THE UNREASONABLE EFFECTIVENESS OF RANDOM PRUNING: RETURN OF THE MOST NAIVE BASELINE FOR SPARSE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Random pruning is arguably the most naive way to attain sparsity in neural networks, but has been deemed uncompetitive by either post-training pruning or sparse training. In this paper, we focus on sparse training and highly a perhaps counter-intuitive finding, that random pruning at initialization (PaI) can be quite powerful for the sparse training of modern neural networks. Without any delicate pruning criteria or carefully pursued sparsity structures, we empirically demonstrate that sparsely training a randomly pruned network from scratch can match the performance of its dense equivalent. There are two key factors that contribute to this revival: (i) *the network sizes matter*: as the original dense networks grow wider and deeper, the performance of training a randomly pruned sparse network will quickly grow to matching that of its dense equivalent, even at high sparsity ratios; (ii) *appropriate layer-wise sparsity ratios* can be pre-chosen for sparse training, which shows to be another important performance booster. Simple as it looks, a randomly pruned subnetwork of Wide ResNet-50 can be sparsely trained to outperforming a dense Wide ResNet-50, on ImageNet. We also observed such randomly pruned networks outperform dense counterparts in other favorable aspects, such as out-of-distribution detection, uncertainty estimation, and adversarial robustness. Overall, our results strongly suggest there is larger-than-expected room for sparse training at scale, and the benefits of sparsity might be more universal beyond carefully designed pruning. We have included our codes as supplementary files.

1 INTRODUCTION

Most recent breakthroughs in deep learning are fairly achieved with the increased complexity of over-parameterized networks (Brown et al., 2020; Raffel et al., 2019; Dosovitskiy et al., 2020; Fedus et al., 2021; Jumper et al., 2021; Berner et al., 2019). It is well-known that large models train better (Neyshabur et al., 2018; Novak et al., 2018; Allen-Zhu et al., 2019), generalize better (Hendrycks & Dietterich, 2019; Xie & Yuille, 2019; Zhao et al., 2017), and transfer better (Chen et al., 2020; 2021). However, the upsurge in the investment of large models exacerbates the gap between research and real-life applications since many practical applications demand compact and efficient networks.

Neural network pruning, since proposed by (Mozer & Smolensky, 1989; Janowsky, 1989), has evolved as the most common technique in literature to reduce the computational and memory requirements of neural networks. Over the past few years, numerous pruning criteria have been proposed, including magnitude (Mozer & Smolensky, 1989; Han et al., 2015; Frankle & Carbin, 2018; Mocanu et al., 2018), Hessian (LeCun et al., 1990; Hassibi & Stork, 1993), mutual information (Dai et al., 2018), Taylor expansion (Molchanov et al., 2016), movement (Sanh et al., 2020), connection sensitivity (Lee et al., 2019), etc. Motivated for different scenarios, pruning can occur after training (Han et al., 2015; Frankle & Carbin, 2018; Molchanov et al., 2016; Lee et al., 2021), during training (Zhu & Gupta, 2017; Gale et al., 2019; Louizos et al., 2018), and even before training (Mocanu et al., 2018; Lee et al., 2019; Gale et al., 2019). The last regime can be further categorized into "static sparse training" (Mocanu et al., 2016; Gale et al., 2019; Lee et al., 2019; Wang et al., 2020) and "dynamic sparse training" (Mocanu et al., 2018; Bellec et al., 2018; Evci et al., 2020a; Liu et al., 2021).

While random pruning is a universal method that can happen at any stage of training, training a randomly pruned network from scratch is arguably the most appealing way, owing to its "end-to-end" saving potential for the entire training process besides the inference. Due to this reason, we focus on random pruning at initialization (or sparse training) in this paper. When new pruning approaches bloom, random pruning naturally becomes their performance's empirical "lower bound" since the connections are randomly chosen without any good reasoning. Likely due to the same reason, the results of random pruning at initialization (as "easy to beat" baselines to support fancier new pruning methods) reported in the literature are unfortunately vague, often inconsistent, and sometimes casual. For instance, it is found in Liu et al. (2020b) that randomly pruned sparse networks can be trained from scratch to match the full accuracy of dense networks with only 20% parameters, whereas around 80% parameters are required to do so in Frankle et al. (2021). The differences may arise from architecture choices, training recipes, distribution hyperparameters/layer-wise ratios, and so on.

In most pruning literature (Gale et al., 2019; Lee et al., 2019; Frankle et al., 2021; Tanaka et al., 2020), random pruning usually refers to randomly removing the same proportion of parameters per layer, ending up with uniform layer-wise sparsities. Nevertheless, researchers have explored other pre-defined layer-wise sparsities, e.g., uniform+ (Gale et al., 2019), *Erdős-Rényi* random graph (ER) (Mocanu et al., 2018), and *Erdős-Rényi-Kernel* (ERK) (Evci et al., 2020a). These layer-wise sparsities naturally fit the category of random pruning, as they require no training to obtain the corresponding sparsity ratios (followed by sampling randomly). We assess random pruning at initialization with these layer-wise sparsity ratios, in terms of various perspectives besides the predictive accuracy.

Our main findings during this course of study are summarized below:

- We find that the network size matters for the effectiveness of random pruning at initialization. With small networks, random pruning at initialization hardly matches the full accuracy even at mild sparsities (10%, 20%). However, as the networks grow wider and deeper, the performance of training a randomly pruned sparse network will quickly grow to matching that of its dense equivalent, even at high sparsity ratios.
- We further identify that appropriate layer-wise sparsity ratios can be an important booster for the performance of random pruning at initialization. We investigate several options to pre-define layer-wise sparsity ratios before any training; one of them is able to push the performance of a completely random sparse Wide ResNet-50 over the densely trained ResNet-50 on ImageNet.
- We systematically assess the performance of random pruning at initialization and are somehow surprised by its accuracy and robustness. The accuracy achieved by ERK ratio can even surpass the ones learned by complex criteria, e.g., SNIP, GraSP. In addition, randomly pruned sparse networks are found to outperform dense networks in other favorable aspects, such as out-of-distribution (OoD) detection, adversarial robustness, and uncertainty estimation.

2 Related Work

2.1 Static Sparse Training

Static sparse training represents a class of methods that aim to train a sparse subnetwork with a fixed sparse connectivity pattern during the course of training. We divide the static sparse training into random pruning and non-random pruning according to whether the connection is randomly selected.

Random Pruning. Static sparse training with random pruning samples masks within each layer in a random fashion based on pre-defined layer-wise sparsities. The most naive approach is pruning each layer uniformly with the same pruning ratio, i.e., uniform pruning (Mariet & Sra, 2015; He et al., 2017; Suau et al., 2019; Gale et al., 2019). Mocanu et al. (2016) proposed a non-uniform and scale-free topology, showing better performance than the dense counterpart when applied to restricted Boltzmann machines (RBMs). Later, expander graphs were introduced to build sparse CNNs and showed comparable performance against the corresponding dense CNNs (Prabhu et al., 2018; Kepner & Robinett, 2019). While not initially designed for static sparse training, ER (Mocanu et al., 2018) and ERK (Evci et al., 2020a) are two advanced layer-wise sparsities introduced from the field of graph theory with strong results.

Non-Random Pruning. Instead of pre-choosing a sparsity ratio for each layer, many works utilize the proposed saliency criteria to learn the layer-wise sparsity ratios before training, also termed as pruning at initialization (PaI). Lee et al. (2019) first introduced SNIP that chooses structurally important connections at initialization via the proposed connection sensitivity. Following SNIP, many efficient criteria have been proposed to improve the performance of non-random pruning at initialization, including but not limited to gradient flow (Wang et al., 2020), synaptic strengths (Tanaka et al., 2020), neural tangent kernel (Liu & Zenke, 2020), and iterative SNIP (de Jorge et al., 2020; Verdenius et al., 2020).

The most closely related works of our work are Su et al. (2020) and Frankle et al. (2021). Su et al. (2020) proposed several sanity-check and showed that SNIP, GraSP, and Lottery Tickets (LT)[1] hardly exploit any information from the training data and very robust to the mask shuffling. They further show that random pruning at initialization with hand-designed layer-wise sparsity ratios outperforms or attains a similar performance compared with PaI methods. Frankle et al. (2021) further confirmed that what PaI discovers are the layerwise proportions rather than the specific weights or values, whereas magnitude pruning after training learns both, reflecting a broader challenge inherent to pruning at initialization. While prior works have observed that random pruning can be more competitive in certain cases, they did not give principled guidelines on when and how it can become that good; nor do they show it can match the performance of dense networks on ImageNet. Standing on the shoulders of giants, our work summarized principles by more rigorous studies, and demonstrate the strongest result so far, that randomly pruned sparse Wide ResNet-50 can be sparsely trained to outperform a dense Wide ResNet-50, on ImageNet. Moreover, compared with the ad-hoc sparsity ratios used in (Su et al., 2020), we show that ERK (Evci et al., 2020a) and our modified ERK+ are more general sparsity ratios that consistently demonstrate competitive performance without careful layer-wise sparsity design for every architecture. Specifically, ERK+ ratio achieves similar accuracy with the dense Wide ResNet-50 on ImageNet while being data free, feedforward free, and dense initialization free.

