Sparse Double Descent: Where Network Pruning Aggravates Overfitting

Zheng He [1] **Zeke Xie** [2 3] **Quanzhi Zhu** [1] **Zengchang Qin** [1]

Abstract

People usually believe that network pruning not only reduces the computational cost of deep networks, but also prevents overfitting by decreasing model capacity. However, our work surprisingly discovers that network pruning sometimes even aggravates overfitting. We report an unexpected *sparse double descent* phenomenon that, as we increase model sparsity via network pruning, test performance first gets worse (due to overfitting), then gets better (due to relieved overfitting), and gets worse at last (due to forgetting useful information). While recent studies focused on the *deep double descent* with respect to model overparameterization, they failed to recognize that sparsity may also cause double descent. In this paper, we have three main contributions. First, we report the novel sparse double descent phenomenon through extensive experiments. Second, for this phenomenon, we propose a novel learning distance interpretation that the curve of ℓ_2 learning distance of sparse models (from initialized parameters to final parameters) may correlate with the sparse double descent curve well and reflect generalization better than minima flatness. Third, in the context of sparse double descent, a winning ticket in the lottery ticket hypothesis surprisingly may not always win.

1. Introduction

Deep neural networks (DNNs) have achieved great empirical success in recent years, which are usually trained with much more model parameters than training examples (LeCun et al., 2015). The extreme overparameterization gives DNNs excellent approximation as well as a prohibitively large model capacity (Cybenko, 1989; Funahashi, 1989; Hornik et al., 1989; Hornik, 1993). Overparameterized DNNs can even easily memorize entire random-labeled dataset (Zhang et al., 2017), which suggests that DNNs are kind of "good at" overfitting.

However, in practice, DNNs do not learn via pure memorization and often achieve higher generalization performance on many tasks than smaller models (Szegedy et al., 2015; Neyshabur et al., 2015; Arpit et al., 2017). Recent studies even reported an interesting deep double descent phenomenon (Belkin et al., 2019; Nakkiran et al., 2020; Loog et al., 2020; d'Ascoli et al., 2020; Yang et al., 2020) that, as the model capacity increases, test performance first gets better, then gets worse (due to classical overfitting), and gets better at last (due to relieved overfitting). Deep double descent motivated us to rethink the widely held viewpoint that network pruning reduces model capacity thus mitigates overfitting (LeCun et al., 1990; Hassibi & Stork, 1992; Molchanov et al., 2017; Hoefler et al., 2021).

As the extreme overparameterization of DNNs may cause deep double descent, may the sparsification of DNNs also cause a double descent phenomenon symmetric to the existing "deep double descent"? Our answer is affirmative. We call such phenomenon "sparse double descent".

Our main contributions are summarized as follows:

- To the best of our knowledge, our work is the first to report the sparse double descent phenomenon (see Figure 1). More specifically, we demonstrate that high model sparsities may significantly mitigate overfitting, while moderate model sparsities may lead to severer overfitting. And extreme model sparsities ($\rightarrow 100\%$) tend to lose all learned information.
- The ℓ_2 learning distance of models (from initialized parameters to final parameters) may be correlated with a double descent curve and reflects generalization better than minima flatness for sparse models.
- Contrary to the *lottery ticket hypothesis* Frankle & Carbin (2019), we find the retraining a sparse model from its original initialization may not win at all time. For example, in some cases, a randomly reinitialized pruned model could largely surpass model with the

[1] Intelligent Computing and Machine Learning Lab, School of ASEE, Beihang University, Beijing, China [2] The University of Tokyo [3] RIKEN Center for AIP. Correspondence to: Zheng He <zhenghe@buaa.edu.cn>, Zengchang Qin <zcqin@buaa.edu.cn>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

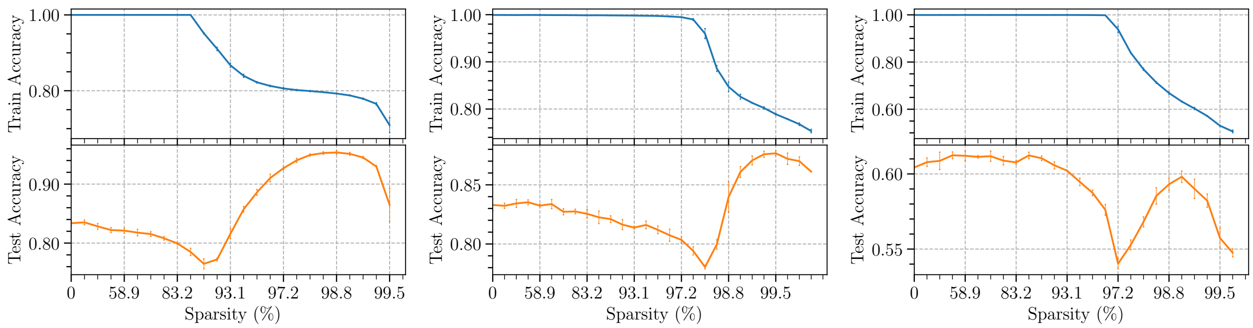


Figure 1. Sparse Double Descent of LeNet-300-100 on MNIST and ResNet-18 on CIFAR with 20% symmetric label noise. We evaluate how the train and test performance of networks depend on model sparsity. **Left**: MNIST. **Middle**: CIFAR-10. **Right**: CIFAR-100.

original initialization at certain sparsities.

This paper is organized as follows. In Section 2, we demonstrate empirical evidence for sparse double descent through extensive experiments. In Section 3, we analyze how learning distance matters to the sparse double descent compared with minima flatness. In Section 4, we include further empirical analysis and discuss the relationship between our results and some related work. In Section 5, we conclude with our main work.

2. Sparse Double Descent

In this section, we conducted extensive experiments to demonstrate sparse double descent with respect to model sparsity. We also identified four phases of model sparsity in empirical results. In our experiments, we grow model sparsity gradually via network pruning, and use the train and test accuracy to demonstrate how overfitting and generalization depend on model sparsity.

2.1. Overview

Although network pruning has been widely investigated at the target of storage and computational savings (Han et al., 2015; 2016; Liu et al., 2017; Molchanov et al., 2017; Li et al., 2017; Louizos et al., 2018; Frankle & Carbin, 2019; Liu et al., 2019), a broad consensus on how pruning will act on generalization has not been achieved yet (Hoefler et al., 2021).

