6 Appendix

6.1 Algorithm

The pesudocode of AST is presented in Algorithm 1. To generate and sparsely train each sub-net, AST adapts the gradual pruning scheme combining with *prune-and-regrow* from GraNet [14]. Specifically, given initial sparsity s_i, target sparsity s_f, gradual pruning frequency ΔT, starting and end epoch of gradual pruning t_0 and t_f, pruning iterations n, the pruning rate of each pruning iteration is defined as:

$$s_t = s_f + (s_i - s_f)(1 - \frac{t - t_0}{n\Delta t}), t \in t_0, t0 + \Delta t, ..., t0 + n\Delta t \tag{6}$$

Algorithm 1 The pseudocode of AST.

Require: Model weight W, number of sub-nets N, initial sparsity s_i, target sparsity $[s_f^1, ..., s_f^N]$, gradual pruning frequency ΔT, extended adjustment period $\Delta\tau$.

```
random initialize $W$ with initial sparsity $s_i$
for each training iteration t do
    get current number of training iterations $t_c$
    switching to current sub-net $W_i$ where $i \leftarrow t_c$ mod $(N * \Delta\tau)$
    training $W_i \leftarrow SGD(W_i)$
    if ($t$ mod $\Delta T$ == 0) then
        gradual pruning with the pruning rate produced by Eq. 6 with target sparsity $s_f^i$
        prune-and-regrow by Eq.1 and Eq.2
        if $i\ != 0$ then
            gradient correction within the inner-group by Eq.4
        end if
    end if
end for
```

6.2 Detailed explanation of rational of AST

Following the proof of Reptile [18], the expectation of $\alpha\overline{H}_1 g_2$ can be further expressed as the gradient inner product of two consecutive sub-nets:

$$\begin{aligned}\mathbb{E}_{1,2}[\alpha\overline{H}_1 g_2] &= \mathbb{E}_{1,2}[\alpha\overline{H}_2 g_1] \\ &= \frac{1}{2}\mathbb{E}_{1,2}[\alpha\overline{H}_1 g_2 + \alpha\overline{H}_2 g_1] \\ &= \frac{1}{2}\mathbb{E}_{1,2}[\frac{\partial}{\partial w_1}(g_1 \cdot g_2)]\end{aligned}$$

As shown in Eq.6.2, it is clear to see that the term $-\alpha\overline{H}_1 g_2$ is the direction hat serves to maximize the inner product of two consecutive mini-batches. Thus, it proves that the proposed AST has implicit regularization to alignment the weight update between sub-nets.

6.3 Detailed experimental setup of AST

The training hyper-parameters of the compared individual sparse training works are same for CIFAR-10 and CIFAR-100. But this line of works adapt different hyper-parameters to achieve good accuracy on ImageNet. The report accuracy of Rigl [4] uses 4096 batchsize and trains the model on 100 epochs with initial learning rate 1.6. GraNet [14] uses 64 bathsize, 100 training epochs, and set the initial learning rate to 0.1. Mest [26] uses a larger 2048 batchsize and trains the model on 150 epochs with the initial learning rate 2.048. In this case, to evaluate our method and fair compare with them, we conduct the basic training settings like the original ResNet by using 256 batchsize and training the model on 100 epochs with 0.1 initial learning rate. We believe a more fine-grained hyper-parameter setting could lead to better accuracy. We only run AST on ImageNet once due to the limited resources by using four Nvidia RTX A4000 GPUs.

Table 7: ImageNet accuracy and training cost comparison with SoTA works on ResNet-50.

Method	**ImageNet-2012**			
ResNet-50	Dense model Acc. = 76.8			
Prune Ratio	80%		90%	
	Individual Training			
	Top-1 Acc. (%)	**Training Cost**	**Top-1 Acc. (%)**	**Training Cost**
SNIP [11]	69.7	1×	62.0	1×
SET [15]	72.9	1×	69.6	1×
DSR [17]	73.3	1×	71.6	1×
RigL [4]	74.6	1×	72.0	1×
GraNet [14]	76.0	1×	74.5	1×
MEST + EM [26]	75.8	1×	73.6	1×
	Training once for all			
Jointly-Trained [24]($s_i = 0\%$)	$71.9_{0.5\times}$	1×	$65.0_{0.25\times}$	1×
AST + GC ($\mathbf{s_i = 50\%}$)	**73.2**	**0.5**×	**73.1**	**0.5**×
AST + GC ($\mathbf{s_i = 80\%}$)	**72.6**	**0.5**×	**72.5**	**0.5**×

Table 8: ImageNet accuracy and training cost comparison with SoTA works on ResNet-18.

