNet V2 seems to have no discernible pattern. Ratios R_C and R_{V2} also stay consistent across the experiment.

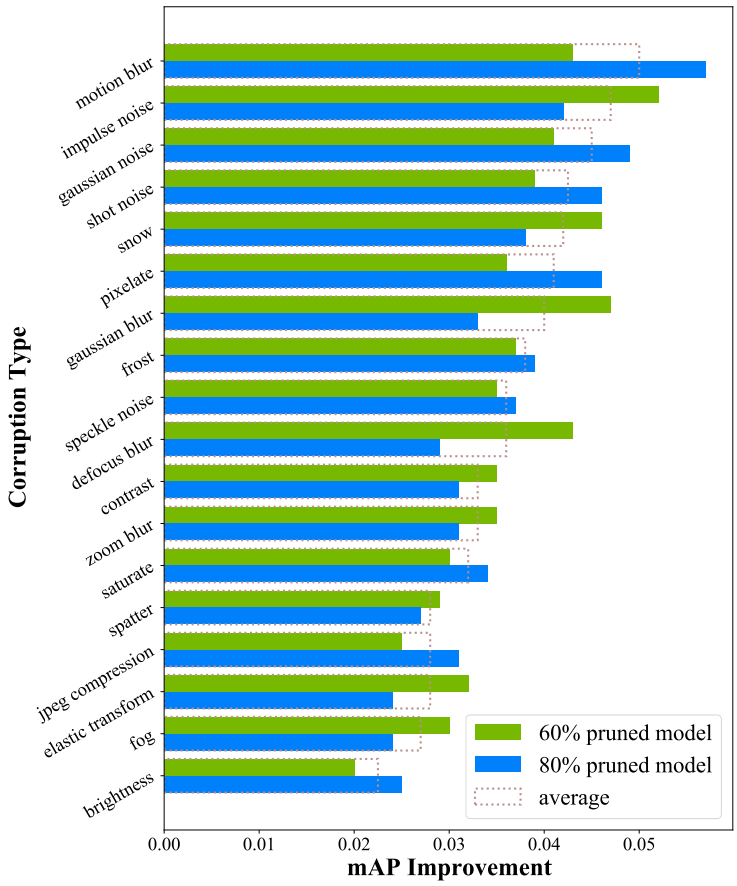

Figure 5: **Pascal VOC-C dataset.** Performance improvement per corruption as the difference between a model trained using AdaSAP minus that of a HALP pruned model.

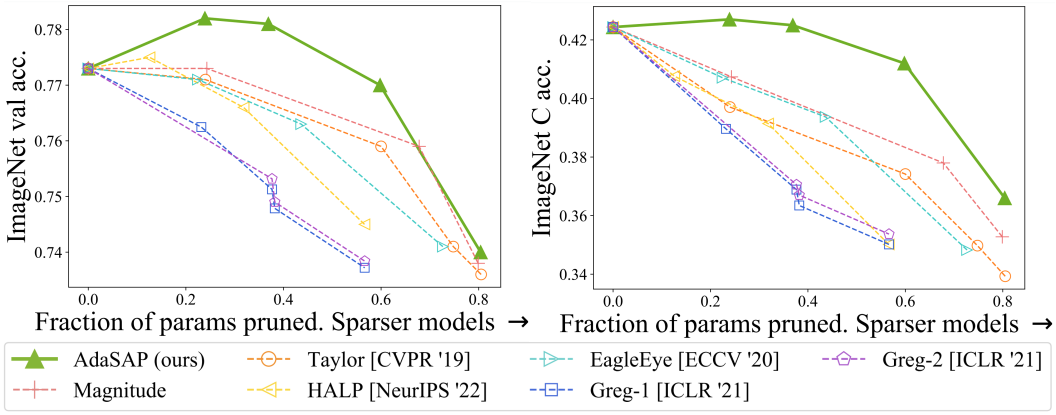

Figure 6: ImageNet Validation and ImageNet C performance on AdaSAP vs. baselines. Contains the same data as used to produce Figure 1 but demonstrates that AdaSAP additionally dominates baselines on both components of the relative robustness metric.

Table 11: **Confidence intervals over three repeats.** EagleEye checkpoints are obtained from the official repository with only one seed.

Method	Size	Val	IN-V2	IN-C
Magnitude	0.20	73.71 ± 0.12	61.21 ± 0.17	35.33 ± 0.18
Taylor	0.20	73.42 ± 0.19	60.36 ± 0.19	33.97 ± 0.12
EagleEye	0.27	74.13	61.30	34.84
AdaSAP$_P$	0.20	$\mathbf{74.54 \pm 0.09}$	$\mathbf{62.21 \pm 0.13}$	$\mathbf{37.30 \pm 0.19}$
Magnitude	0.76	77.32 ± 0.06	65.18 ± 0.27	40.64 ± 0.20
Taylor	0.76	77.05	64.53	39.68
EagleEye	0.78	77.07	64.84	40.67
AdaSAP$_P$	0.77	$\mathbf{78.23 \pm 0.06}$	$\mathbf{65.98 \pm 0.32}$	$\mathbf{43.43 \pm 0.20}$

Table 12: Validation Loss and Accuracy directly before and after pruning (before additional fine-tuning). Across several pruning levels, our method generally reaches the lowest validation loss and accuracy both before and after pruning. Results do not use ASAM.