Prior studies have demonstrated that networks learn simpler patterns first and are less prone to memorize noisy labels with limited capacity (Arpit et al., 2017; Li et al., 2020b). Following Occam's razor, network pruning that aims to reduce parameter counts could also be regarded as some kind of regularization on model capacity (LeCun et al., 1990; Hassibi & Stork, 1992). By restricting a subset of model parameters to a value of zero, pruning imposes sparsity constraints on neural networks and penalizes its redundant expressive power. Pruning is thus sometimes considered as a potential regularizer in recent works. (Molchanov et al., 2017; Ahmad & Scheinkman, 2019; Xia et al., 2021; Zhang et al., 2021)

Moreover, there are some other conjectures on how pruning might benefit generalization, e.g., pruning creates sparsified versions of data representation, which introduce noise and encourage flatness into neural networks (Han et al., 2017; Bartoldson et al., 2020). And flatness of minima is usually correlated with good generalization (Hochreiter & Schmidhuber, 1995; 1997; Hardt et al., 2016; Keskar et al., 2017; Zhu et al., 2019; Xie et al., 2020; 2022b). Given the discussions above, it is intuitive to suppose that pruning can enhance model performance and prevent overfitting.

However, our experimental results reveal that moderately sparse networks sometimes generalize even worse than their dense counterpart, meaning sparsity may aggravate overfitting under some circumstances. In this paper, we particularly evaluate overfitting and generalization in the presence of noisy labels. This setting is common in related work (Yang et al., 2020; Stephenson et al., 2021; Xie et al., 2021a;b), because it is helpful to understanding severe overfitting in real-world problems. As DNNs overfit easily and noisy labels exist pervasively in real-world datasets (Shankar et al., 2020; Northcutt et al., 2021a;b), learning with noisy labels has been a popular topic recently. Thus, we consider learning with noisy labels as an important problem setting to evaluate overfitting and generalization.

We find that in the presence of label noise, the double descent phenomenon occurs through model sparsification. Namely, when noisy labels are added to the training data, the sparse double descent can be observed robustly across different datasets, architectures, pruning settings and label noise types. The existence of sparse double descent raises doubts on conventional wisdom that pruning relieves overfitting, and offers new insights into understanding the relationship between sparsity and generalization.

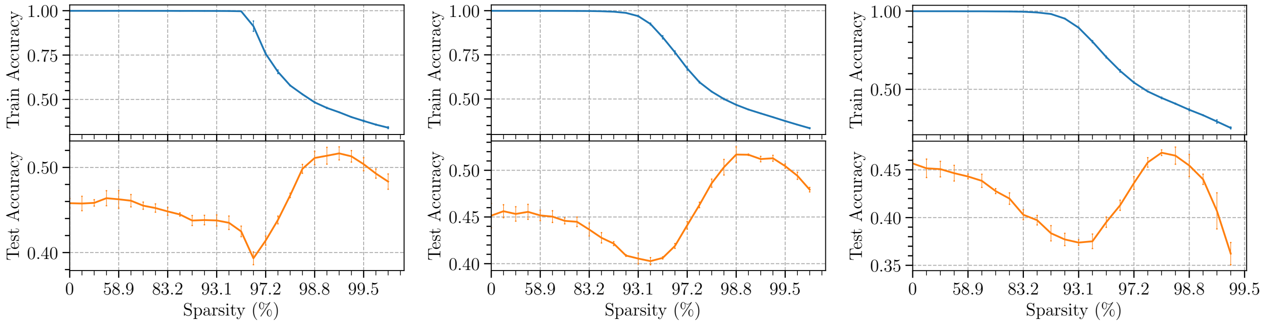


Figure 2. Sparse Double Descent of ResNet-18 on CIFAR-100 with 40% symmetric label noise, pruned using different strategies. We plot the train and test accuracy against sparsity. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

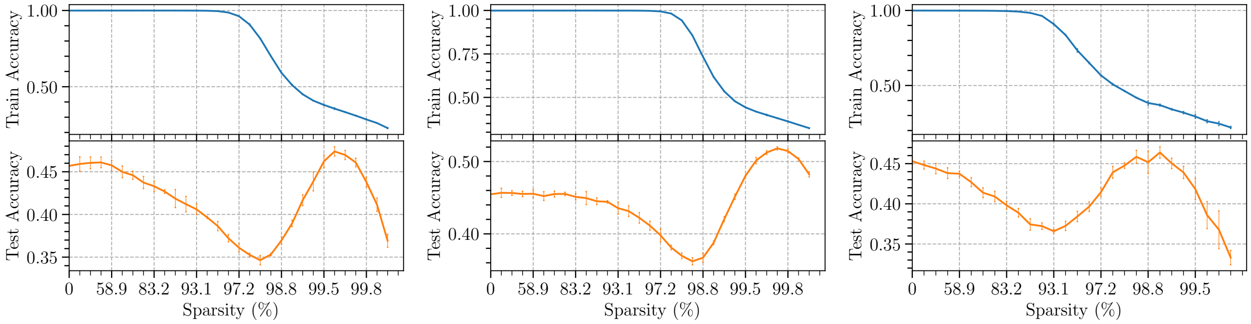


Figure 3. Sparse Double Descent of ResNet-18 on CIFAR-100 with 40% symmetric label noise, and different retraining methods.We plot the train and test accuracy against sparsity. **Left**: Finetuning. **Middle**: Learning rate rewinding. **Right**: Scratch retraining.

2.2. Experimental Setup

We describe the main experimental setup used throughout this paper. Particularly, we also vary several experimental choices, e.g., models and datasets, pruning strategies, retraining methods and label noise settings, to verify the generalizability of the sparse double descent phenomenon. Preliminaries, experimental details as well as more experimental results are given in the Appendix.

Models and datasets. We train a fully-connected LeNet-300-100 (LeCun et al., 1998) on MNIST (LeCun et al., 1998), a ResNet-18 (He et al., 2016) on CIFAR-10 or CIFAR-100 (Krizhevsky et al., 2009). We use the SGD optimizer with momentum 0.9 and adopt commonly used hyperparameters for training and pruning. We repeat experiments five (on MNIST) or three (on other datasets) times with different seeds and plot the mean and standard deviation. We also test a VGG-16 (Simonyan & Zisserman, 2015) on CIFAR datasets (see Figure 20 and 27). In the context of larger model and dataset, we train a ResNet-101 on Tiny ImageNet dataset[1] (see Figure 30), which is a reduced version of ImageNet (Deng et al., 2009).

Network pruning. Pruning is an effective technique to enhance the efficiency of deep networks with limited computational budget, by removing dispensable weights, filters or other structures from neural networks (LeCun et al., 1990; Han et al., 2015; Li et al., 2017; Liu et al., 2017). In this work, we do not chase state-of-the-art accuracy nor the computing resource efficiency; thus we simply remove each weights individually (i.e., unstructured pruning), for this method can adjust easily to different tasks and architectures.

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, we define a neural network classifier function as $f(\mathbf{w}; \mathcal{D})$, where $\mathbf{w} \in \mathbb{R}^d$ is the set of weights, and d is the total number of weights. As weight pruning removes parameters individually, we introduce a binary masks $\mathbf{m} \in \{0, 1\}^d$ as auxiliary to represent the remained weights $\mathbf{w} \odot \mathbf{m}$. The classifier function under sparsity constraints is denoted as $f(\mathbf{w} \odot \mathbf{m}; \mathcal{D})$, where $\odot$ is the element-wise product. The model sparsity of a pruned network is defined as $1 - \sum_{i=1}^{d} \mathbf{m}_i / d$.