Method	**ImageNet-2012**			
ResNet-18	Dense model Acc. = 69.76			
Prune Ratio	80%		90%	
	Individual Training			
	Top-1 Acc. (%)	**Training Cost**	**Top-1 Acc. (%)**	**Training Cost**
GraNet [14]	-	1×	63.1	1×
	Training once for all			
AST + GC ($\mathbf{s_i = 80\%}$)	**62.3**	**0.5**×	**62.1**	**0.5**×

Table 9: CIFAR-100 accuracy and training cost comparison with SoTA works on wide ResNet-32.

Dataset	**CIFAR-100 Acc. (%)**			**Training FLOPS**
ResNet-32	**Dense Model Acc. = 74.94%**			1.37e+16 (1×)
	Individual Training			
Lottery Ticket [5]	68.99	65.02	57.37	-
SNIP [11]	68.89	65.22	54.81	-
DSR [17]	69.63	68.20	61.24	-
GraNet [14]	73.18	72.56	69.89	1.51e+16 (1.13×)
MEST [26]	69.35±0.36	67.85±0.23	62.58±0.31	1.47e+16 (1.07×)
MEST+EM [26]	70.44±0.26	68.43±0.32	64.59±0.27	-
MEST+EMS [26]	71.30±0.31	70.36±0.05	67.16±0.25	-
	Training once for all			
Jointly-Trained ($s_i = 50\%$) [24]	70.40±0.14	69.32±0.84	66.85±0.59	1.45e+16 (1.09×)
AST ($s_i = 0\%$)	73.12±0.10	72.39±0.14	68.06±0.21	**6.47e+15** (**0.48**×)
AST ($s_i = 90\%$)	69.82±0.12	69.22±0.07	69.37±0.15	**5.03e+15** (**0.38**×)
AST+GC ($s_i = 0\%$)	**73.41**±**0.04**	**72.57**±**0.15**	**68.42**±**0.15**	**6.47e+15** (**0.48**×)[1]
AST+GC ($s_i = 90\%$)	**70.11**±**0.39**	**70.01**±**0.54**	**67.15**±**0.31**	**5.03e+15** (**0.38**×)[1]

[1] For the wide ResNet-32 model, the step-wise and layer-wise gradient projection requires 1.86e+12 FLOPS, which is minimum compared to the majority of training.

6.4 Additional experimental results

To evaluate the effect of initial sparsity, we conduct the experiment by using 50% initial sparsity of ResNet-50 on ImageNet as shown in Table.8. Compared to 80% initial sparsity, 50% initial sparsity could achieve 0.6% accuracy gain on both 80% and 90% prune ratio. Furthermore, we also conduct the experiment on the smaller ResNet-18, which achieve 62.3% and 62.1% accuracy on 80% and 90% prune ratio respectively.

6.5 Training cost comparison

In addition to the training rounds comparison, we analyze the training cost of the proposed AST method in terms of the detailed metrics. Table 9 summarizes the number of total FLOPs of training multiple sparse deep neural networks. Compared to the joint-training scheme [24] or seperately trained GraNet [14], the proposed AST algorithm achieves up to 2.63× training cost reduction, while maintaining the similar inference accuracy as the individual training baseline. Furthermore, the detailed computation cost of the ImageNet experiments are reported in Table 3.

6.6 The impact of different extended adjustment (EA) periods

Regarding the **Observation 1**, we have summarized the comparison results between the investigated Completely-subset (CS) scheme and the Non-disjoint scheme (ND) in Table 1 of the original manuscript. As a result, on CIFAR-10 dataset, the ND scheme has the outperformed performance compared to the constrained completely-subset scheme (CS).

For the **Observation 2**, to validate the effectiveness of the proposed extended adjustment method (EA), we add an ablation study on various adjustment periods $\Delta\tau$, which is used to determine the frequency of sub-nets switching. As shown in Table R1, compared to the smaller adjustment period (i.e., $\Delta\tau = 0, 90$), $\Delta\tau = 300$ achieves the best accuracy on all three sparsity levels. The reason is that performing sub-net switching frequently elevates the instability of model optimization.

Table 10: The impact of the extended adjustment period. Given the wide ResNet-32 and CIFAR-100 dataset, sweep the sparsity update interval from 0 epoch up to 300 steps.

Dataset	**CIFAR-100 Acc (%)**		
ResNet-32	Dense Model Acc. = 74.94		
$\Delta\tau$ (steps)	90%	95%	98%
0	72.92±0.27	72.25±0.20	68.20±0.07
90	73.28±0.13	72.72±0.20	68.25±0.03
300	**73.41±0.04**	**72.57±0.15**	**68.42±0.15**

6.7 AST vs. Naive fine-tuning

We analyze the impact of fine-tuning based on the following three perspectives:

1. A short time of fine-tuning from the dense pre-trained model.
2. Start with sparse pre-training, fine-tune the high sparsity models with a short epochs, while keep the overall training cost (time) as same as AST.
3. To further clarify the advantages of AST, we also investigate another perspective: a short time of fine-tuning from the sparse pre-trained model.