2.2 Dynamic Sparse Training

In contrast to static sparse training, dynamic sparse training stems from randomly initialized sparse subnetworks, and meanwhile dynamically explores new sparse connectivity during training. Dynamic sparse training starts from Sparse Evolutionary Training (SET) (Mocanu et al., 2018; Liu et al., 2020a) which initializes the sparse connectivity with *Erdős-Rényi* (Erdős & Rényi, 1959) topology and periodically explores the sparse connectivity via a prune-and-grow scheme during the course of training. While there exist numerous pruning criteria in the literature, simple magnitude pruning typically performs well in the field of dynamic sparse training. On the other hand, the criteria used to grow weights back differs from method to method, including randomness (Mocanu et al., 2018; Mostafa & Wang, 2019), momentum (Dettmers & Zettlemoyer, 2019), gradient (Evci et al., 2020a; Jayakumar et al., 2020; Liu et al., 2021). Besides the prune-and-grow scheme, layer-wise sparsities are vital to achieving high accuracy. (Mostafa & Wang, 2019; Dettmers & Zettlemoyer, 2019) reallocates weights across layers during training based on reasonable heuristics, demonstrating performance improvement. Evci et al. (2020a) extended ER to CNNs and showed considerable performance gains to sparse CNN training with the *Erdős-Rényi-Kernel* (ERK) ratio. Even though dynamic sparse training achieves promising sparse training performance, it changes the sparse connectivity during training and is thus out of the scope of random pruning.

3 Methodology

We conduct extensive experiments to systematically evaluate the performance of random pruning. The experimental settings are described below.

3.1 Layer-wise Sparsities Ratios

Denote s^l as the sparsity of layer l. Random pruning, namely, removes weights or filters in each layer randomly to the target sparsity s^l. Different from works that learn layer-wise sparsities and model weights together during training using iterative global pruning techniques (Frankle & Carbin, 2018),

[1] Lottery Tickets in Su et al. (2020) refers to the original Lottery Tickets Hypothesis proposed in Frankle & Carbin (2018), which retrains the discovered subnetwork with the original initialization.

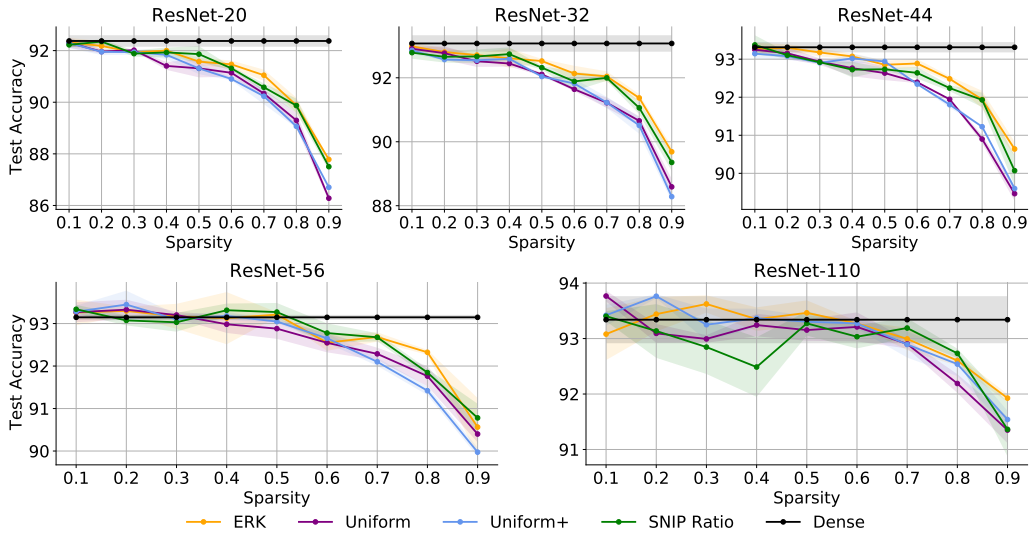

Figure 1: **From shallow to deep.** Test accuracy of training randomly pruned subnetworks from scratch with different depth on CIFAR-10. ResNet-A refers to a ResNet model with A layers in total.

soft threshold (Kusupati et al., 2020), and dynamic reparameterization (Mostafa & Wang, 2019), etc., the layer-wise sparsities of random pruning is pre-defined before training. Many pre-defined layer-wise sparsity ratios in the literature are suited for random pruning, while they may not initially be designed for random pruning. We choose the following 6 layer-wise sparsity ratios to study.

ERK. Introduced by Mocanu et al. (2018), Erdős-Rényi (ER) sparsifies the Multilayer Perceptron (MLP) with a random topology in which larger layers are allocated with higher sparsity then smaller layers. Evci et al. (2020a) further proposed a convolutional variant (Erdős-Rényi-Kernel (ERK)) that takes the convolutional kernel into consideration. Specifically, the sparsity of the convolutional layer is scaled proportional to $1 - \frac{n^{l-1}+n^l+w^l+h^l}{n^{l-1}\times n^l \times w^l \times h^l}$ where n^l refers to the number of neurons/channels in layer l; w^l and h^l are the corresponding width and height.

ERK+. We modify ERK by forcing the last fully-connected layer as dense if it is not, while keeping the overall parameter count the same. Doing so improves the test accuracy of Wide ResNet-50 on ImageNet[2] as shown in Section 4.2.

Uniform. Each layer is pruned with the same pruning rate so that the pruned network ends up with a uniform sparsity distribution, e.g., Zhu & Gupta (2017).

Uniform+. Instead of using a completely uniform sparsity ratio, Gale et al. (2019) keep the first convolutional layer dense and maintain at least 20% parameters in the last fully-connected layer.

SNIP ratio. SNIP selects important weights based on the connection sensitivity score $|g \odot w|$, where w and g is the network weight and gradient, respectively. The weights with the lowest scores in *one iteration* are pruned before training. While not initially designed for random pruning, we adjust SNIP for random pruning by only keeping its layer-wise sparsity ratios, while discarding its mask positions. New mask positions (non-zero elements) are re-sampled through a uniform distribution $\sim \text{Uniform}(0, 1)$ with a probability of $1 - s^l$. Such layer-wise sparsities are obtained in a (slightly) data-driven way, yet very efficiently *before training* without any weight update. The SNIP ratio is then treated as another pre-defined (i.e., before the training starts) sampling ratio for random pruning.

GraSP ratio. GraSP is an other method that pursues network pruning before training. Specifically, GraSP removes weights that have the least effect on the gradient norm based on the score of $-w \odot Hg$, where H is the Hessian matrix and g is the gradient. Same with SNIP, we keep the layer-wise sparsity ratios of GraSP and re-sample its mask positions.

[2]For datasets with fewer classes, e.g, MNIST, CIFAR-10, ERK+ scales back to ERK as the last fully-connected layer is already dense.

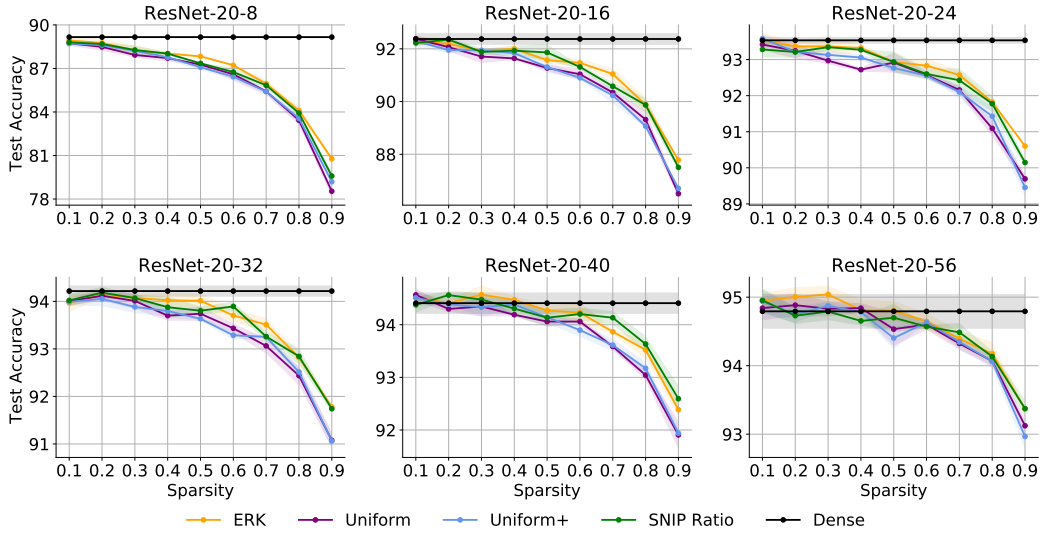

Figure 2: **From narrow to wide.** Test accuracy of training randomly pruned subnetworks from scratch with different width on CIFAR-10. ResNet-A-B refers to a ResNet model with A layers in total and B filters in the first convolutional layer.

3.2 Architectures, Datasets, and Implementations

Table 1: Summary of the architectures, datasets and hyperparameters used in this paper.

Model	Mode	Data	#Epoch	Batch Size	LR	Momentum	LR Decay, Epoch	Weight Decay
ResNets	Dense	CIFAR-10/100	160	128	0.1	0.9	10×, [80, 120]	0.0005
	Sparse	CIFAR-10/100	160	128	0.1	0.9	10×, [80, 120]	0.0005
Wide ResNets	Dense	ImageNet	90	192*4	0.4	0.9	10×, [30, 60, 80]	0.0001
	Sparse	ImageNet	100	192*4	0.4	0.9	10×, [30, 60, 90]	0.0001

Our main experiments are conducted with the CIFAR version of ResNet (He et al., 2016) with varying depths and widths on CIFAR-10/100 (Krizhevsky et al., 2009), VGG with varying depths on CIFAR-10/100, and the ImageNet version of ResNet and Wide ResNet-50 (Zagoruyko & Komodakis, 2016) on ImageNet (Deng et al., 2009). For ImageNet, we follow the common setting in sparse training (Dettmers & Zettlemoyer, 2019; Evci et al., 2020b; Liu et al., 2021) and train sparse models for 100 epochs. All models are trained with stochastic gradient descent (SGD) with momentum. We share the summary of the architectures, datasets, and hyperparameters in Table 1.