Pruning strategies. We use three existing pruning heuristics summed up by Blalock et al. (2020), i.e., magnitude-based pruning, gradient-based pruning and random pruning. Magnitude-based pruning is one of the most commonly used baselines, and has been shown to achieve comparable performance to many complex techniques (Han et al., 2015;

[1] This dataset is from the Tiny ImageNet Challenge: https://tiny-imagenet.herokuapp.com/

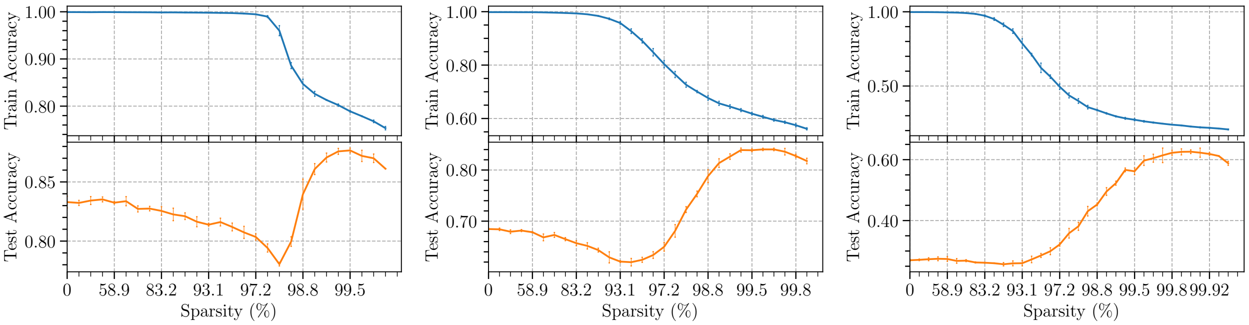


Figure 4. Sparse Double Descent of ResNet-18 on CIFAR-10 with various symmetric label noise rate. We evaluate how the train and test performance of networks depend on model sparsity. **Left**: $\epsilon = 20\%$. **Middle**: $\epsilon = 40\%$. **Right**: $\epsilon = 80\%$.

2016; Gale et al., 2019). Gradient-based pruning preserves training dynamics and provides possibility to prune a network early in training (Lee et al., 2019; 2020). And random pruning is often regraded as a naive method, setting the performance benchmark that any elaborately designed method should surpass (Frankle et al., 2021).

We prune weights in a network globally by comparing them across layers with the mentioned heuristics. For mainline experiments, we utilize the magnitude-based pruning if not otherwise noted. Weights with the lowest absolute magnitudes in a network will be removed after one pruning operation.

Retraining. A common approach to recover network performance after pruning is retraining, which means training the pruned networks for some extra epochs. Along side the sparse structures induced by different pruning strategies, re-training methods also affect network performance by determining which point on the optimization landscape to start training from, i.e., near initialization or close to the final weights; or which learning rate schedule to utilize. We mainly utilized the technique *lottery ticket rewinding* (LTR) proposed in *lottery ticket hypothesis* (Frankle & Carbin, 2019) to retrain a network from near initialization.

In addition to LTR, we also consider other existing retraining techniques like finetuning, learning rate rewinding, and scratch retraining. Interestingly, the sparse double descent phenomenon exists consistently despite different retraining settings (see Figure 3). We mainly demonstrate the results of LTR unless otherwise specified, and leave more experimental results in the Appendix. In all experiments, networks are pruned, retrained and pruned iteratively, and 20% of weights will be removed during each pruning.

Label noise settings. To verify the generalizability of our observations, we run experiments on three types of synthetic label noise, i.e., symmetric noise, asymmetric noise and pairflip noise, which are widely used in prior works (Ma et al., 2018; Li et al., 2020a; Xia et al., 2021). The noise rate ϵ is set to 20%, 40% and 80%. Most results and discussions are based on symmetric label noise except where otherwise provided.

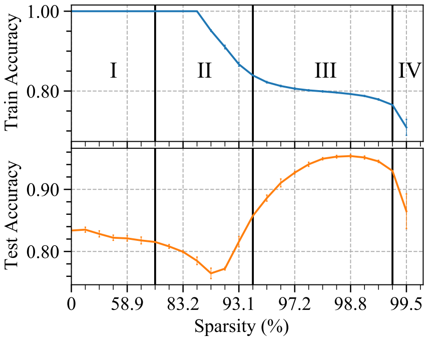


Figure 5. Illustration of four phases using the result of LeNet-300-100 on MNIST with 20% symmetric label noise. I: Light Phase. II: Critical Phase. III: Sweet Phase. IV: Collapsed Phase.

2.3. Effect of Model Sparsity on Overfitting

Empirical evidence of sparse double descent. From Figure 1, we empirically observe the sparse double descent for both ResNet-18 on CIFAR and LeNet-300-100 on MNIST in the presence of noisy labels. This suggests that sparse double descent may generally exist in various models and datasets, when overfitting is severe. Figures in section C.1 summarize the double descent behavior of sparse neural networks across different datasets, architectures and settings of pruning and label noise. In most cases, increasing sparsity of networks results in a first decrease then increase, and then decrease again in the test accuracy. Note that this increase is not caused by incomplete training of models, for models across all sparsities are able to converge to steady states (Figure 35, 36). Particularly, the right plot in Figure 2 shows that even random pruning may cause sparse double descent.

Four phases of model sparsity. Given the empirical ob-

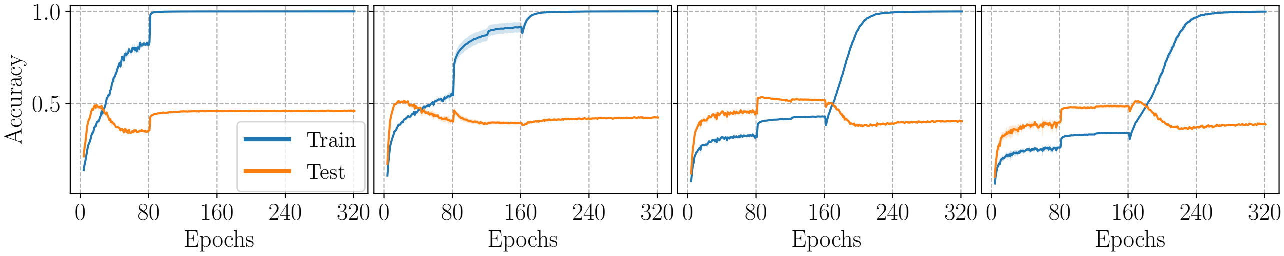


Figure 6. Accuracy as a function of epochs during sparse and re-dense training process. Results are from ResNet-18 networks trained on CIFAR-100 with $\epsilon = 40\%$. Pruned weights are recovered and trained from zero at epoch 160. The re-dense networks are optimized using learning rate of 0.001 for another 160 epoch. From left to the right, we display results of dense model, and models in Critical Phase, Sweet Phase and Collapsed Phase respectively.

servations, we identified four phases of model sparsity (see Figure 5). Different from previous work (Frankle et al., 2020; 2021), we define these four phases in terms of training accuracy and test accuracy.

