Appendix

We explore a variety of deep neural networks (DNNs) to support our results. All experiments on CIFAR-10 and CIFAR-100 use the basic data augmentation including random cropping and random horizontal flipping. No other tricks are used. We use 150 training epochs ($T = 150$) and decay the initial learning rate after 80, 120 epochs by a factor of 10 (with step learning rate scheduler). The optimizer is stochastic gradient descent (SGD) and the momentum is 0.9. This training setup is used for pre-training ($f(x; \theta_T)$), pruning & fine-tuning ($(\theta_T \odot m)_T$), and sparse training ($(\theta_0 \odot m)_T$) as introduced in Section 3.

A. Revisit Lottery Tickets

We show the experiment results of MobileNet-V2 on CIFAR-10, and ResNet-20, VGG-11, and MobileNet-V2 on CIFAR-100 over a range of different sparsity ratios with the masks generated from iterative pruning (Frankle & Carbin, 2018) at learning rate 0.01 and 0.1, respectively. We conduct each experiment *five* times (result variation shown in figures). We set the same training epochs (i.e., 150 epochs) for training the original DNNs with initial weights $f(x; \theta_0)$ (i.e., pretraining), training from randomly reinitialized weights with the mask $f(x; \theta_0' \odot m)$ (random reinitialization), and training from the initial weights with the mask $f(x; \theta_0 \odot m)$ ("winning ticket").

CIFAR-10 Results: Figure 1(a) and 1(b) illustrate the result on MobileNet-V2 using CIFAR-10. The pre-trained MobileNet-V2's accuracy on CIFAR-10 is 92.20% at initial learning rate 0.01, and 93.86% at initial learning rate 0.1.

CIFAR-100 Results: Figure 1(c) and 1(d) show the result on MobileNet-V2 using CIFAR-100. The pre-trained MobileNet-V2's accuracy on CIFAR-100 is 73.10% at initial learning rate 0.01, and 74.76% at initial learning rate 0.1. Figure 1(e) and 1(f) show the result on ResNet-20 for CIFAR-100. The pre-trained ResNet-20's accuracy on CIFAR-100 is 63.10% at initial learning rate 0.01, and 67.15% at initial learning rate 0.1 (see the significant gap here). Figure 1(g) and 1(h) show the result on VGG-11 for CIFAR-100. The pre-trained VGG-11's accuracy on CIFAR-100 is 67.74% at initial learning rate 0.01, and 69.83% at initial learning rate 0.1. In the case of MobileNet-V2 on CIFAR-100 at low learning rate, we observe that the "winning ticket" can outperform the random reinitialization but failed to restore the baseline accuracy (73.10%). This indicates the low learning rate is not desirable. For all illustrated cases, the "winning ticket"'s accuracy is close to the random reinitialization at the initial learning rate 0.1. While in the case of learning rate 0.01, the "winning ticket" can outperform the random reinitialization over different sparsity ratios. Note there is a clearly accuracy gap between the pretrained DNNs with the initial learning rate 0.1 and with the initial learning rate 0.01.

From these experiments, the winning property exists at a low learning rate but does not exist at a relatively high learning rate. However, we would like to point out that the relatively high learning rate of 0.1 (which is, in fact, the standard learning rate on these datasets) results in *notably higher accuracy* in the pretrained DNNs than the low learning rate (MobiletNet-V2 on CIFAR-10 93.86% vs. 92.20%, MobiletNet-V2 on CIFAR-100 74.76% vs. 73.10%, VGG-11 on CIFAR-100 69.83% vs. 67.74%, ResNet-20 on CIFAR-100 67.15% vs. 63.10%). We should not draw conclusion basd on the low (insufficient) learning rate in general.

B. Weight Correlation in DNN Pre-Training

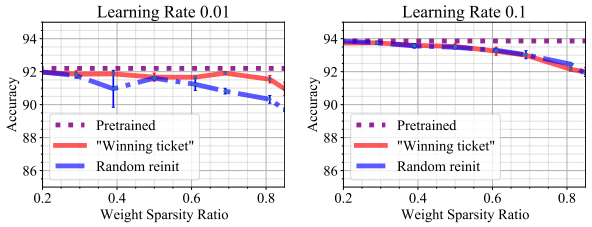
We investigate the *correlation indicator* between the initial weights θ_0 and the trained weights θ_T from DNN pretraining on VGG-11, ResNet-20, and MobileNet-V2 on CIFAR-10 and CIFAR-100 under learning rates of 0.01 and 0.1, respectively.

We have performed experiments to derive $R_p(\theta_0, \theta_T)$ on different DNN pretraining with different initial learning rates. Figure 2 illustrates the correlation indicator between the initial weights θ_0 and the trained weights θ_T from DNN pretraining at learning rates of 0.01 and 0.1 on VGG-11 and MobileNet-V2 using CIFAR-10/100, respectively. We use the same hyperparameters mentioned in the setup without additional training tricks. Figure 2(a) and 2(b) illustrate the result on VGG-11 for CIFAR-10/100. Figure 2(c) and 2(d) illustrate the result on MobileNet-V2 for CIFAR-10/100.