4 Experimental Results

In this section, we report the results of random pruning under various evaluation measurements. For each measurement, the trade-off between sparsity and the corresponding metric is reported. Moreover, we also alter the depth and width of the model to check how the performance alters as the model size alters. The results of CIFAR-10/100 are averaged over 3 runs, whereas the experiments of ImageNet are only ran once due to the limited resources. We report the results on CIFAR-10 and ImageNet of ResNet in the main body of this paper. The results on CIFAR-100 of ResNet and CIFAR-10/100 of VGG are shown in Appendix C and Appendix D, respectively. Unless otherwise stated, the results of CIFAR-100 are qualitatively similar with CIFAR-10. We provide a scaled-up overview of the accuracy and parameter count with various ResNet on ImageNet in Appendix E. We report the trade-off between test accuracy and FLOPs of various randomly pruned ResNets and the corresponding dense ResNets on ImagetNet in Appendix F. We see that the random pruning gets larger efficiency (the difference between the test FLOPs of sparse models and dense models) with the increased model size on ImageNet.

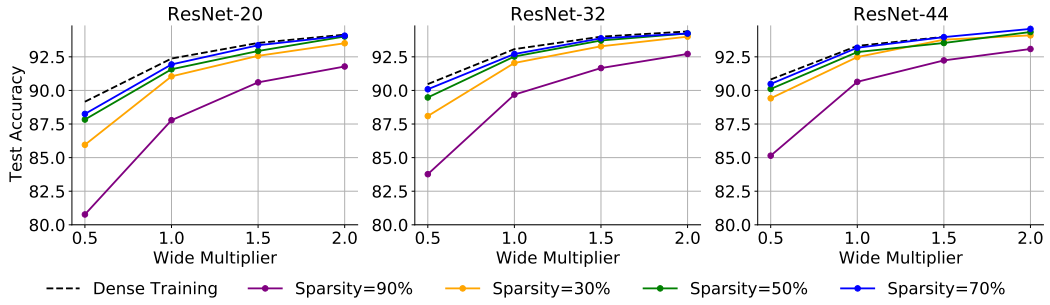

Figure 3: Test accuracy of training randomly pruned ResNet-20, ResNet-32, and ResNet-44 from scratch with ERK ratios when model width varies on CIFAR-10.

4.1 Predictive Accuracy on CIFAR-10

We first demonstrate the performance of random pruning on the most common metric – test accuracy. To avoid overlapping of multiple curves, we share the results of GraSP in Appendix B. The main observations are as follows:

① **Performance of random pruning improves with the size of the network.** We vary the depth of ResNet and report the test accuracy achieved by various pruning methods in Figure 1. When operating on small networks, e.g., ResNet-20 and ResNet-32, we can hardly find matching subnetworks even at mild sparsities, i.e., 10%, 20%. With larger networks, e.g., ResNet-56 and ResNet-110, random pruning can match the dense performance at 60% $\sim$ 70% sparsity.

Similar behavior can also be observed when we increase the width of ResNet-20. Training with random pruning can just match the full accuracy as the model width increases as shown in Figure 2. This encouraging trend is also shared with deeper architectures i.e., ResNet-32, ResNet-44 in Figure 3. The performance gap between densely trained networks and randomly pruned networks shrinks quickly as the model gets wider.

② **The performance difference between different pruning methods becomes indistinct as the model size increases.** Even though uniform sparsities fail to match the accuracy achieved by non-uniform sparsities (ERK and SNIP) with small models, their test accuracy raises to a comparable level as non-uniform sparsities with large models, e.g., ResNet-110 and ResNet-20-56.

③ **Random pruning with ERK even outperforms pruning with the well-versed methods (SNIP, GraSP).** Without using any information, e.g., gradient and magnitude, training a randomly pruned subnetwork with ERK topology leads to expressive accuracy, even better than the ones trained with the delicately designed sparsity ratios, i.e., SNIP, and GraSP (shown in Appendix B).

4.2 Predictive Accuracy on ImageNet

We have learned from Section 4.1 that larger networks result in stronger random subnetworks on CIFAR-10/100. We are also interested in how far we can go with random pruning on large-scale datasets, i.g., ImageNet (Deng et al., 2009), a non-saturated dataset on which deep neural networks are less over-parameterized than on CIFAR-10/100.

Performance improvement of ERK+. We first demonstrate the performance improvement on ImageNet caused by ERK+. As mentioned earlier, ERK naturally allocates higher sparsities to the larger layers while allocating lower sparsities to the smaller ones. As a consequence, the last fully-connected layer is very likely to be dense for the datasets with only a few classes, e.g., CIFAR-10. However, for the datasets with a larger number of classes, e.g., ImageNet, the fully-connected layer is usually sparse. We empirically find that allocating more parameters to the last fully-connected layer while keeping the overall parameter count fixed leads to higher accuracy for ERK on ImageNet. Test accuracy of ERK with Wide ResNet-50 at various sparsity of the last layer is reported in Figure 4-left. Consequently, we keep the last fully-connected layer of ERK dense for ImageNet and term this modified variant as ERK+.

Compare random pruning with its dense equivalents. To draw a more solid conclusion, we train large, randomly pruned Wide ResNet-50 on ImageNet and compare it to the dense equivalents with

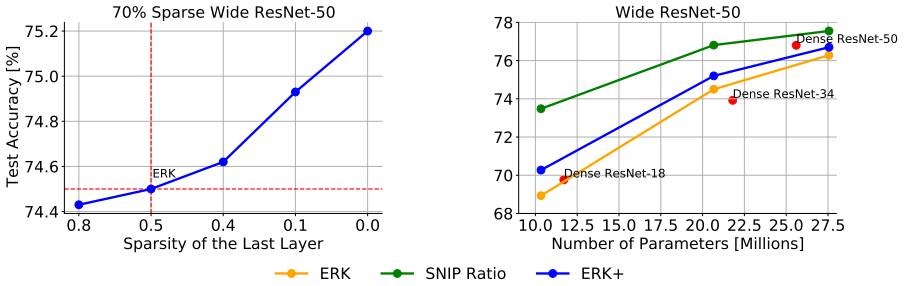

Figure 4: **Test accuracy of Wide ResNet-50 on ImageNet.** *Left:* Performance of ERK with various sparsity of the last fully-connected layer. We vary the sparsity of the last fully-connected layer while keeping the overall sparsity fixed as 70%. Results of the original ERK are indicated with dashed red lines. *Right:* Performance comparison between randomly pruned Wide ResNet-50 and the corresponding dense equivalents with a similar parameter count.

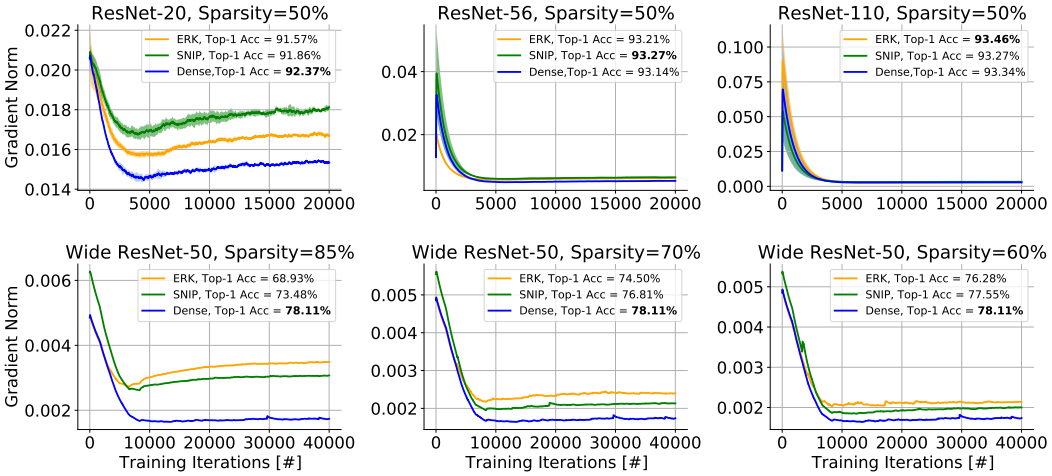

Figure 5: **Gradient norm of randomly pruned networks during training.** *Top:* Comparison between gradient norm of SNIP and ERK with 50% sparse ResNet-20, ResNet-56, and ResNet-110 on CIFAR-10. *Bottom:* Comparison between gradient norm of SNIP and ERK with Wide ResNet-50 on ImageNet at various sparsities.

the same number of parameters on ImageNet. As shown in Figure4-right, all randomly pruned networks outperform the dense ResNet-34 with the same number of parameters. ERK+ consistently achieves higher accuracy than ERK, even closely approaching the strong baseline – dense ResNet-50. More interestingly, the layer-wise sparsities discovered by SNIP boost the accuracy of sparse Wide ResNet-50 over dense ResNet-50, highlighting the importance of layer-wise sparsity ratios on sparse training . Given the fact that the performance gap between SNIP ratio and ERK on CIFAR-10 is somehow vague, our results highlight the necessity of evaluating any proposed pruning methods with large-scale models and datasets, e.g., ResNet-50 on ImageNet.

4.3 Understanding Random Pruning via Gradient Flow

The performance gap between the ERK and SNIP ratio on ImageNet raises the question – what benefits do the layer-wise sparsities of SNIP provide to sparse training? Since SNIP takes gradients into account when determines the layer-wise sparsities, we turn to gradient flow in the hope of finding some insights on this question. The form of gradient norm we choose is the effective gradient flow (Tessera et al., 2021) which only calculates the gradient norm of active weights. We measure the gradient norm of sparse networks generated by SNIP and ERK[3] during the early training phase. The results are depicted in Figure 5.