First, *Light Phase* indicates low sparsities where the network is so overparameterized that pruned network can still reach similar accuracy to the dense model. Second, *Critical Phase* means an interval around the *interpolation threshold* (Nakkiran et al., 2020) where training accuracy is going to drop and test accuracy is about to first decrease then increase as sparsity grows. Third, *Sweet Phase* are high sparsities where test accuracy can significantly be boosted. Fourth, *Collapsed Phase* are those beyond, where both training accuracy and test accuracy drops significantly.

Here we illustrate how the sparse double descent is affected by data itself. As shown in Figure 4, increasing the fraction of label noise shifts the Critical Phase towards models with larger capacity, which is to say, lower sparsities. On the other hand, in order to combat the side effects brought by the existence of heavier labels noise, more parameters in the network need to be pruned. Thus, Sweet Phase generally moves to higher sparsities as noisy labels increase.

The most interesting discovery is the existence of Critical Phase, where overfitting becomes the most severe. To the best of our knowledge, we are the first to report the novel Critical Phase, where pruning, no matter reasonable heuristic-based pruning or random pruning, may decrease test accuracy while maintaining training accuracy, thus hurting generalization.

Discussion of related work. The observation that the Critical Phase may exist in random pruning contradicts with the conventional wisdom that random parameter perturbations can relieve overfitting (Harutyunyan et al., 2020; Xie et al., 2021a). On the other hand, a number of papers on sparsity (Molchanov et al., 2017; Lee et al., 2019; Hooker et al., 2019; Goel & Chen, 2021) only revealed that sparsity impairs memorization, but failed to uncover the unexpected loss of generalization performance as well. Liebenwein et al. (2021) studied the performance of pruned networks under distributional shifts, but did not reveal the nontrivial changing curve of test accuracy against sparsity.

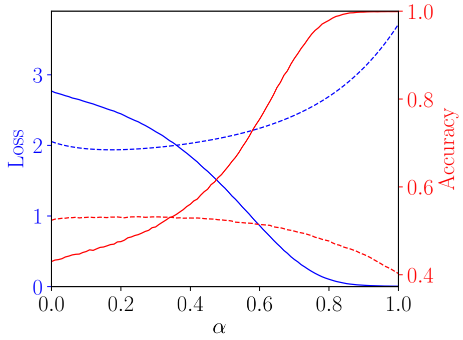


Figure 7. Linear interpolation plots for ResNet-18 on CIFAR-100 in Sweet Phase with $\epsilon = 40\%$. Note that $\alpha = 0$ corresponds to sparse solutions, while $\alpha = 1$ corresponds to the re-dense solutions (see methodology in Appendix A.5). The blue lines are loss curves and the red lines are accuracy curves; solid lines are about training data and dashed lines are about testing data.

A recent work Chang et al. (2021) studied double descent of pruned models with respect to the number of original model parameters. However, Chang et al. (2021) still focused on how model performance depends on overparameterization of original models rather than pruned models, while our work focused on the influence of sparsity on overfitting and generalization.

In summary, previous studies did not expect the existence of Critical Phase with respect to model sparsity, and thus, failed to discover sparse double descent.

3. Why Sparse Double Descent Occurs

In this section, we try to understand and analyze why sparse double descent occurs. We believe that the mechanism

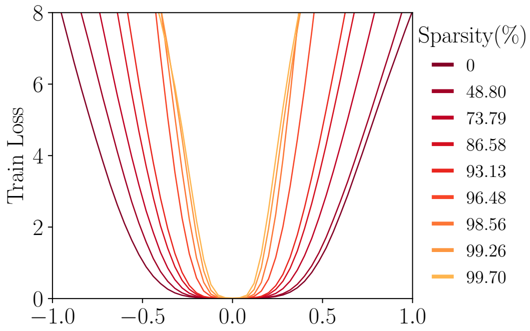


Figure 8. The 1-D loss visualization of minima found by re-dense training using filter normalization (Li et al., 2018). Result is of ResNet-18 on CIFAR-100 with $\epsilon = 40\%$. The sparsity of particular model is measured before the re-dense step. The sparser the original pruned network is, the sharper minima it might converge to after re-dense training.

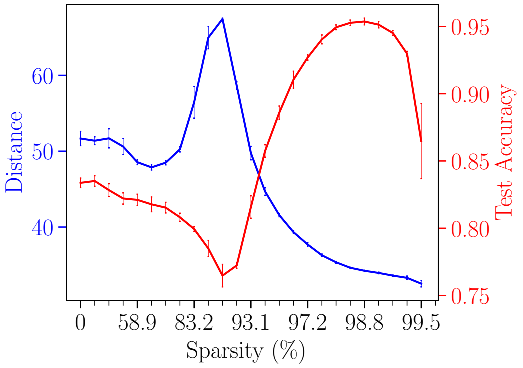


Figure 9. The curve of learning distance for LeNet-300-100 on MNIST with $\epsilon = 20\%$ may explain the double descent of test accuracy. As model sparsity increases, learning distance coincides the changes of test accuracy. The blue lines refer to ℓ_2 learning distance and the red lines are test accuracy.

behind sparse double descent can help to understand the conventional deep double descent with respect to model overparameterization from a new perspective.

3.1. Minima Flatness Cannot Explain Sparse Double Descent

Flatness is a widely used measure to capture the generalization behavior of neural networks (Jiang et al., 2020). Flatter minima are usually believed to imply robustness to perturbations in model parameters (Xie et al., 2021a; Foret et al., 2020), low complexity (Blier & Ollivier, 2018; Xie et al., 2022a), and better generalization (Keskar et al., 2017; Wu et al., 2017; Chaudhari et al., 2019). Previous works hypothesized that pruning could encourage the optimizer to move towards flatter minima that benefit generalization (Han et al., 2017; Bartoldson et al., 2020). May such minima flatness hypothesis explain sparse double descent?

We note that by removing a quantity of parameters, pruning restricts the movement of optimizer in low-dimension space, and drastically changes the loss landscape. Thus, measuring the minima flatness of pruned networks and comparing them across different sparsities possibly leads to unfair comparison. We are therefore motivated to seek indirect evidence that can estimate flatness with the same dimensions.

We apply the re-dense training approach to connect a sparse network with a dense one, which is introduced in Appendix A.4: after training a sparse network to reach convergence, we recover its pruned weights, and further retrain the whole network for certain epochs. If the optimizer reaches a flat basin of loss landscape during sparse training, we may suspect that a small learning rate in the re-dense training stage will continually attract optimizer around this basin. Then the final re-dense solutions will have comparable generalization performance to the sparse ones.