We can observe that $R_p(\theta_0, \theta_T)$ at a learning rate of 0.01 has a notably higher correlation compared to the case of learning rate 0.1. This observation indicates that the large-magnitude weights of θ_0 are not fully updated at a low learning rate of 0.01, indicating that the pre-trained DNN is not well-trained. In the case of learning rate 0.1, the weights are sufficiently updated thus largely independent from the initial weights ($R_p(\theta_0, \theta_T) \approx p$, where $p = 10\%, 20\%, 30\%, 40\%, 50\%$), indicating a well-trained DNN.

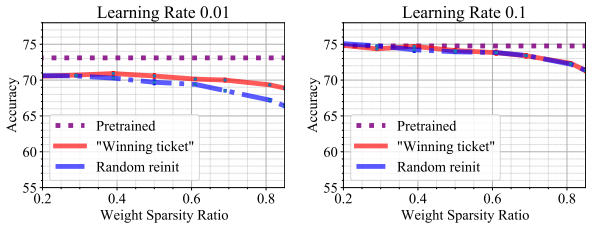
C. Pruning & Fine-tuning

Consider the "pruning & fine-tuning" case formally defined in Section 3, in which we apply mask m on the trained weights θ_T from DNN pretraining, and then perform fine-tuning for another T epochs. The final weights are denoted by $(\theta_T \odot m)_T$. We study accuracy of the "pruning & fine-tuning" result $f\big(x; (\theta_T \odot m)_T\big)$ at different sparsity ratios, with learning rates of 0.01 and 0.1 on different DNNs using CIFAR-10 and CIFAR-100. We use the same hyperparame-



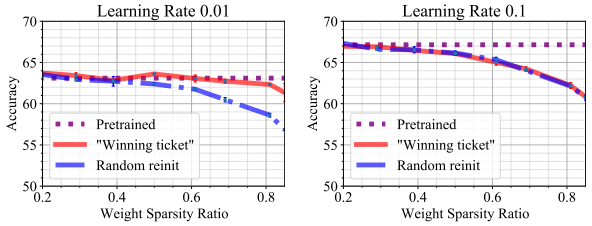
(a) Iterative pruning at learning rate of 0.01 on MobileNet-V2 using CIFAR-10.

(b) Iterative pruning at learning rate of 0.1 on Mobilenet-V2 using CIFAR-10.



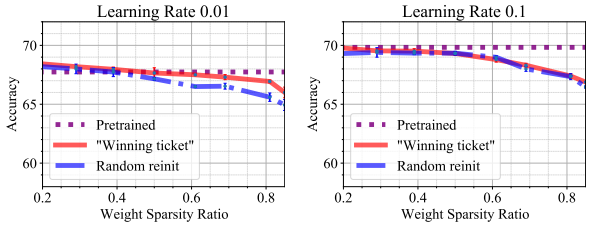
(c) Iterative pruning at learning rate of 0.01 on MobileNet-V2 using CIFAR-100.

(d) Iterative pruning at learning rate of 0.1 on MobileNet-V2 using CIFAR-100.



(e) Iterative pruning at learning rate of 0.01 on ResNet-20 using CIFAR-100.

(f) Iterative pruning at learning rate of 0.1 on ResNet-20 using CIFAR-100.



(g) Iterative pruning at learning rate of 0.01 on VGG-11 using CIFAR-100.

(h) Iterative pruning at learning rate of 0.1 on VGG-11 using CIFAR-100.

Figure 1. Accuracy illustration of random reinitialization and "winning tickets" for MobileNet-V2 on CIFAR-10, and MobileNet-V2, ResNet-20 and VGG-11 on CIFAR-100 at learning rates 0.01 and 0.1.

ters as mentioned in the setup ($T = 150$). The accuracies of the pretrained DNNs with corresponding learning rates are also provided. Figure 3(a) and 3(b) illustrate the "pruning & fine-tuning" result on MobileNet-V2 for CIFAR-10 using learning rates of 0.01 and 0.1, respectively. Figure 3(c) and 3(d) illustrate the "pruning & fine-tuning" result on MobileNet-V2 for CIFAR-100 with learning rates of 0.01 and 0.1, respectively. In the case of MobileNet-V2 for CIFAR-100 with the initial learning rate 0.1, the "pruning & fine-tuning" scheme consistently perform better than the pretrained dense DNN (74.76%).

We can observe that $f\big(x; (\theta_T \odot m)_T\big)$ achieves relatively high accuracy, close to or higher than the accuracy of the pretrained DNN at the same learning rate (even at the desirable learning rate 0.1).