[3]The gradient norm patterns of ERK and ERK+ are too similar to distinguish. After specifically zooming in the initial training phase, we find that the gradient norm of ERK+ is slightly higher than ERK.

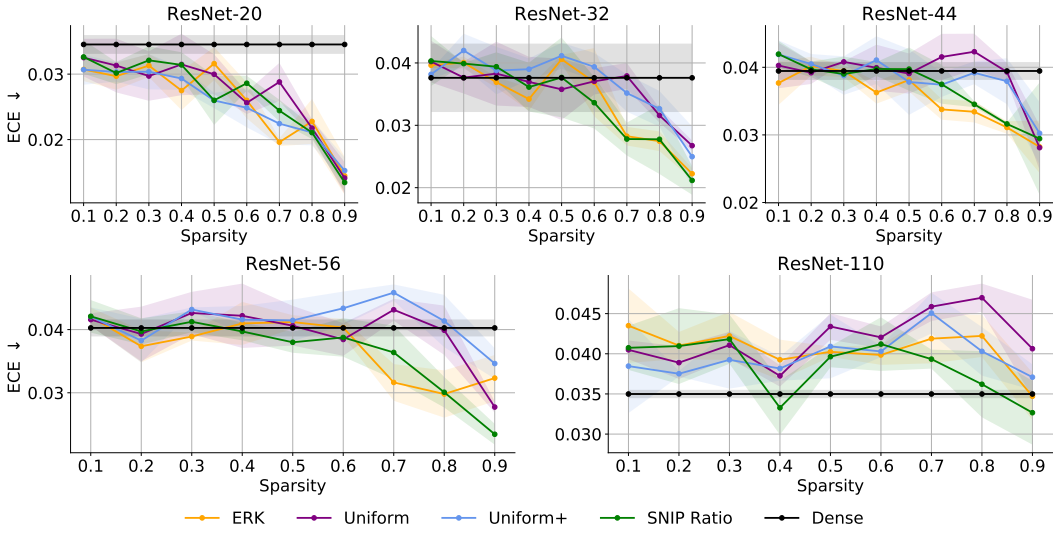

Figure 6: **Uncertainty estimation (ECE).** The experiments are conducted with various models on CIFAR-10. Lower ECE values represent better uncertainty estimation.

Overall, we see that the SNIP ratio indeed brings benefits to the gradient norm at the initial stage compared with ERK. Considering only SNIP (green lines) and ERK (orange lines), the ones with higher gradient norm at the beginning always achieve higher final accuracy on both CIFAR-10 and ImageNet. This result is on par with prior work (Wang et al., 2020), which claims that increasing the gradient norm at the beginning likely leads to higher accuracy. However, this observation does not hold for dense Wide ResNet-50 on ImageNet (blue lines). Even though dense models have a lower gradient norm than SNIP, they consistently outperform SNIP in accuracy by a large margin.

Instead of focusing on the initial phase, we note that the gradient norm early in the training (the flat phase after gradient norm drops) could better understand the behaviour of networks trained under different scenarios. We empirically find that final performance gaps between sparse networks and dense networks is highly correlated with gaps of gradient norm early in training. Training with small networks (e.g., ResNet-20) on CIFAR-10 results in large performance and gradient norm gaps between sparse networks and dense networks, whereas these two gaps simultaneously vanish when trained with large networks, e.g., ResNet-56 and ResNet-110. Similar behavior can be observed in Wide ResNet-50 on ImageNet in terms of sparsity. Both the performance and gradient norm gaps decrease gradually as the sparsity level drops from 85% to 60%. This actually makes sense, since large networks (either large model size or large number of parameters) are likely to still be over-parameterized after pruning, so that all pruning methods (including random pruning) can preserve gradient flow along with test accuracy equally well.

Our findings suggest that only considering gradient flow at initialization might not be sufficient for sparse training. More efforts should be invested to study the properties that sparse network training misses after initialization, especially the early stage of training (Frankle et al., 2020b). Techniques that change the sparse pattern during training (Mocanu et al., 2018; Evci et al., 2020a) and weight rewinding (Frankle et al., 2020a; Renda et al., 2020) could serve as good starting points.

4.4 Broader Evaluation of Random Pruning

In most pruning literature, test accuracy is the core quality that researchers consider. However, the evaluation of other perspectives is also important for academia and industry before replacing dense networks with sparse networks on a large scope. Therefore, we thoroughly evaluate random pruning from a broader perspective. Specifically, we assess random pruning from perspectives of out-of-distribution (OoD) performance (Hendrycks et al., 2020), adversarial robustness (Goodfellow et al., 2014), and uncertainty estimation (Lakshminarayanan et al., 2016).

Measurements. We briefly introduce each measurement and the corresponding metric used here. For more details, please see Appendix A. Formally, let's denote a (sparse) network $f : \mathcal{X} \to \mathcal{Y}$ trained on samples from distribution $\mathcal{D}$. $f(x, \theta)$ and $f(x, \theta \odot m)$ refers to the dense network and the

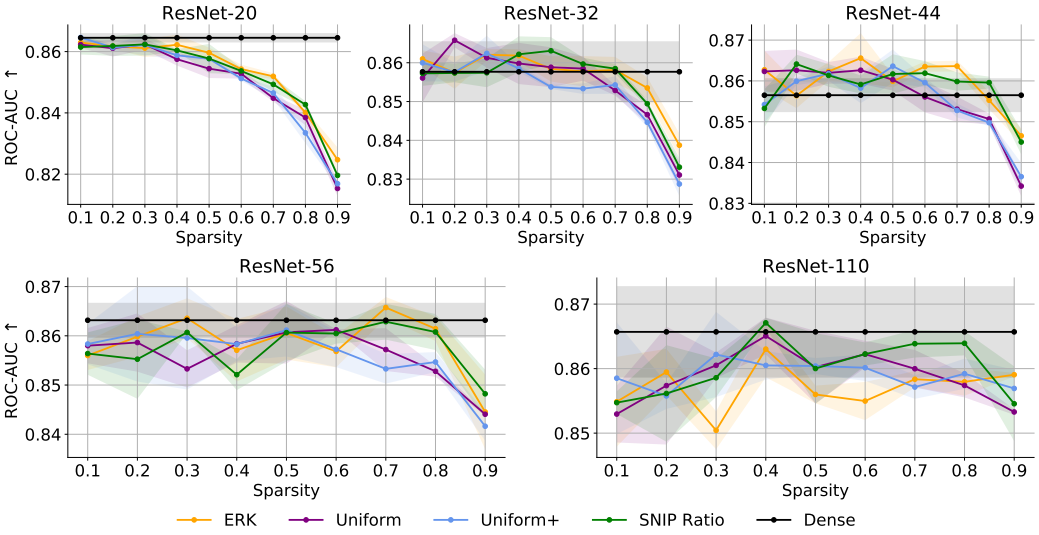

Figure 7: **Out-of-distribution performance (ROC-AUC).** Experiments are conducted with models trained on CIFAR-10, tested on CIFAR-100. Higher ROC-AUC refers to better OoD performance.

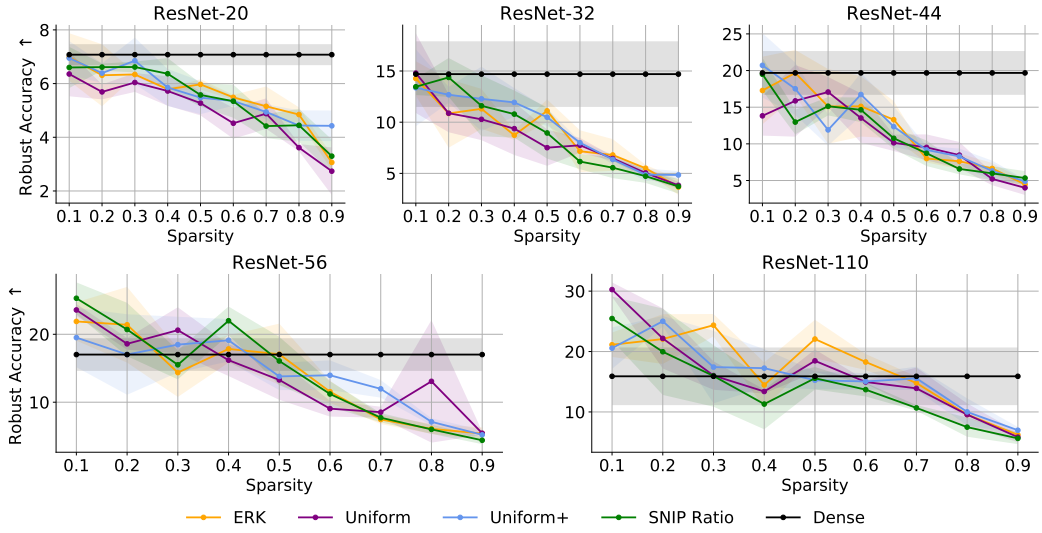

Figure 8: **Adversarial robustness.** The experiments are conducted with various models on CIFAR-10. Higher values represent better adversarial robustness.

pruned network, respectively. To evaluate whether random pruning brings benefits to OoD, we follow the classic routines (Augustin et al., 2020; Meinke & Hein, 2019), adopting SVHN (Netzer et al., 2011), CIFAR-100, and CIFAR-10 with random Gaussian noise (Hein et al., 2019) as OoD datasets for models trained on CIFAR-10; ImageNet-O as the OoD dataset for models trained on ImageNet. For adversarial robustness, we use Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) to generate adversarial samples with a perturbation strength of $\frac{8}{255}$. We choose two widely-used metrics, expected calibration error (ECE) (Guo et al., 2017) and negative log-likelihood (NLL) (Friedman et al., 2001) for the uncertainty estimation.