To demonstrate our points, we will focus on four model sparsities in the following analysis: 1) the zero sparsity (dense model), 2) the sparsity where the last test accuracy degrades the most (in Critical Phase), 3) the sparsity where the test accuracy reaches a peak (in Sweet Phase), and 4) the sparsity where both the train accuracy and test accuracy suffer (in Collapsed Phase).

Figure 6 surprisingly shows that sparse solutions are not stable in dense subspace. Once the sparsity constraints are removed, the model will escape from pruned solutions and overfit severely. The phenomenon that neural networks escape from highly sparse solutions during re-dense training, even with a small learning rate, raises doubts on the conjecture regarding minima flatness.

With linear interpolation of loss function, we find a monotonically decreasing path from the high-loss point to low-loss point in Figure 7. The existence of such path also demonstrates that these highly sparse solutions may no longer be minimizers in the loss landscape of dense model, thus allowing for the escape phenomenon during re-dense training process. Given this observation, we conjecture that sparsity may restrict the movement of optimizers, and trap them near initialization, which would be normally skipped when training dense networks.

Furthermore, the final solutions of re-dense training do not generalize well, and even have higher sharpness than the original dense models (see the 1-D visualization of re-dense solutions at various sparsities in Figure 8). The empirical results refute that highly sparse solutions stick around flat basins of minimizers.

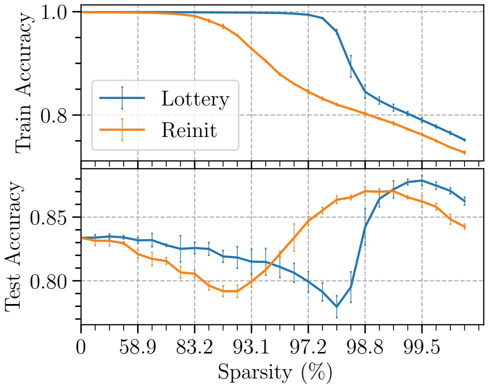


Figure 10. Performance of ResNet-18 on CIFAR-10 with $\epsilon = 20\%$ when retrained from either the original initialization (lottery tickets), or a random reinitialization. Reinitialization results sometimes surpass lottery results.

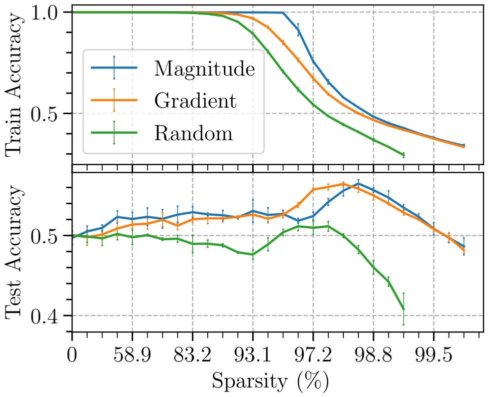


Figure 11. The train accuracy and best test accuracy as functions of sparsities. Models are ResNet-18 on CIFAR-100 with $\epsilon = 40\%$, pruned with three different pruning strategies.

3.2. The Learning Distance Hypothesis for Sparse Double Descent

Learning distance has been observed to be very related to generalization in deep learning (Neyshabur et al., 2019; Nagarajan & Kolter, 2019). Several studies suggested that neural networks need to move far from initialization to have large model capacity and overfit noisy labels, while avoid overfitting when staying close (Li et al., 2020b; Hu et al., 2020; Stephenson et al., 2021). And prior work Nagarajan & Kolter (2019) provided empirical and theoretical evidences that model capacity could be restricted by the ℓ_2 distance from initialization.

Given the failure of minima flatness hypothesis, we propose a novel learning distance hypothesis for sparse double descent that sparsity affects the learning distance, while learning distance correlates with model capacity and generalization ability.

For illustration, we denote weights at initialization by $\mathbf{w}_{init} \in \mathbb{R}^d$, and weights trained after the i_{th} pruning by $\mathbf{w}^i_{learned} \in \mathbb{R}^d$, where the pruned weights are regarded as zero weights. We define the learning distance D as the ℓ_2 distance from the original initialization to learned parameters, namely, $D(\mathbf{w}_{init}, \mathbf{w}^i_{learned}) = ||\mathbf{w}_{init} - \mathbf{w}^i_{learned}||_2$. Note that, the weights at initialization refer to the initialized parameters of the original dense model, instead of the sparse initialization of the pruned model.

In order to demonstrate the relationship between the learning distance and model performance, we plot both learning distance and test accuracy against sparsity in Figure 9. We surprisingly find that the changing curve of learning distance correlates with test accuracy. Our experimental results suggest that staying closer to initialization coincides with better robustness: in Critical Phase, sparse solutions are located farther from initialization, and present an inferior performance; while at high sparsities in Sweet Phase, sparse minimizers stay closer and manifest robustness (although in Collapsed Phase, too few parameters remain trainable that learning distance becomes less informative). Furthermore, the turning point of test accuracy from decreasing to increasing is usually consistent with a peak in learning distance. This suggests a relatively strong correlation between learning distance and sparse double descent.

While a number of generalization measures have been studied in deep learning, previous studies rarely touched how these generalization measures including learning distance may reflect double descent in deep learning. Our finding may shed light on theoretically understanding sparse double descent from the perspective of learning distance.

4. Empirical Analysis and Discussion

In this section, we present more empirical results about spare double descent. We discuss their meaning and some related work.

Lottery tickets may not win at all time. The *lottery ticket hypothesis* (Frankle & Carbin, 2019) supposes that a large network contains a subnetwork, which could be trained effectively together with the original initialization, while performs far worse if reinitialized. However, against the well-known lottery ticket hypothesis, we identify that reinitialized sparse models could outperform lottery ticket models in some circumstances (see Figure 10, 33 and 34). For the reinitialized models, the sparse double descent still holds, but with the Critical Phase and Sweet Phase both moved to left. This way, reinitialized models could beat lottery ticket models at the same sparsity but different phases. The results show that models with the same sparse structure but

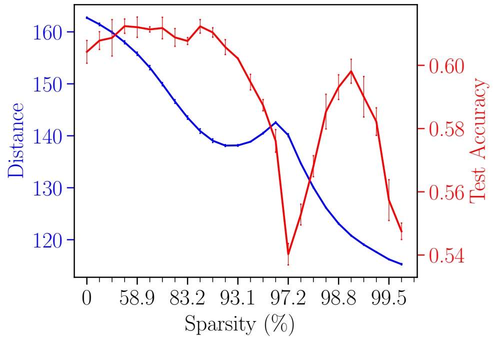


Figure 12. The ℓ_2 learning distance and test accuracy as functions of sparsity for CIFAR-100 with $\epsilon = 20\%$. The blue lines are ℓ_2 learning distance curves and the red lines are accuracy curves.

different initialization behave distinctively. Our finding suggests that a winning lottery ticket may not always win in the presence of sparse double descent.