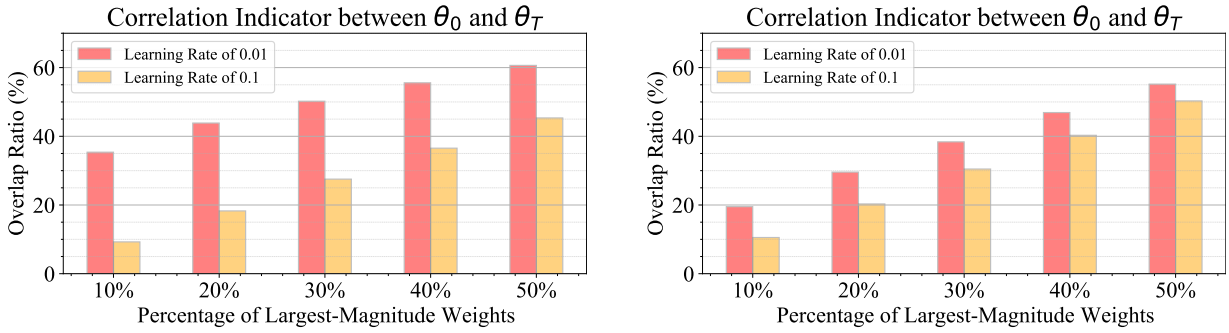
D. Sparse Correlation

We study the correlation between $\theta_0 \odot m$ ($\theta_0' \odot m$) and $(\theta_T \odot m)_T$ to shed some light on the cause of winning property. We illustrate the correlation on ResNet-20, VGG-11 and MobileNet-V2 for CIFAR-10/100 at learning rate 0.01, 0.1, respectively. We show the correlation indicator between $\theta_0 \odot m$ ("winning ticket") and $(\theta_T \odot m)_T$, and between $\theta_0' \odot m$ (random reinitialization) and $(\theta_T \odot m)_T$ at learning rate 0.01, 0.1. Figure 4 illustrates the result of ResNet-20 for CIFAR-100 at the learning rate 0.01 and 0.1. Figure 5 shows the result of VGG-11 for CIFAR-10/100 and Figure 6 shows the result of MobileNet-V2 for CIFAR-10/100 at learning rates 0.01, 0.1, respectively. In the case of high learning rate 0.1, the weight correlation between $\theta_0 \odot m$ ("winning ticket") and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights), and between $\theta_0' \odot m$ (random reinitialization) and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights) are similar (and minor) under different sparsity ratios.

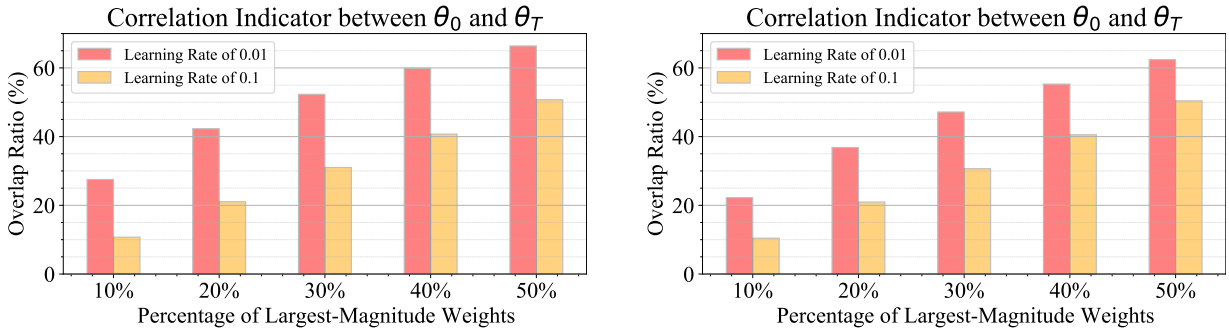
From these results we can observe the positive correlation between $\theta_0 \odot m$ and $(\theta_T \odot m)_T$ at the low learning rate, when the winning property exists. Such correlation is minor in the other cases.

E. Different Pruning Algorithms

We explore the different pruning algorithms on ResNet-20, MobileNet-V2 and VGG-11 using CIFAR-10/100. We use the desirable learning rate 0.1, $T = 150$ epochs, and the same hyperparameters introduced in Section 4.1. We compare accuracy between pruning & fine-tuning (i.e., training (fine-tuning) from $\theta_T \odot m$) and the two sparse training scenarios "winning ticket" (i.e., training from $\theta_0 \odot m$) and random reinitialization (i.e., training from $\theta_0' \odot m$) at different sparsity ratios. We investigate three pruning algorithms to derive mask m: Iterative pruning algorithm, ADMM-based pruning (Zhang et al., 2018) and one-shot pruning algorithm. We explore accuracy comparison results between pruning & fine-tuning and the two sparsity training scenarios. Figure 7 and 8 illustrate the accuracy comparison on MobileNet-V2 using CIFAR-10 and CIFAR-100, respectively. Figure 9 shows the result on ResNet-20 using CIFAR-100. Figure 10 illustrates the result on VGG-11 for CIFAR-100.