Due to the limited space, we report the results of CIFAR-10 in the main paper. The results on ImageNet are reported in Appendix G. In general, random pruning achieves quite strong results on uncertainty estimation, OoD robustness, and adversarial robustness without any fancy techniques. We summary main observations as below:

① Randomly pruned networks enjoy better uncertainty estimation than their dense counterparts. Figure 6 shows that randomly pruned ResNet-20 matches or even improves the uncertainty

estimation of dense networks with a full range of sparsities. ECE of random pruning grows with the model size, in line with the finding of dense networks in Guo et al. (2017). Still, random pruning is capable of sampling matching subnetworks with high sparsities (e.g., 80%) except for the largest model, ResNet-110. Results of NLL are presented in Appendix H.2, where randomly pruned networks also match NLL of dense networks at extremely high sparsities.

② **Random pruning produces extremely sparse yet robust subnetworks on OoD.** Figure 7 plots the results of networks trained on CIFAR-10, tested on CIFAR-100. The other two sets of experiments are reported in Appendix H.1. Large network sizes significantly improve the OoD performance of random pruning. As the model size increases, random pruning can match the OoD performance of dense networks with only 20% parameters. Again, SNIP ratio and ERK outperform uniform sparsities in this setting.

③ **Random pruning improves the adversarial robustness of large models.** The adversarial robustness of large models (e.g., ResNet-56 and ResNet-110) improves significantly with mild sparsities in Figure 8. One explanation here is that, while achieving high clean accuracy, these over-parameterized large models are highly overfitted on CIFAR-10, and thus suffer from poor performance on adversarial examples (shown by Tsipras et al. (2018); Zhang et al. (2019) as well). Sparsity induced by random pruning serves as a cheap type of regularization which possibly mitigates this overfitting problem.

5 Conclusion

In this work, we systematically revisit the underrated baseline of sparse training – random pruning. Our results highlight a counter-intuitive finding, that is, training a randomly pruned network from scratch without any delicate pruning criteria can be quite performant. With proper network sizes and layer-wise sparsity ratios, random pruning can match the performance of dense networks even at extreme sparsities. Impressively, a randomly pruned subnetwork of Wide ResNet-50 can be trained to outperforming a strong benchmark, dense ResNet-50 on ImageNet. Moreover, training with random pruning intrinsically brings significant benefits to other desirable aspects, such as out-of-distribution, uncertainty estimation and adversarial robustness. Our paper indicates that in addition to appealing performance, large models also enjoy strong robustness to pruning. Even if we pruning with complete randomness, large models can preserve their performance well.

6 Reproducibility

We have shared the architectures, datasets, and hyperparameters used in this paper in Section 3.2. Besides, the different measurements and metrics used in this paper are shared in Appendix A. We have also submitted our code as supplementary files.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in-and out-distribution improves explainability. In *European Conference on Computer Vision*, pp. 228–245. Springer, 2020.

Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. Deep rewiring: Training very sparse deep networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ_wN01C-.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Dkebiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16306–16316, 2021.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

Bin Dai, Chen Zhu, Baining Guo, and David Wipf. Compressing neural networks using the variational information bottleneck. In *International Conference on Machine Learning*, pp. 1135–1144. PMLR, 2018.

Pau de Jorge, Amartya Sanyal, Harkirat S Behl, Philip HS Torr, Gregory Rogez, and Puneet K Dokania. Progressive skeletonization: Trimming more fat from a network at initialization. *arXiv preprint arXiv:2006.09081*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Tim Dettmers and Luke Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Paul Erdős and Alfréd Rényi. On random graphs i. *Publicationes Mathematicae (Debrecen)*, 6: 290–297, 1959.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pp. 2943–2952. PMLR, 2020a.

Utku Evci, Yani A Ioannou, Cem Keskin, and Yann Dauphin. Gradient flow in sparse neural networks and how lottery tickets win. *arXiv preprint arXiv:2010.03533*, 2020b.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020a.

Jonathan Frankle, David J Schwab, and Ari S Morcos. The early phase of neural network training. *arXiv preprint arXiv:2002.10365*, 2020b.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ig-VyQc-MLK.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *arXiv preprint arXiv:1810.09619*, 2018.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations*, 2015.

Babak Hassibi and David G Stork. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann, 1993.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1389–1397, 2017.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

Steven A Janowsky. Pruning versus clipping in neural networks. *Physical Review A*, 39(12):6600, 1989.

Siddhant Jayakumar, Razvan Pascanu, Jack Rae, Simon Osindero, and Erich Elsen. Top-kast: Top-k always sparse training. *Advances in Neural Information Processing Systems*, 33, 2020.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Jeremy Kepner and Ryan Robinett. Radix-net: Structured sparse matrices for deep neural networks. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 268–274. IEEE, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Aditya Kusupati, Vivek Ramanujan, Raghav Somani, Mitchell Wortsman, Prateek Jain, Sham Kakade, and Ali Farhadi. Soft threshold weight reparameterization for learnable sparsity. In *International Conference on Machine Learning*, pp. 5544–5555. PMLR, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pp. 598–605, 1990.

Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive sparsity for the magnitude-based pruning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=H6ATjJ0TKdf.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

Shiwei Liu, Decebal Constantin Mocanu, Amarsagar Reddy Ramapuram Matavalam, Yulong Pei, and Mykola Pechenizkiy. Sparse evolutionary deep learning with over one million artificial neurons on commodity hardware. *Neural Computing and Applications*, 2020a.

Shiwei Liu, Tim Van der Lee, Anil Yaman, Zahra Atashgahi, Davide Ferraro, Ghada Sokar, Mykola Pechenizkiy, and Decebal Constantin Mocanu. Topological insights into sparse neural networks. *arXiv preprint arXiv:2006.14085*, 2020b.

Shiwei Liu, Lu Yin, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Do we actually need dense over-parameterization? in-time over-parameterization in sparse training. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 6989–7000. PMLR, 2021.

Tianlin Liu and Friedemann Zenke. Finding trainable sparse networks through neural tangent transfer. In *International Conference on Machine Learning*, pp. 6336–6347. PMLR, 2020.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *International Conference on Learning Representations*, 2018.

Zelda Mariet and Suvrit Sra. Diversity networks: Neural network compression using determinantal point processes. *arXiv preprint arXiv:1511.05077*, 2015.

Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. *arXiv preprint arXiv:1909.12180*, 2019.

Decebal Constantin Mocanu, Elena Mocanu, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. A topological insight into restricted boltzmann machines. *Machine Learning*, 104(2): 243–270, Sep 2016. ISSN 1573-0565. doi: 10.1007/s10994-016-5570-z. URL https://doi.org/10.1007/s10994-016-5570-z.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *International Conference on Learning Representations*, 2016.

Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pp. 4646–4655. PMLR, 2019.

Michael C Mozer and Paul Smolensky. Using relevance to reduce network size automatically. *Connection Science*, 1(1):3–16, 1989.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Ameya Prabhu, Girish Varma, and Anoop Namboodiri. Deep expander networks: Efficient deep networks from graph theory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–35, 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. *arXiv preprint arXiv:2003.02389*, 2020.

Victor Sanh, Thomas Wolf, and Alexander M Rush. Movement pruning: Adaptive sparsity by fine-tuning. *arXiv preprint arXiv:2005.07683*, 2020.

Jingtong Su, Yihang Chen, Tianle Cai, Tianhao Wu, Ruiqi Gao, Liwei Wang, and Jason D Lee. Sanity-checking pruning methods: Random tickets can win the jackpot. *Advances in Neural Information Processing Systems. arXiv:2009.11094*, 2020.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. NETWORK COMPRESSION USING CORRELATION ANALYSIS OF LAYER RESPONSES, 2019. URL https://openreview.net/forum?id=rkl42iA5t7.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *arXiv preprint arXiv:2006.05467*, 2020.

Kale-ab Tessera, Sara Hooker, and Benjamin Rosman. Keep the gradients flowing: Using gradient flow to study sparse network optimization. *arXiv preprint arXiv:2102.01670*, 2021.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Stijn Verdenius, Maarten Stol, and Patrick Forré. Pruning via iterative ranking of sensitivity statistics. *arXiv preprint arXiv:2006.00896*, 2020.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgsACVKPH.

Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. *arXiv preprint arXiv:1906.03787*, 2019.

Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 111–120, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

A MEASUREMENTS AND METRICS

Formally, let's denote a (sparse) network $f : \mathcal{X} \to \mathcal{Y}$ trained on samples from distribution $\mathcal{D}$. $f(x, \theta)$ and $f(x, \theta \odot m)$ refers to the dense network and the pruned network, respectively. The test accuracy of the pruned subnetworks is typically reported as their accuracy on test queries drawn from $\mathcal{D}$, i.e., $\mathbb{P}_{(x,y)\sim\mathcal{D}}(f(x, \theta \odot m) = y)$. In addition to the test accuracy, we also evaluate random pruning from the perspective of natural corruption robustness, out-of-distribution performance, and uncertainty estimation.

Adversarial Robustness Despite of the remarkable ability solving classification problems, the predictions of deep neural networks can often be fooled by small adversarial perturbations (Szegedy et al., 2013; Papernot et al., 2016). Prior works have shown the possibility of finding the sweet point of sparsity and adversarial robustness (Guo et al., 2018; Ye et al., 2019). As the arguably most naive method of inducing sparsity, we are also interested in if training a randomly pruned subnetwork can improve the adversarial robustness of deep networks. We follow the classical method proposed in Goodfellow et al. (2014) and generate adversarial examples with Fast Gradient Sign Method (FGSM). Specifically, input data is perturbed with $\epsilon \text{sign}(\nabla_x \mathcal{L}(\theta, x, y))$, where ϵ refers to the perturbation strength, which is chosen as $\frac{8}{255}$ in our paper.