Same as the experimental setup in the main paper, we conduct the experiments based on the wide ResNet-32 model on CIFAR-100 dataset. Given the dense pre-trained model, we separately fine-tune the dense model to achieve 90%, 95%, and 98% sparsity with minimum efforts. As shown in Table 11, fine-tuning from a dense model in a short period cannot achieve comparable accuracy as the proposed AST algorithm. Furthermore, the 160 epochs of pre-training and additional fine-tuning elevate the overall training costs.

In addition to the dense model fine-tuning, we address the second concern of the reviewer by performing the sparse progressive training while keeping the overall training cost to be the same as a single AST training. With the wide ResNet-32 model, we first sparsify the model to 90% sparsity from scratch with 60 epochs. Subsequently, we prune the 90% sparse model to 95% and then to 98% sparsity with 50 epochs of fine-tuning. Compared to the single AST training, the total training effort is the same (60+50+50=160 epochs). As shown in Table 12, such an individual pruning method failed to achieve the performance as the proposed AST training method. The large accuracy gap suggests the necessity of the proposed alternative sparsification training.

Table 11: Fine-tune to high sparsity models from a pre-trained **dense** checkpoint with minimum training effort (up to 30 epochs of fine-tuning).

Dataset	CIFAR-100 Acc. (%)		
ResNet-32	Dense Model Acc. = 74.94		
Sparsity	90%	95%	98%
160+10 Epochs	70.06±0.08	62.56±0.15	44.47±0.86
160+20 Epochs	72.13±0.23	67.66±0.04	56.50±0.16
160+30 Epochs	72.76±0.19	68.89±0.35	59.34±0.79
AST+GC (160 epochs)	**73.41±0.04**	**72.57±0.15**	**68.42±0.15**

Table 12: Progressive sparse fine-tuning on CIFAR-100 dataset with wide ResNet-32 model: Start from scratch, train a 90% sparse model with 60 epochs then fine-tuning to 95% and 98% sparsity with 50 epochs each. The total training effort is same as a single AST run (160 epochs).

Dataset	**CIFAR-100 Acc. (%)**		
ResNet-32	Dense Model Acc. = 74.94		
Sparsity	90%	95%	98%
Epoch	60	50	50
Progressive Fine-tune	71.68±0.06	71.11±0.04	68.02±0.14
AST+GC	**Training Epoch = 160**		
	73.41±0.04	**72.57±0.15**	**68.42±0.15**

Furthermore, we investigate the impact of fine-tuning based on a pre-trained **sparse** model. We first fully train a sparse subnet with 90% sparsity (with 160 epochs) and prune the resultant model to 95% and 98% with a minimum amount of fine-tuning. As shown in Table 13, fine-tuning the 90% sparse model to 95% or 98% sparsity with up to 30 epochs cannot achieve comparable accuracy as AST, with even higher total training effort.

The experimental results in Table 11, Table 12, and Table 13 suggest that it is difficult for individual fine-tuning to achieve the level of high sparsity and high accuracy as the proposed AST, regardless of the initial sparsity of the inherited model checkpoint.

Table 13: Fine-tune to high sparsity model from a pre-trained **sparse** checkpoint (90% sparsity) with minimum training effort (up to 30 epochs of fine-tuning).

Dataset	**CIFAR-100 Acc. (%)**	
ResNet-32	90% Sparse Model Acc. = 73.16	
Sparsity	95%	98%
160+10 Epochs	69.63±0.09	59.25±0.44
160+20 Epochs	70.95±0.20	64.49±0.23
160+30 Epochs	71.70±0.28	66.67±0.24
AST-GC (160 Epochs)	**72.57±0.15**	**68.42±0.15**

6.8 Inference acceleration and computation reduction of AST

Table 14: Inference acceleration and negligible accuracy drop of the proposed AST algorithm with structured fine-grained sparsity on ResNet-18 model.

Dataset	**CIFAR-10 Acc. (%)**					**Training Cost**
N:M Sparse Pattern	Dense Model	2:4	3:4	7:8	15:16	
Individually Trained (SR-STE)	95.07	94.89	94.47	94.25	93.92	2.33e+16 (3.95×)
AST + GC	-	94.63	94.26	94.31	93.79	**5.91e+15** (**1**×)
Inference FLOPS / 10K images	5.12e+12	2.56e+12	1.28e+12	6.40e+11	3.19e+11	-