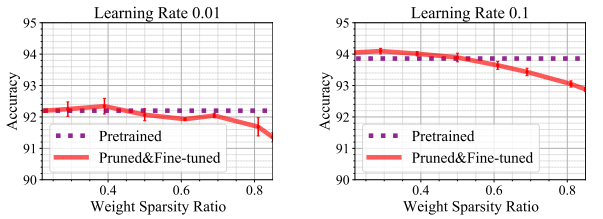


(a) VGG-11 for CIFAR-10 (b) VGG-11 for CIFAR-100

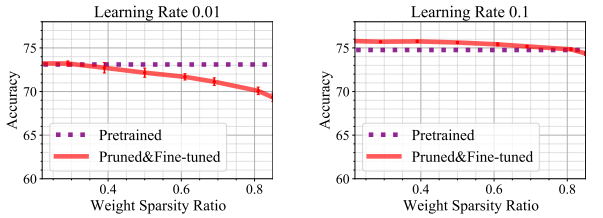


(c) MobileNet-V2 for CIFAR-10 (d) MobileNet-V2 for CIFAR-100

Figure 2. The overlap ratios (when $p = 10\%, 20\%, 30\%, 40\%$ and 50%) between the initial weights θ_0 and the pretrained weights θ_T at learning rate of 0.01 and 0.1 on VGG-11 and MobileNet-V2 using CIFAR-10/100.



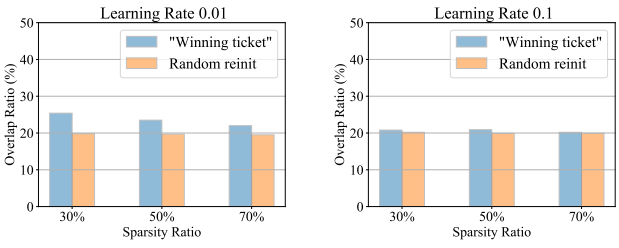
(a) MobileNet-V2 for CIFAR-10 with learning rate 0.01. (b) MobileNet-V2 for CIFAR-10 with learning rate 0.1.



(c) MobileNet-V2 for CIFAR-100 with learning rate 0.01. (d) MobileNet-V2 for CIFAR-100 with learning rate 0.1.

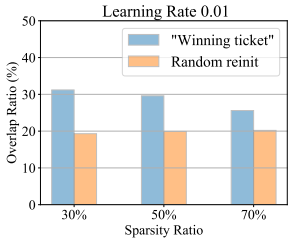
Figure 3. Accuracy of $f\left(x; (\theta_T \odot m)_T\right)$ ("pruning & fine-tuning") at different sparsity ratios with masks generated by iterative pruning on MobileNet-V2 using CIFAR-10/100.

From these results, we can clearly observe the accuracy gap between pruning & fine-tuning and the two sparse training cases (lottery ticket setting). For MobiletNet-V2 on CIFAR-100, with the masks generated from iterative pruning and ADMM-based pruning, the pruning & fine-tuning scheme can consistently outperform the pretrained dense DNN up to sparsity ratio 85%. Similarly results can be observed on VGG-11 using CIFAR-100. Meanwhile, at sparsity ratio 0.39 (39%), the pruning & fine-tuning scheme with mask

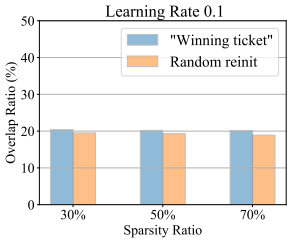


(a) ResNet-20 for CIFAR-100 with learning rate 0.01. (b) ResNet-20 for CIFAR-100 with learning rate 0.1.

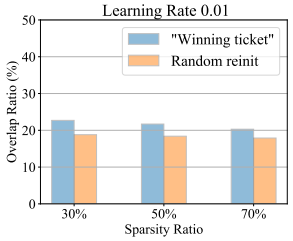
Figure 4. The weight correlation (overlap ratio) comparison at $p = 0.2$, between $\theta_0 \odot m$ ("winning ticket") and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights), and between $\theta_0' \odot m$ (random reinitialization) and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights) under 0.3, 0.5, 0.7 sparsity ratios on ResNet-20 using CIFAR-100.



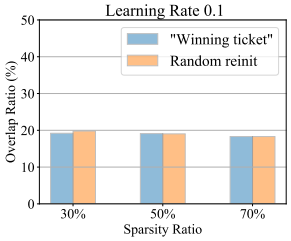
(a) VGG-11 for CIFAR-10 with learning rate 0.01.



(b) VGG-11 for CIFAR-10 with learning rate 0.1.

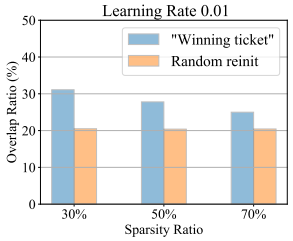


(c) VGG-11 for CIFAR-100 with learning rate 0.01.