Out-of-distribution performance. The investigation of out-of-distribution (OoD) generalization is of importance for machine learning in both academic and industry fields. Since the i.i.d. assumption can hardly be satisfied, especially those high-risk scenarios such as healthcare, and military. We evaluate whether random pruning brings benefits to OoD. Following the classic routines (Augustin et al., 2020; Meinke & Hein, 2019), SVHN (Netzer et al., 2011), CIFAR-100, and CIFAR-10 with random Gaussian noise (Hein et al., 2019) are adopted for models trained on CIFAR-10; SVHN, CIFAR-10, and CIFAR-100 with random Gaussian noise are used for models trained on CIFAR-100.

Uncertainty estimation. In the security-critical scenarios, e.g., self-driving, the classifiers not only must be accurate but also should indicate when they are likely to be incorrect (Guo et al., 2017). To test effects of the induced sparsity on uncertainty estimation, we choose two widely-used metrics, expected calibration error (ECE) (Guo et al., 2017) and negative log likelihood (NLL) (Friedman et al., 2001).

B PREDICTIVE ACCURACY OF RESNET ON CIFAR-10 INCLUDING GRASP

To avoid multiple lines overlap with each other, we report the results of GraSP ratio on CIFAR-10 separately in this Appendix. It is surprising to find that random pruning with GraSP ratio (red lines) consistently have lower accuracy than ERK and SNIP, except for extremely high sparsites, i.e., 0.8 and 0.9.

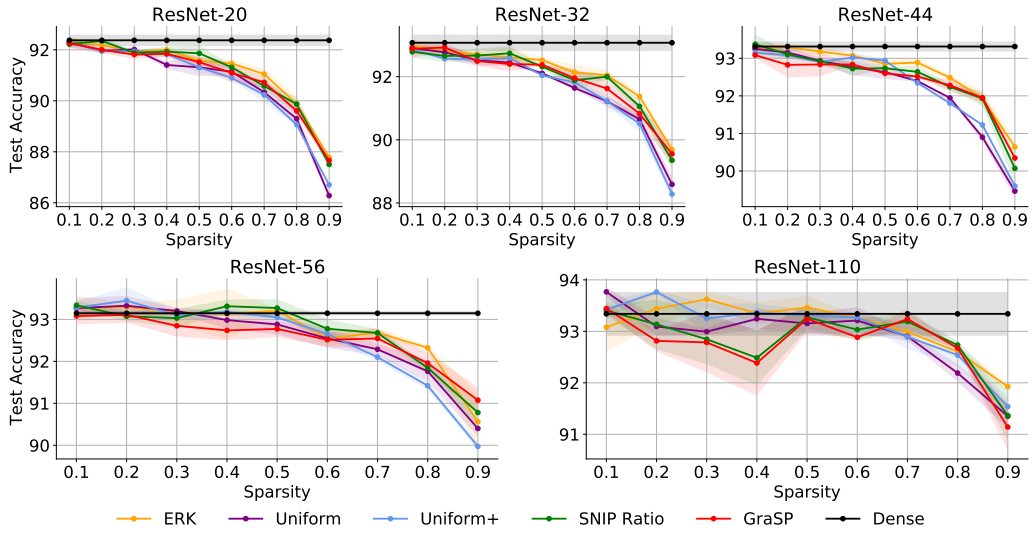

Figure 9: **From shallow to deep.** Test accuracy of training randomly pruned subnetworks from scratch with different depth on CIFAR-10. ResNet-A refers to a ResNet model with A layers in total.

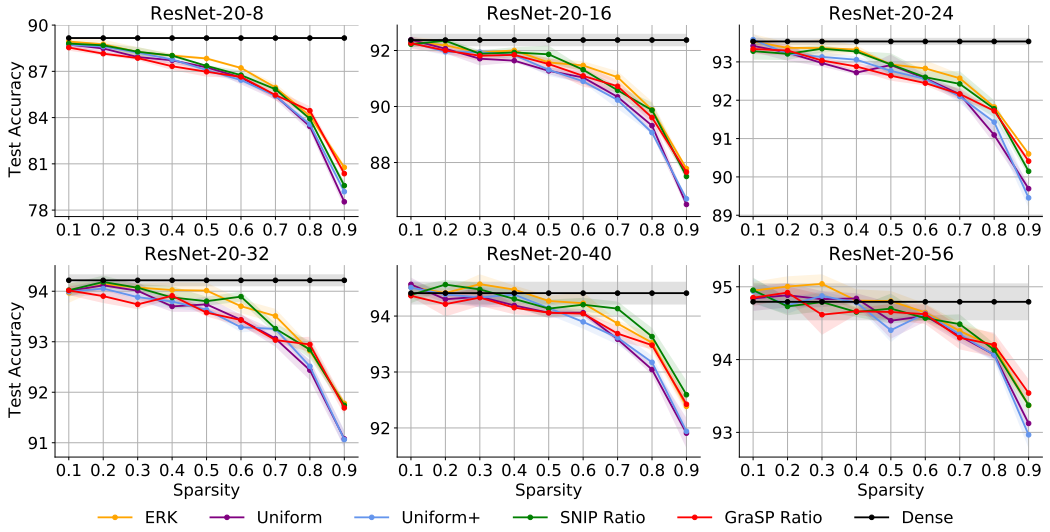

Figure 10: **From narrow to wide.** Test accuracy of training randomly pruned subnetworks from scratch with different width on CIFAR-10. ResNet-A-B refers to a ResNet model with A layers in total and B filters in the first convolutional layer.

C PREDICTIVE ACCURACY OF RESNET ON CIFAR-100

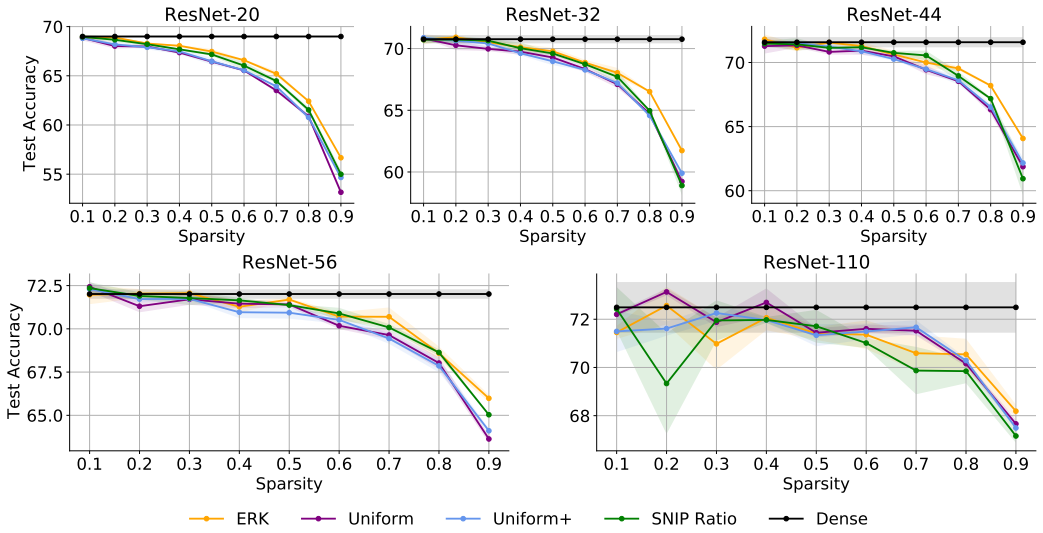

Figure 11: **From shallow to deep.** Test accuracy of training randomly pruned subnetworks from scratch with different depth on CIFAR-100. ResNet-A refers to a ResNet model with A layers in total.

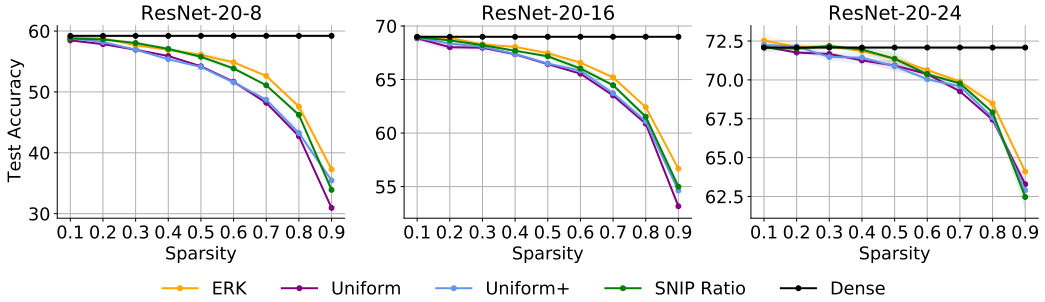

Figure 12: **From narrow to wide.** Test accuracy of training randomly pruned subnetworks from scratch with different width on CIFAR-100. ResNet-A-B refers to a ResNet model with A layers in total and B filters in the first convolutional layer.

D PREDICTIVE ACCURACY OF VGG ON CIFAR-10/100

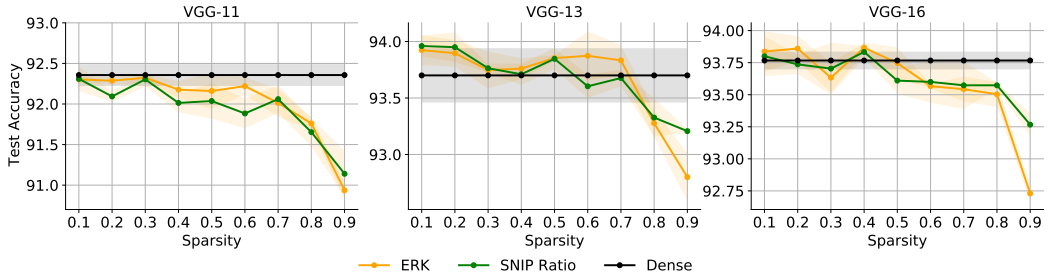

Figure 13: **From shallow to deep.** Test accuracy of training randomly pruned VGGs from scratch with different depth on CIFAR-10.