The best test accuracy benefits from pruning. Although the test accuracy at the last epoch shows double descent with respect to pruning, we find that the best test accuracy across all epochs in training may not possess such well-defined trend. Instead, the best accuracy might increase in general from Light Phase till early Sweet Phase before finally drops down, with only a slight drop in Critical Phase (Figure 11). So why the best test accuracy sometimes does not show a double descent trend? As deep double descent actually also depends on the number of training epochs (Nakkiran et al., 2020), we conjecture that training epochs of DNNs also plays a key role in model capacity in the context of sparse double descent.

Does learning distance tell the whole story? Results in Figure 9, 12 and Appendix C.7 show that when sparsity is higher than critical (interpolation threshold) sparsity, learning distance of sparse networks declines continuously as pruning, and test accuracy climbs up to a peak before finally goes down. This observation supports our proposed hypothesis to some extent, and is consistent with prior theoretical studies that small distance from initialization may relieve overfitting (Nagarajan & Kolter, 2019; Li et al., 2020b). However, in Figure 12, the test accuracy of Critical Phase model is inferior to that of dense model, while the critical learning distance is not higher to the dense one (which differs from the MNIST results). It is natural to wonder why the learning distance curves of CIFAR do not present a peak in Critical Phase as distinct as those of MNIST?

One possible reason might be the inherent property of rectified neural networks: *scale invariance*. As we utilize ReLU as non-linearities activation, the network remains unchanged if we multiply the weights in one layer by t, and the next layer by $1/t$. In this way, the network's behavior remains unchanged but its distance from original parameters increases. Furthermore, this invariance is even more prominent when batch normalization is used, where the output of each layer will be re-scaled (Dinh et al., 2017; Li et al., 2018). In this case, the influence made by parameter removal during pruning might be offset by reparametrization effect of batch normalization, as such, changing the behavior of learning distance curve.

We suggest the learning distance as a potentially helpful generalization measure for sparse models, because this hypothesis correlates with the sparse double descent results to a certain extent. Proving it theoretically and rigorously for sparse neural networks remains as an open question. We will leave it for future work.

Possible explanations for the imperception of sparse double descent without label noise. We'd like to further discuss why the double descent phenomenon is rarely exhibited in existing pruning literature as well as our experiments under zero label noise (see Figure 31). Several possible explanations might account for its imperception:

(1) The double descent phenomenon itself is not stable (Ba et al., 2020; Nakkiran et al., 2020; Yang et al., 2020). To observe it in modern network architectures, we sometimes need to inject label noise (Nakkiran et al., 2020; Yang et al., 2020). Moreover, even though a "plateau" or a small peak in test error around the interpolation point of dense models is reported on the noiseless CIFAR dataset (Nakkiran et al., 2020), it will still be too hard to distinguish from the irregular fluctuations of test accuracy in pruning cases (Frankle & Carbin, 2019; Liu et al., 2019; Renda et al., 2020).

(2) The Critical Phase often occur within quite thin sparsity ranges, and the Sweet Phase relies on relatively high sparsities (generally greater than 90% under label noise). Many studies might not report enough points in the accuracy-sparsity tradeoff curve (Molchanov et al., 2017; Liu et al., 2017; Le & Hua, 2021);

(3) The regularization of pruning may come from other sources except for parameter reduction, like noise injection in model parameters(Bartoldson et al., 2020), which might offset the performance loss brought by pruning when training without label noise.

Nevertheless, under label noise settings, we can amplify the impact of reduced capacity on model performance brought about by sparsity, and reconcile the conventional understanding and the modern practice of network pruning.

Relation with theory of sparse learning. So far, the relationship between sparsity, learning dynamics and generalization remains as open question and has received growing attention from researchers. Emerging studies from the perspective of loss landscape provide enlightening insight into understanding the behaviors of sparse regimes. Evci et al. (2020) revealed that the existence of bad solutions in *sparse subspace* (namely, the sparsity pattern found by pruning), and illustrate the difficulty of escaping from bad solutions to good ones. And Lin et al. (2022) provide theoretical justification that sparsity can deteriorate the loss landscape by creating spurious local minima or spurious valleys. Our work is motivated by these findings, and what's more, moves a step further by empirically demonstrating that the reshaping effect on loss landscape at high sparsities by network pruning is actually beneficial in the presence of label noise.

Connection to robust learning with label noise. While our focus has been on the characteristics of sparse neural networks under noisy labels, there are other research hot-spots concerning label-noise learning, e.g., designing state-of-the-art robust training algorithms (Han et al., 2018; Jiang et al., 2018; Li et al., 2019; 2020a). Among these methods, we find CDR proposed by Xia et al. (2021) particularly related regarding the way to hinder memorization. Using a similar criterion to gradient-based pruning, they identify non-critical parameters and penalize them during optimization. By deactivating redundant parameters, memorization of noisy labels is hindered, and test performance before early stopping is enhanced. While our results reveal that, with a large proportion of parameters being removed permanently, performance after early stopping could also be boosted greatly.

5. Conclusion

In this paper, we reassess some common beliefs concerning the generalization properties of sparse networks, and illustrate the inapplicability of these viewpoints. Instead, we report an unexpected sparse double descent phenomenon. And our proposed learning distance hypothesis correlates with previous theoretical studies, and accounts for this phenomenon to a certain extent. We provide some insight into the optimization dynamics and generalization ability of sparse regimes, which we hope will guide progress towards further understanding for the theory of both sparse learning and deep learning.

References

Ahmad, S. and Scheinkman, L. How can we be so dense? the benefits of using highly sparse representations. *arXiv preprint arXiv:1903.11257*, 2019.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242. PMLR, 2017.

Ba, J., Erdogdu, M. A., Suzuki, T., Wu, D., and Zhang, T. Generalization of two-layer neural networks: An asymptotic viewpoint. In *8th International Conference on Learning Representations, ICLR*, 2020.

Bartoldson, B., Morcos, A. S., Barbu, A., and Erlebacher, G. The generalization-stability tradeoff in neural network pruning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20852–20864, 2020.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1903070116.

Blalock, D., Ortiz, J. J. G., Frankle, J., and Guttag, J. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

Blier, L. and Ollivier, Y. The description length of deep learning models. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 2220–2230, 2018.

Chang, X., Li, Y., Oymak, S., and Thrampoulidis, C. Provable benefits of overparameterization in model compression: From double descent to pruning neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6974–6983, 2021.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

d'Ascoli, S., Refinetti, M., Biroli, G., and Krzakala, F. Double trouble in double descent: Bias and variance (s) in the lazy regime. In *International Conference on Machine Learning*, pp. 2280–2290. PMLR, 2020.

Evci, U., Pedregosa, F., Gomez, A., and Elsen, E. The difficulty of training sparse neural networks, 2020.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR*, 2019.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Pruning neural networks at initialization: Why are we missing the mark? In *9th International Conference on Learning Representations, ICLR*, 2021.

Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3): 183–192, 1989.

Gale, T., Elsen, E., and Hooker, S. The state of sparsity in deep neural networks. *CoRR*, abs/1902.09574, 2019.

Goel, P. and Chen, L. On the robustness of monte carlo dropout trained with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2219–2228, 2021.

Goodfellow, I. J. and Vinyals, O. Qualitatively characterizing neural network optimization problems. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I. W., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*, 28, 2015.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations, ICLR*, 2016.

Han, S., Pool, J., Narang, S., Mao, H., Gong, E., Tang, S., Elsen, E., Vajda, P., Paluri, M., Tran, J., Catanzaro, B., and Dally, W. J. DSD: dense-sparse-dense training for deep neural networks. In *5th International Conference on Learning Representations, ICLR*, 2017.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234, 2016.

Harutyunyan, H., Reing, K., Steeg, G. V., and Galstyan, A. Improving generalization by controlling label-noise information in neural network weights. In *International Conference on Machine Learning*, 2020.

Hassibi, B. and Stork, D. G. Second order derivatives for network pruning: optimal brain surgeon. In *Proceedings of the 5th International Conference on Neural Information Processing Systems*, pp. 164–171, 1992.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hochreiter, S. and Schmidhuber, J. Simplifying neural nets by discovering flat minima. In *Advances in neural information processing systems*, pp. 529–536, 1995.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., and Peste, A. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *arXiv preprint arXiv:2102.00554*, 2021.

Hooker, S., Courville, A., Clark, G., Dauphin, Y., and Frome, A. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Hornik, K. Some new results on neural network approximation. *Neural networks*, 6(8):1069–1072, 1993.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Hu, W., Li, Z., and Yu, D. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *8th International Conference on Learning Representations, ICLR*, 2020.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *8th International Conference on Learning Representations, ICLR*, 2020.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR*, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Le, D. H. and Hua, B. Network pruning that matters: A case study on retraining variants. In *9th International Conference on Learning Representations, ICLR*, 2021.

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

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436, 2015.

Lee, N., Ajanthan, T., and Torr, P. H. S. Snip: single-shot network pruning based on connection sensitivity. In *7th International Conference on Learning Representations, ICLR*, 2019.

Lee, N., Ajanthan, T., Gould, S., and Torr, P. H. S. A signal propagation perspective for pruning neural networks at initialization. In *8th International Conference on Learning Representations, ICLR*, 2020.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. In *5th International Conference on Learning Representations, ICLR*, 2017.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pp. 6391–6401, 2018.

Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5051–5059, 2019.

Li, J., Socher, R., and Hoi, S. C. H. Dividemix: Learning with noisy labels as semi-supervised learning. In *8th International Conference on Learning Representations, ICLR*, 2020a.

Li, M., Soltanolkotabi, M., and Oymak, S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pp. 4313–4324. PMLR, 2020b.

Liebenwein, L., Baykal, C., Carter, B., Gifford, D., and Rus, D. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *Proceedings of Machine Learning and Systems*, 3:93–138, 2021.

Lin, D., Sun, R., and Zhang, Z. On the landscape of one-hidden-layer sparse networks and beyond. *Artificial Intelligence*, pp. 103739, 2022.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. In *7th International Conference on Learning Representations, ICLR*, 2019.

Loog, M., Viering, T., Mey, A., Krijthe, J. H., and Tax, D. M. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S., Xia, S., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pp. 3355–3364. PMLR, 2018.

Molchanov, D., Ashukha, A., and Vetrov, D. P. Variational dropout sparsifies deep neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2498–2507. PMLR, 2017.

Nagarajan, V. and Kolter, J. Z. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. In *8th International Conference on Learning Representations, ICLR*, 2020.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *ICLR (Workshop)*, 2015.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. The role of over-parametrization in generalization of neural networks. In *7th International Conference on Learning Representations, ICLR*, 2019.

Northcutt, C., Jiang, L., and Chuang, I. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021a.

Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks. *CoRR*, abs/2103.14749, 2021b.

Renda, A., Frankle, J., and Carbin, M. Comparing rewinding and fine-tuning in neural network pruning. In *8th International Conference on Learning Representations, ICLR*, 2020.

Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., and Schmidt, L. Evaluating machine accuracy on ImageNet. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8634–8644. PMLR, 13–18 Jul 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Stephenson, C., Padhy, S., Ganesh, A., Hui, Y., Tang, H., and Chung, S. On the geometry of generalization and memorization in deep neural networks. In *9th International Conference on Learning Representations, ICLR*, 2021.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Wu, L., Zhu, Z., et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., and Chang, Y. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2021.

Xie, Z., Sato, I., and Sugiyama, M. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2020.

Xie, Z., He, F., Fu, S., Sato, I., Tao, D., and Sugiyama, M. Artificial neural variability for deep learning: On overfitting, noise memorization, and catastrophic forgetting. *Neural computation*, 33(8):2163–2192, 2021a.

Xie, Z., Yuan, L., Zhu, Z., and Sugiyama, M. Positive-negative momentum: Manipulating stochastic gradient noise to improve generalization. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11448–11458. PMLR, 18–24 Jul 2021b.

Xie, Z., Tang, Q.-Y., Cai, Y., Sun, M., and Li, P. On the power-law spectrum in deep learning: A bridge to protein science. *arXiv preprint arXiv:2201.13011*, 2022a.

Xie, Z., Wang, X., Zhang, H., Sato, I., and Sugiyama, M. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *International Conference on Machine Learning*, 2022b.

Yang, Z., Yu, Y., You, C., Steinhardt, J., and Ma, Y. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pp. 10767–10777. PMLR, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR*, 2017.

Zhang, D., Ahuja, K., Xu, Y., Wang, Y., and Courville, A. C. Can subnetwork structure be the key to out-of-distribution generalization? In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12356–12367. PMLR, 2021.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7654–7663. PMLR, 2019.

A. Methodology

Here, we introduce the methods used in this paper in detail.

A.1. Pruning Strategies

We used three network pruning strategies in the paper. Unless particularly specified, results and discussions are based on the magnitude-based pruning.

- *Magnitude-based pruning*: prunes the weights with the lowest absolute magnitudes $|\mathbf{w}|$.
- *Gradient-based pruning*: prunes the weights with the lowest absolute values of magnitude multiplies gradient $|\frac{\partial L}{\partial \mathbf{w}} \odot \mathbf{w}|$, with L be the loss function evaluated on a random batch of inputs.
- *Random pruning*: issues each weight with a random score sampled independently from the uniform distribution $\mathcal{U}(0, 1)$, and prunes the weights with the lowest scores.