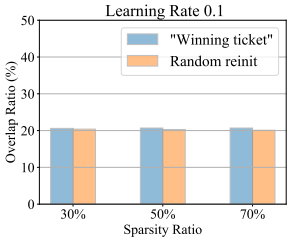


(d) VGG-11 for CIFAR-100 with learning rate 0.1.

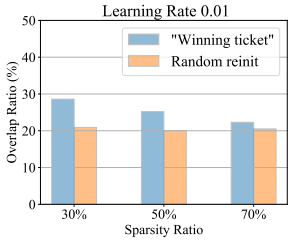
Figure 5. The weight correlation (overlap ratio) comparison at $p = 0.2$, between $\theta_0 \odot m$ ("winning ticket") and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights), and between $\theta_0' \odot m$ (random reinitialization) and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights) under 0.3, 0.5, 0.7 sparsity ratios on VGG-11 using CIFAR-10/100.



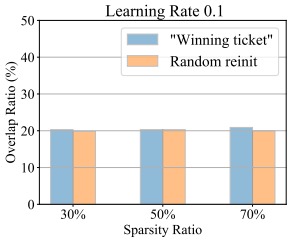
(a) MobileNet-V2 for CIFAR-10 with learning rate 0.01.



(b) MobileNet-V2 for CIFAR-10 with learning rate 0.1.



(c) MobileNet-V2 for CIFAR-100 with learning rate 0.01.



(d) MobileNet-V2 for CIFAR-100 with learning rate 0.1.

Figure 6. The weight correlation (overlap ratio) comparison at $p = 0.2$, between $\theta_0 \odot m$ ("winning ticket") and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights), and between $\theta_0' \odot m$ (random reinitialization) and $(\theta_T \odot m)_T$ (pruned&fine-tuned weights) under 0.3, 0.5, 0.7 sparsity ratios on MobileNet-V2 using CIFAR-10/100.

generated from ADMM-based pruning can achieve accuracy 76.04% while the pretrained DNN's accuracy is only 74.76% (under the desirable learning rate 0.1).

We observe the notable advantage of pruning & fine-tuning over the lottery ticket setting, even with a weak one-shot pruning algorithm for mask generation. Note there is no accuracy difference between the two sparse training cases. Pruning & fine-tuning under ADMM-based pruning can restore the accuracy of pretrained DNN with the highest sparsity ratio compared to the other two pruning algorithms. Clearly, the consistent advantage of pruning & fine-tuning is attributed to the fact that mask m is applied to pretrained weights θ_T instead of the initialized weights θ_0. In fact, information in θ_T is important for the sparse subnetwork to maintain accuracy of the pretrained dense network. Or in other words, weights in the desirable sparse subnetwork should have correlation with θ_T instead of θ_0.

Further we evaluate the relative performance (accuracy) of these three pruning algorithms. We combine the above results and demonstrate the accuracy performances of pruning & fine-tuning and sparse training ("winning ticket" case), under all three pruning algorithms. Figure 11(a) and 11(b) show the overall accuracy performance comparison on MobileNet-V2 using CIFAR-10 and CIFAR-100, respectively. Figure 12(a) shows the result on ResNet-20 using CIFAR-100 and Figure 12(b) shows the result on VGG-11 using CIFAR-100.

We observe the order in the accuracy performance: ADMM-based pruning on top, iterative pruning in the middle, and one-shot pruning the lowest. This order is the same for pruning & fine-tuning and sparse training. Note that the pruning algorithm is utilized to generate mask m, while the other conditions are the same (i.e., θ_T, fine-tuning T epochs on $\theta_T \odot m$, or sparse training on $\theta_0 \odot m$). Hence, the relative performance is attributed to the quality in mask generation. We can conclude that the selection of pruning algorithm is critical in generating the sparse subnetwork as the quality of mask generation plays a key role in the context of pruning scenario.

F. An Analysis from Weight Correlation Perspective

We provide the correlation between $(\theta_T \odot m)_T$ and θ_0, and between $(\theta_T \odot m)_T$ and θ_T for VGG-11, ResNet-20 and MobileNet-V2 using CIFAR-10/100. The training epoch $T = 150$ and the initial learning rate is 0.1. The masks are generated by ADMM-based pruning algorithm. Note that θ_0 and θ_T are dense models, while $(\theta_T \odot m)_T$ is a sparse model. To utilize the *correlation indicator*, we extend the correlation scenario of dense DNNs vs. dense DNNs to sparse DNNs vs. dense DNNs by restricting p less than

$(1-$sparsity ratio$)$ of sparse DNNs. In this experiment, we consider weight correlation at $p = 0.2$ and the sparsity ratio is 0.50 (50%) for the DNNs. The results are illustrated in Table 1. The results indicate that there is a lack of correlation between $(\theta_T \odot m)_T$ and θ_0, but there is a correlation between $(\theta_T \odot m)_T$ and θ_T. It further strengthens the conclusion that it is not desirable to have the weight correlation between final-trained weights and weight initialization.