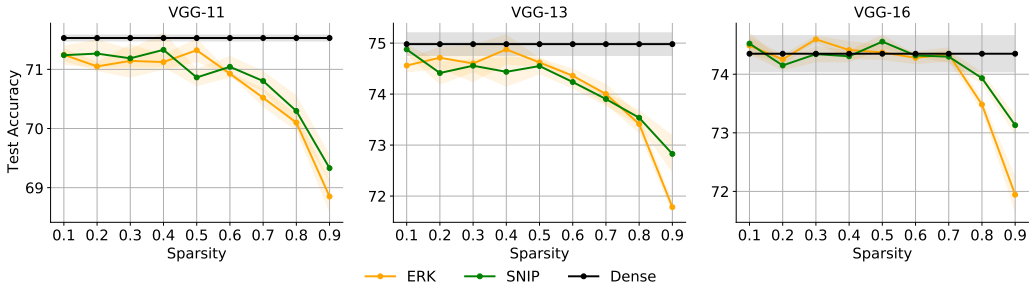

Figure 14: **From shallow to deep.** Test accuracy of training randomly pruned VGGs from scratch with different depth on CIFAR-100.

E SAME SCALE OF EXPERIMENTS ON IMAGENET

We provide a scaled-up overview of the accuracy and parameter count with various ResNet on ImageNet. We have expanded the experiments on ImageNet with various architectures from ResNet-18 to ResNet-101 and Wide ResNet-50. Due to the limited time and resources, we choose the SNIP and ERK+ sparsity ratios for Wide ResNet-50 and only SNIP ratios for the rest of the architectures. The sparsity of randomly pruned subnetworks is set as [0.7, 0.5, 0.3] from left to right. We run all this set of experiments with A100 GPUs whose accuracy is slightly different from the original submission (ran with V100).

The results are shared in Figure 15. Overall, we observe a very similar pattern as illustrated in the main paper. On smaller models like ResNet-18 and ResNet-34, random pruning at initialization can not find matching subnetworks. When the model size gets considerably larger (ResNet-101 and Wide ResNet-50), random pruning can match the dense models at 50% sparsity.

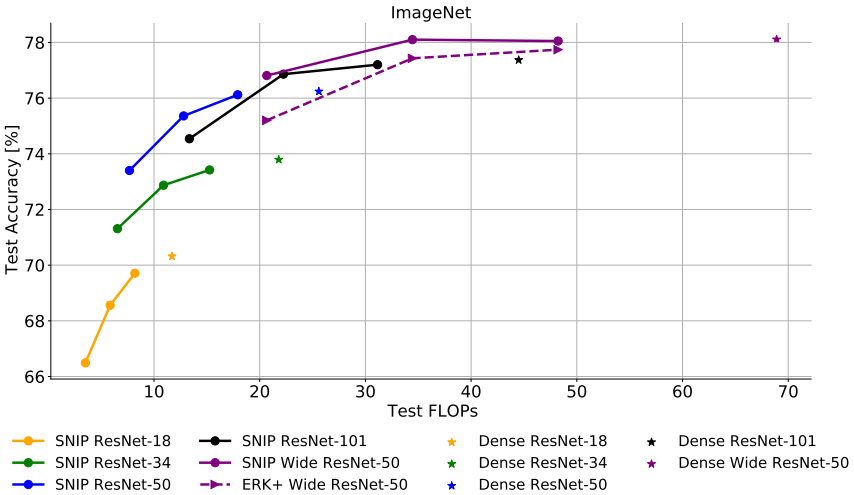

Figure 15: **Same Scale of Experiments on ImageNet**: Test accuracy and parameter count of various randomly pruned ResNets trained from scratch on ImageNet.

F TRADE-OFF BETWEEN ACCURACY AND FLOPS OF VARIOUS RESNET MODELS ON IMAGENET

We provide a scaled-up overview of the trade-off between accuracy and test FLOPs with various ResNet on ImageNet. The results are shared in Figure 16. We can see that the number of FLOPs required by the pruned Wide ResNet-50 is still larger than the one required by the dense ResNet-50. Yet, the corresponding performance gain is also notable (2%). More importantly, when the model size is large enough, the accuracy of the randomly pruned sparse networks quickly grows and matches the corresponding large dense network (not only the small-dense models). Random pruning at initialization gets larger efficiency (the difference between the test FLOPs of sparse models and dense models) when the model size increases.

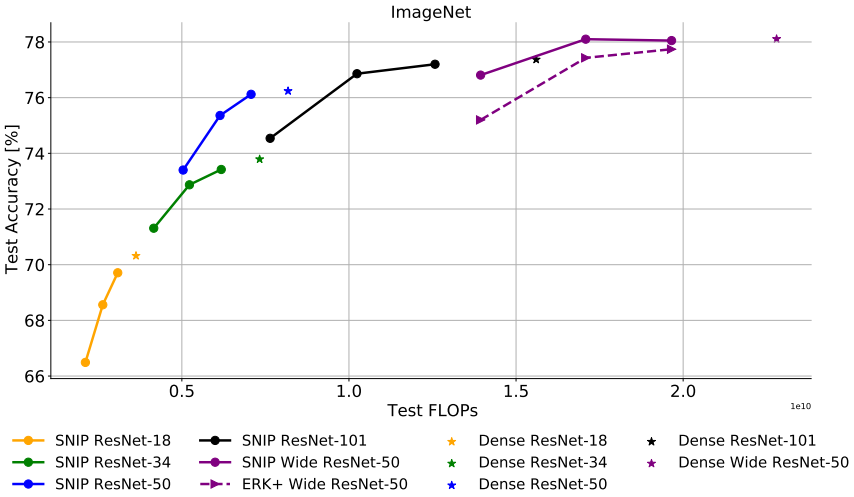

Figure 16: **Trade-off between Accuracy and FLOPs**: Test accuracy and FLOPs of various randomly pruned ResNets trained from scratch on ImageNet.

G BROADER EVALUATION OF RANDOM PRUNING ON IMAGENET

G.1 ADVERSARIAL ROBUSTNESS ON IMAGENET

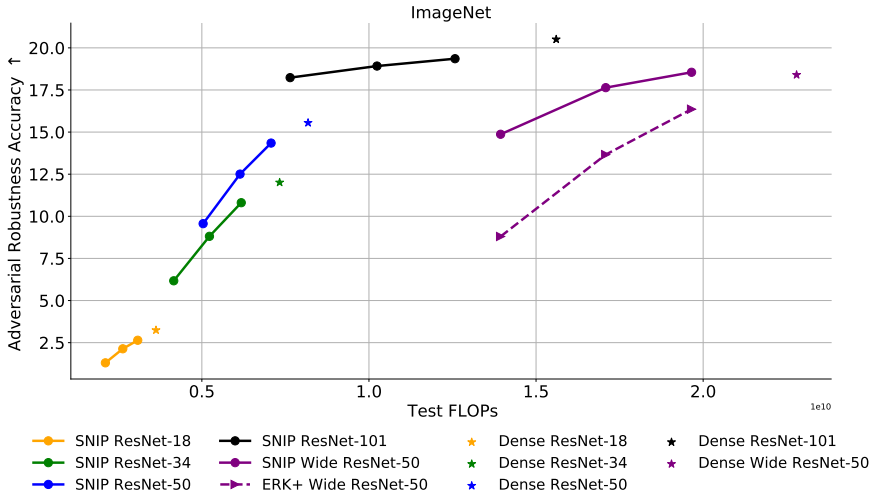

Figure 17: **Adversarial Robustness**: The adversarial robustness accuracy of various randomly pruned ResNets trained from scratch on ImageNet.

G.2 OUT-OF-DISTRIBUTION ON IMAGENET

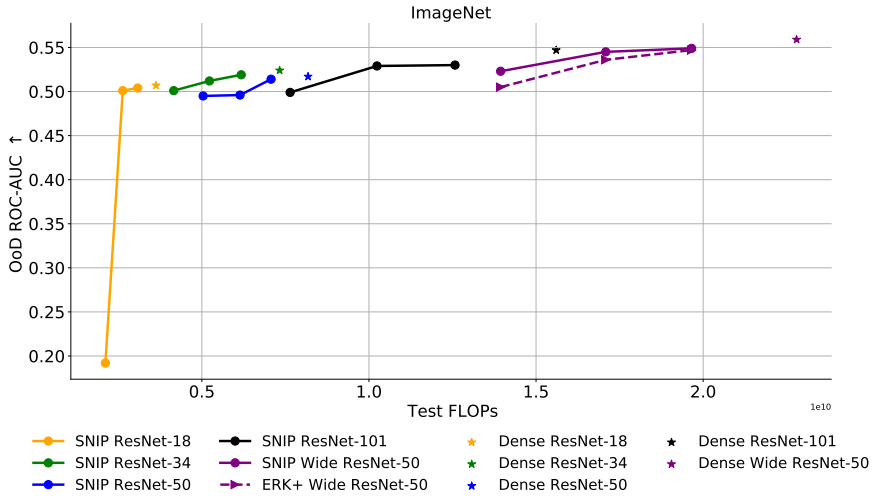

Figure 18: **Out-of-Distribution Performance**: The ROC-AUC of various randomly pruned ResNets trained from scratch on ImageNet-O.