Method	Val Loss Before	Val Loss After	Val Acc Before	Val Acc After
Taylor	**2.245**	3.315	**67.798**	42.915
Mag	2.416	3.288	63.843	43.21
SAM	2.255	3.2	67.577	45.802
AdaSAP$_P$	2.293	**3.146**	66.555	**46.313**
Taylor	2.284	2.69	66.67	57.046
Mag	2.408	2.587	63.844	59.275
SAM	2.327	2.676	65.541	57.449
AdaSAP$_P$	**2.251**	**2.574**	**67.473**	**59.962**
Taylor	2.441	2.42	63.389	63.711
Mag	2.292	2.406	66.714	64.087
SAM	2.401	2.379	64.019	64.589
AdaSAP$_P$	**2.213**	**2.347**	**68.463**	**65.393**

Table 13: **Effects of varying number of epochs of adaptive weight perturbation.** Increasing the number of epochs leads to smaller models but slightly worse validation and ImageNet C performance.

Num epochs	Size ↓	Val	R_{V2}	R_{C}	IN-V2	IN-C
5	0.48	73.89	0.83	0.48	61.29	35.65
10	0.46	73.82	0.83	0.48	60.91	35.59
20	0.44	73.63	0.84	0.48	61.48	35.25
30	0.43	73.47	0.83	0.48	60.72	35.04

Table 14: Comparison of various weight perturbation strategies during robustness encouragement.

Perturbation Type	Val	IN-C	IN-V2
No weight perturbations	73.98	35.84	62.00
Adaptive weight perturbations	74.10	35.72	61.94
Uniform weight perturbations	**74.39**	**35.86**	**62.03**

Table 15: **Comparison of AdaSAP to SAM optimizer.** AdaSAP outperforms SAM and SGD on standard validation performance and ImageNet-C performance, but slightly trails SAM on ImageNet-V2 when both methods are augmented with ASAM.

Method	Size ↓	Val	R_{V2}	R_{C}	IN-V2	IN-C
		SGD				
Dense	1	77.32	0.84	0.54	64.79	42.46
Taylor + SGD	0.20	73.56	0.82	0.46	60.56	33.93
		AdaSAP vs. SAM (without ASAM)				
Taylor + SAM	0.20	73.62	0.83	0.47	61.37	34.49
AdaSAP$_P$	0.20	**74.38**	**0.83**	**0.48**	**62.03**	**35.86**
Taylor + SGD	0.42	75.85	0.84	0.50	63.51	37.84
Taylor + SAM	0.43	76.27	0.84	0.50	63.81	37.72
AdaSAP$_P$	**0.40**	**77.03**	**0.84**	**0.51**	**64.62**	**39.57**
Taylor + SGD	0.76	77.05	0.84	0.52	64.53	39.68
Taylor + SAM	0.76	77.29	0.84	0.52	64.84	40.07
AdaSAP$_P$	0.76	**77.86**	**0.85**	**0.53**	**66.00**	**41.30**
		AdaSAP vs. SAM (with ASAM)				
SAM + ASAM	0.19	73.93	0.84	0.50	61.76	36.66
AdaSAP$_P$ **+ ASAM**	0.19	**74.63**	0.83	0.50	**62.08**	**37.30**
SAM + ASAM	0.41	77.10	0.84	0.53	64.94	40.90
AdaSAP$_P$ **+ ASAM**	**0.40**	**77.27**	0.83	0.53	64.51	**41.23**

C.3 Robustness Encouragement

As mentioned in the main text, we consider various ablations to determine the necessity of various steps of the AdaSAP procedure. In the third step of our procedure, robustness encouragement, we choose to apply uniform perturbations across all weights in the network. This differs from the first step, in which we apply adaptive weight perturbations. In Table 14, we examine the effects of different weight perturbation strategies during the robustness encouragement phase. We can see that while all three strategies lead to relatively close final performance across the three datasets, uniform weight perturbations perform slightly better, suggesting that our choice of applying them in our procedure may be slightly benefitting the performance.

C.4 Importance of Adaptive Weight Perturbations

In Table 15 we extend an ablation from the main paper in which we compare AdaSAP to SAM, effectively evaluating the importance of warmup with adaptive weight perturbations. Here, we perform the comparison on a wider range of sparsities and observe that a similar pattern emerges.