Table 1. Weight correlation analysis at $p = 0.2$ between $(\theta_T \odot m)_T$ and θ_0, and between $(\theta_T \odot m)_T$ and θ_T for VGG-11, ResNet-20, MobileNet-V2 using CIFAR-10/100 at learning rate 0.1 under sparsity ratio 50% and the masks m are generated by ADMM-based pruning algorithm.

Model	Dataset	$R_p((\theta_T \odot m)_T, \theta_0)$	$R_p((\theta_T \odot m)_T, \theta_T)$
ResNet-20	CIFAR-100	20.36%	63.97%
MobileNet-V2	CIFAR-100	20.11%	64.71%
VGG-11	CIFAR-100	20.41%	49.32%
MobileNet-V2	CIFAR-10	20.26%	49.36%
VGG-11	CIFAR-10	20.21%	48.08%

G. Comparison with (Frankle et al., 2019)

The work Frankle et al. (2019) suggests applying mask m to θ_k and then apply sparse training, where θ_k denotes the weights trained from θ_0 for a small number of k epochs. This technique is training from $\theta_k \odot m$, and is in between sparse training (training from $\theta_0 \odot m$) and pruning & fine-tuning (training from $\theta_T \odot m$). We evaluate the relative sparse training performance among $(\theta_0 \odot m)_T$ ("winning ticket"), $(\theta_T \odot m)_T$ (pruned&fine-tuned) and $(\theta_k \odot m)_T$ ("rewind") under a desirable learning rate. We set $T = 150$, $k = 10$ and the initial learning rate is 0.1. The same hyper-parameters are adopted as introduced in Section 4.1. We study the accuracy performance comparison on MobileNet-V2, ResNet-20 and VGG-11 on CIFAR-100. We use the masks generated from the ADMM-based pruning algorithm. Figure 13 illustrates the accuracy comparison results of MobileNet-V2, ResNet-20 and VGG-11 on CIFAR-100. We can observe the order in the accuracy performance: $(\theta_T \odot m)_T$ (pruned&fine-tuned) on top, $(\theta_k \odot m)_T$ ("rewind") in the middle, and $(\theta_0 \odot m)_T$ ("winning ticket") the lowest. As they exhibit the same number of training epochs (please note that m is generated later than θ_k or θ_T), we suggest directly applying the mask m to θ_T and perform fine-tuning, instead of applying to θ_k.

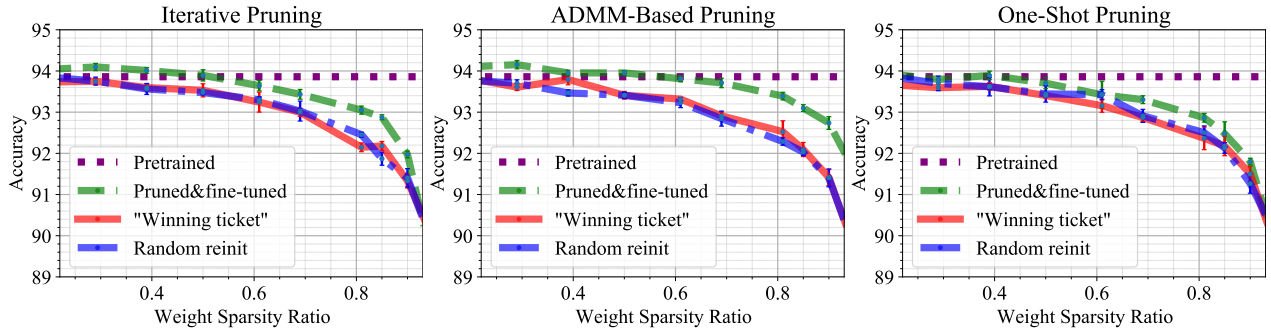


Figure 7. Accuracy of pruning & fine-tuning vs. two sparse training cases ("winning ticket" and random reinitialization) on MobileNet-V2 using CIFAR-10.

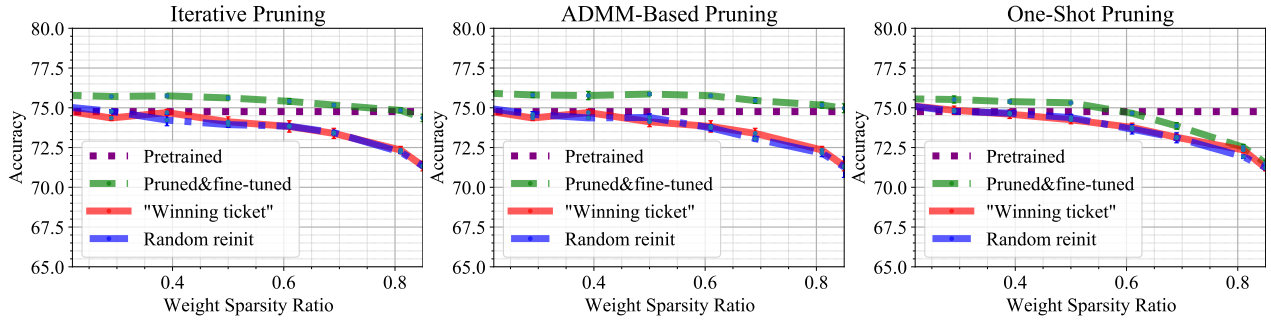


Figure 8. Accuracy of pruning & fine-tuning vs. two sparse training cases ("winning ticket" and random reinitialization) on MobileNet-V2 using CIFAR-100.