G.3 UNCERTAINTY ESTIMATION (ECE) ON IMAGENET

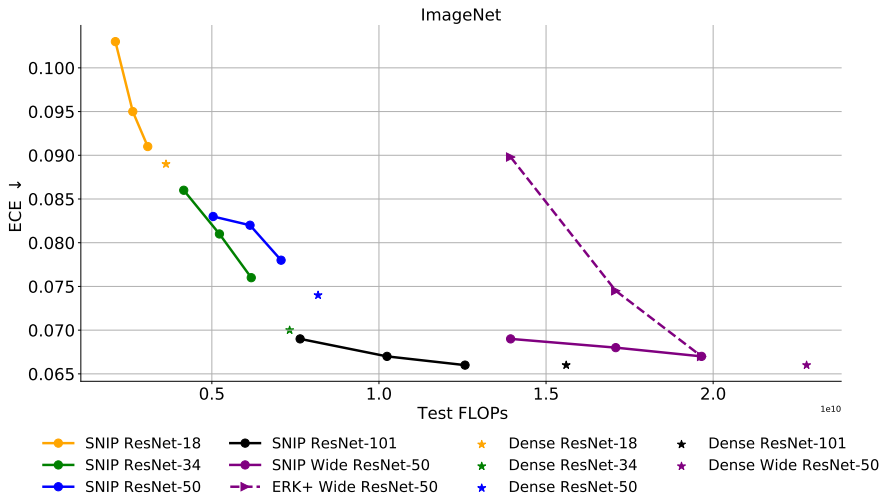

Figure 19: **Uncertainty Estimation (ECE)**: The ECE of various randomly pruned ResNets trained from scratch on ImageNet.

G.4 UNCERTAINTY ESTIMATION (NLL) ON IMAGENET

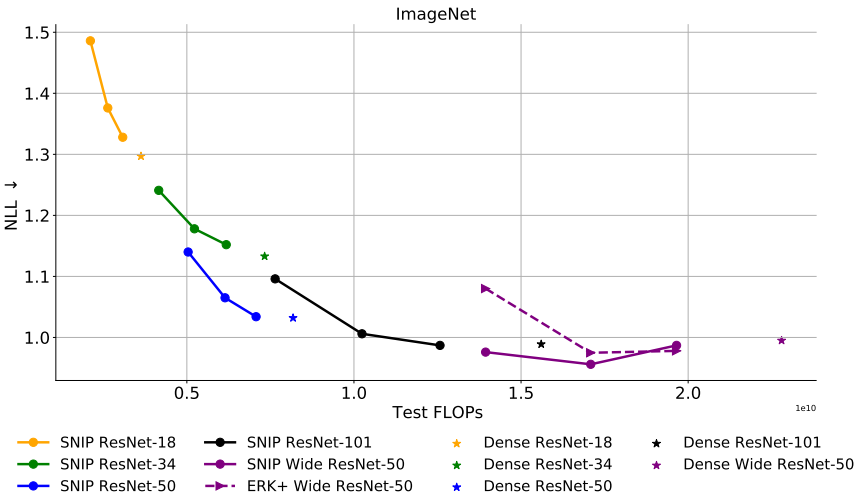

Figure 20: **Uncertainty Estimation (NLL)**: The NLL of various randomly pruned ResNets trained from scratch on ImageNet.

H BROADER EVALUATION OF RANDOM PRUNING ON CIFAR-10

H.1 OUT-OF-DISTRIBUTION PERFORMANCE (ROC-AUC) .

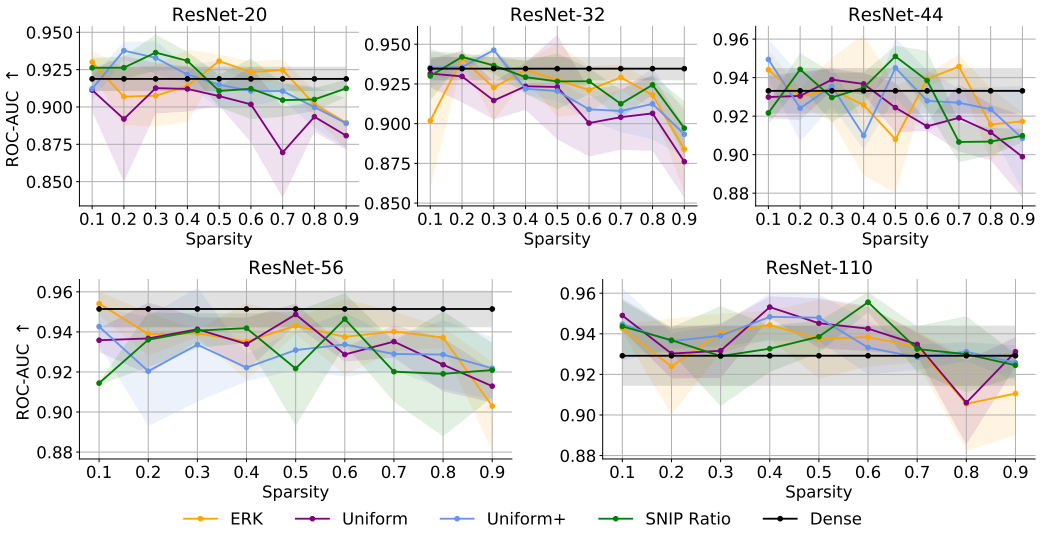

Figure 21: **Out-of-distribution performance (ROC-AUC).** The experiments are conducted with various models trained on CIFAR-10, tested on SVHN. Higher ROC-AUC refers to better OoD performance.

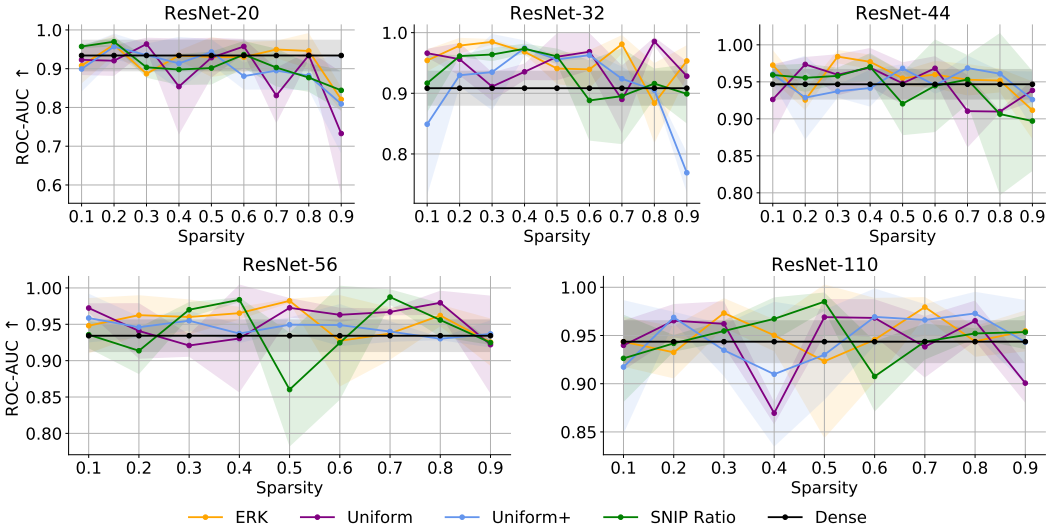

Figure 22: **Out-of-distribution performance (ROC-AUC).** The experiments are conducted with various models trained on CIFAR-10, tested on Gaussian noise. Higher ROC-AUC refers to better OoD performance.

H.2 UNCERTAINTY ESTIMATION (NLL)

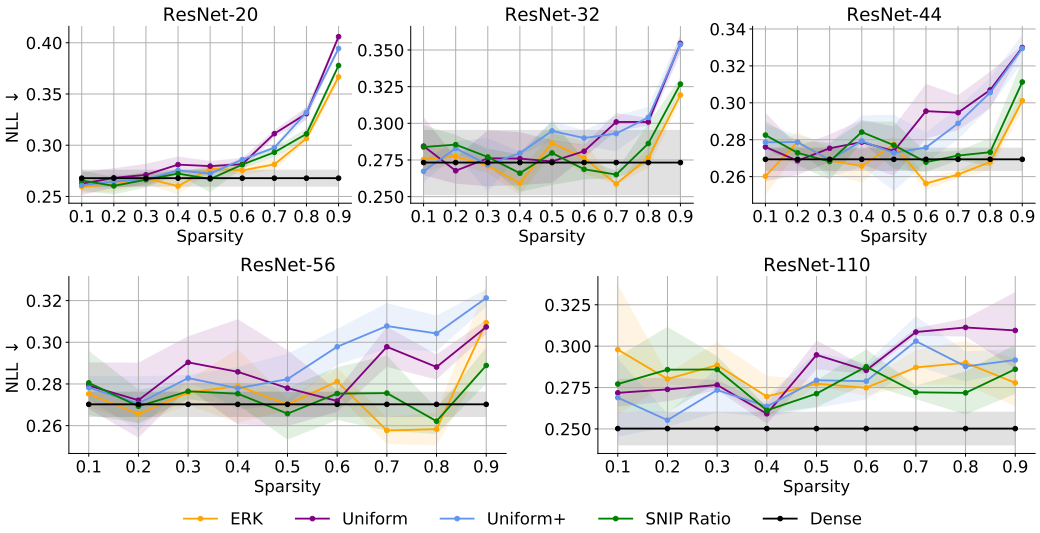

Figure 23: **Uncertainty estimation (NLL).** The experiments are conducted with various models on CIFAR-10. Lower NLL values represent better uncertainty estimation.