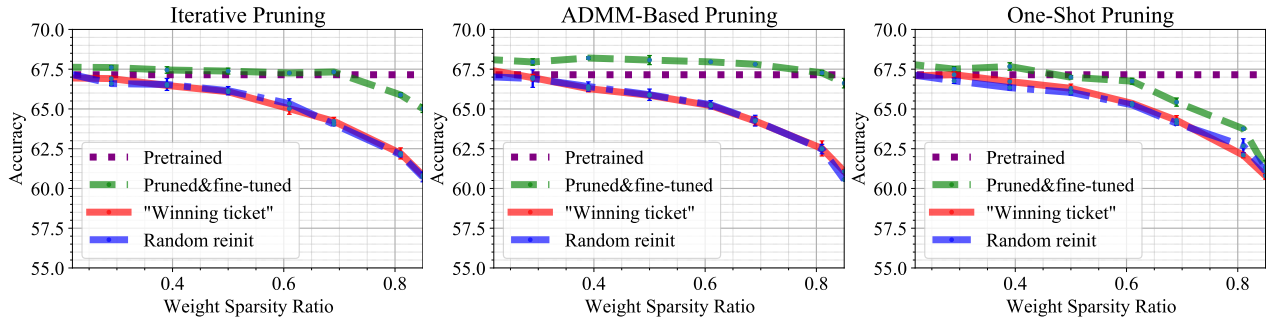


Figure 9. Accuracy of pruning & fine-tuning vs. two sparse training cases ("winning ticket" and random reinitialization) on ResNet-20 using CIFAR-100.

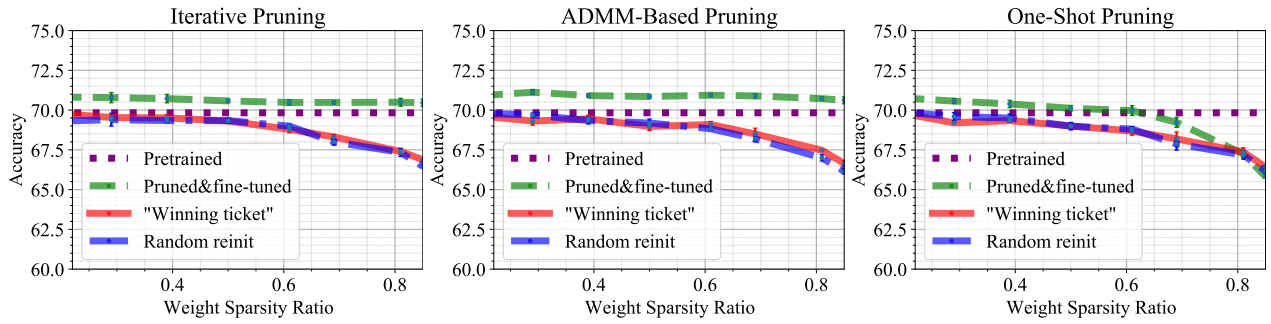
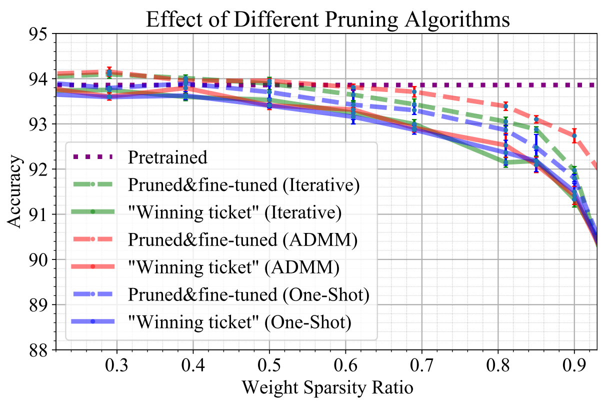
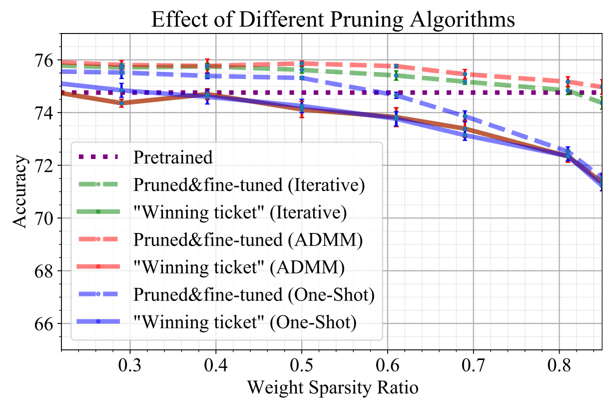


Figure 10. Accuracy of pruning & fine-tuning vs. two sparse training cases ("winning ticket" and random reinitialization) on VGG-11 using CIFAR-100.

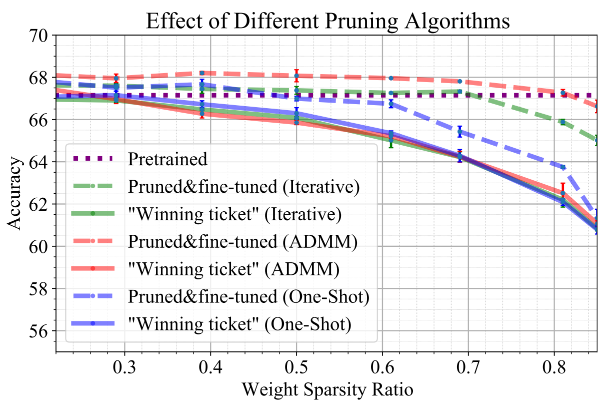


(a) MobileNet-V2 on CIFAR-10 at learning rate 0.1.

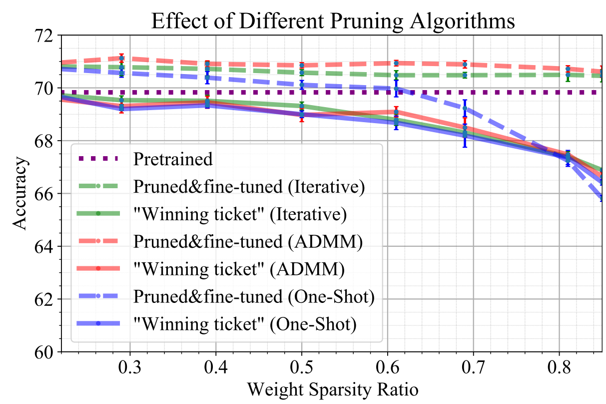


(b) MobileNet-V2 on CIFAR-100 at learning rate 0.1.

Figure 11. Accuracy of pruning & fine-tuning and sparse training ("winning ticket" case), under all three pruning algorithms (iterative pruning, ADMM-based pruning, and one-shot pruning) for mask generation on MobileNet-V2 using CIFAR-10/100.

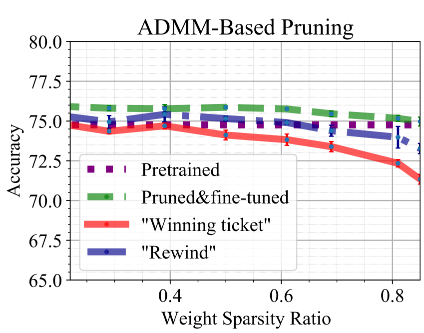


(a) ResNet-20 on CIFAR-100 at learning rate 0.1.

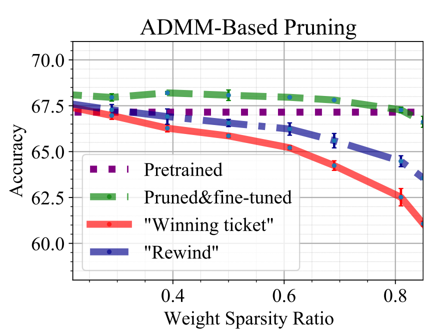


(b) VGG-11 on CIFAR-100 at learning rate 0.1.

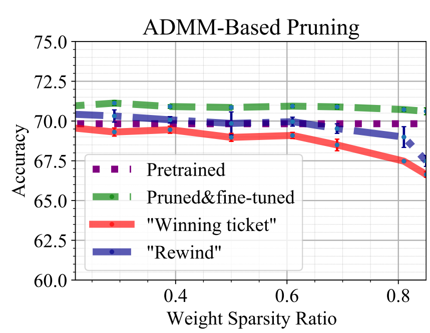
Figure 12. Accuracy of pruning & fine-tuning and sparse training ("winning ticket" case), under all three pruning algorithms (iterative pruning, ADMM-based pruning, and one-shot pruning) for mask generation on ResNet-20 and VGG-11 using CIFAR-100.



(a) MobileNet-V2 for CIFAR-100 at learning rate of 0.1.



(b) ResNet-20 for CIFAR-100 at learning rate of 0.1.



(c) VGG-11 for CIFAR-100 at learning rate of 0.1.

Figure 13. Accuracy performance of $(\theta_T \odot m)_T$ (pruned&fine-tuned), $(\theta_k \odot m)_T$ ("rewind") and $(\theta_0 \odot m)_T$ ("winning ticket") on MobileNet-V2, ResNet-20 and VGG-11 for CIFAR-100 over a range of different sparsity ratios. The masks are generated by ADMM-based pruning algorithm and the initial learning rate is 0.1.