A.2. Retraining Techniques

The typical retraining based pruning procedure consists of three stages (Liu et al., 2019): 1) train a large, dense neural network to completion, 2) prune structures of the trained network according to certain heuristic, 3) retrain the network for T epochs to mitigate accuracy loss. In this paper, we run experiments using the following four retraining methods to verify the robustness of sparse double descent, and implementation details like rewinding iteration and learning rate of finetuning are listed in B:

- Lottery ticket rewinding: rewind the parameters and the learning rate of pruned network to the iteration t at an early training stage, and subsequently retrain the models from there (Frankle & Carbin, 2019) for $T - t$ epochs.
- Finetuning: retrains the pruned model from the final values of parameters, for T epochs (Han et al., 2015). Usually, finetuning will use a fixed small learning rate, i.e., the last learning rate of the original training schedule (Li et al., 2017; Liu et al., 2019).
- Learning rate rewinding: retrains the pruned model for $T - t$ epochs from the final parameters, but reuse the learning rate schedule from the iteration t at the early training phase (Renda et al., 2020). Learning rate rewinding is kind of a hybrid between lottery ticket rewinding and finetuning.
- Scratch retraining: rewind the learning rate schedule of pruned model to the beginning, but use a random reinitialization of parameters, and retrain the model for T epochs. This method is often used as baseline to make comparison with lottery ticket rewinding method. In our paper, during the pruning and scratch training iteration, we use a different initialization each time when we retrain.

A.3. Label Noise Types

Here, we introduce the different types of synthetic label noise used in our paper. Following Xia et al. (2021), we manually generate symmetric, asymmetric and pairflip label noise, and the details are listed below. Most results and discussions are based on symmetric label noise unless otherwise specified.

- Symmetric noise: generated by randomly permuted the labels for a fraction ϵ of the training data.
- Asymmetric noise: generated by flipping ϵ of labels within a set of similar classes. In our work, for MNIST, flip labels $2 \to 7, 3 \to 8, 5 \leftrightarrow 6$. For CIFAR-10, BIRD $\to$ AIRPLANE, DEER $\to$ HORSE, TRUCK $\to$ AUTOMOBILE, CAT $\leftrightarrow$ DOG. And for CIFAR-100, all 100 classes will be divided into 20 super-classes, and then labels in every class will be flipped to the next class with the same super-class.
- Pairflip noise: generated by flipping labels in each class to its adjacent class for ϵ of training data.

A.4. Re-dense Training

Pruning induces sparsity constraints into the objective function optimization problem, which move the optimization to a lower-dimension space. To empirically investigate the impact of sparsity constraints, we present studies which allow pruned weights to return to the model, which is motivated by DSD training (Han et al., 2017) and noted as re-dense training in this paper.

The re-dense training follows the training process of a pruned network after t epochs. During re-dense training, we recover pruned weights in the network, initialized them to zero, and retrain them together with the unpruned weights at the last epoch, for another t epochs with a fixed learning rate. We set the learning rate in re-dense training equal to the last learning rate of sparse training. Other learning hyperparameters (batch size, momentum, weight decay, etc.) are kept the same as original training process.

A.5. 1-D Linear Interpolation.

Visualizing the loss landscape can provide an empirical characterization of the geometry of neural network minimizers (e.g., their sharpness/flatness, or the structures of surrounding parameter space). We present linear interpolation plots of the training loss function along a line segment $\mathbf{w}$ between sparse solutions $\mathbf{w}_s$ and re-dense solutions $\mathbf{w}_r$, using the strategy proposed by Goodfellow & Vinyals (2015). We define $\mathbf{w}(\alpha) = (1-\alpha)\mathbf{w}_s + \alpha\mathbf{w}_r$ for $\alpha \in [0, 1]$ with increment of 0.01. And we compute the loss and accuracy of model with parameters $\mathbf{w}(\alpha)$ respectively. If there exists a monotonically decreasing loss objective from sparse solutions to re-dense solutions, we may conjecture that sparsity obstructs the optimization process with less trainable parameters.

A.6. 1-D Loss Visualization

We also plot 1-D loss function over a center minimizer using filter-wise normalized directions, to visualize the loss curvature and make comparisons between different minimizers. More explanation about this technique can be found in Li et al. (2018).

B. Experimental details

Here, we present the implementation details used in this paper.

We adopt standard implementations of LeNet-300-100 from OpenLTH[2]. LeNet-300-100 is a fully-connected network with 300 units in the first layer and 100 units in the second hidden layer, and ReLU activations.

For ResNet-18 network, we utilize a modified version of PyTorch model. To adapt ResNet-18 for CIFAR-10 and CIFAR-100, the first convolutional layer is equipped with filter of size 3×3 and the max-pooling layer that follows has been eliminated. CIFAR-10 and CIFAR-100 are augmented with per-channel normalization, randomly horizontal flipping, and randomly shifting by up to four pixels in any direction. And for VGG-16 network, we follow the settings from OpenLTH.

The ResNet-101 network for Tiny ImageNet is modified in the same way as the ResNet-18 network. Training instances in Tiny ImageNet are augmented with per-channel normalization, randomly cropping and resizing to a size of 32×32, and randomly horizontal flipping.

In pruning experiments, for LeNet-300-100, we consider all weights from linear layers except for the last layer as prunable parameters; for ResNets, all weights from convolutional and linear layers are set as prunable; while for VGG-16, we prune convolutional weights. We do not prune biases nor the batch normalization parameters. For convolutional and linear layers, the weights are initialized with Kaiming normal strategy and biases are initialized to be zero. Note that, all these settings are adopted from prior work Frankle & Carbin (2019) and never particularly tuned for the sake of fairness.

We run all the MNIST and CIFAR experiments on single GPU and Tiny ImageNet experiments on four GPUs with CUDA 10.1. The training hyperparameters used in our experiments is given as follows. Our code is available in https://github.com/hezheug/sparse-double-descent.

Network	Dataset	Epochs	Batch	Opt.	Mom.	LR	LR Drop	Drop Factor	Weight Decay	Rewind Iter	LR(finetune)	LR(re-dense)
LeNet-300-100	MNIST	200	128	SGD	—	0.1	—	—	—	0	0.1	0.1
ResNet-18	CIFAR-10	160	128	SGD	0.9	0.1	80, 120	0.1	1e-4	1000	0.001	0.001
ResNet-18	CIFAR-100	160	128	SGD	0.9	0.1	80, 120	0.1	1e-4	1000	0.001	0.001
VGG-16	CIFAR-10	160	128	SGD	0.9	0.1	80, 120	0.1	1e-4	2000	0.001	0.001
VGG-16	CIFAR-100	160	128	SGD	0.9	0.1	80, 120	0.1	1e-4	2000	0.001	0.001
ResNet-101	Tiny ImageNet	200	512	SGD	0.9	0.2	100, 150	0.1	1e-4	1000	—	—

[2] https://github.com/facebookresearch/open_lth

C. Additional experiment results and discussion

Here, we will present additional results that are not included in the main body for page limit.

C.1. Sparse double descent phenomenon

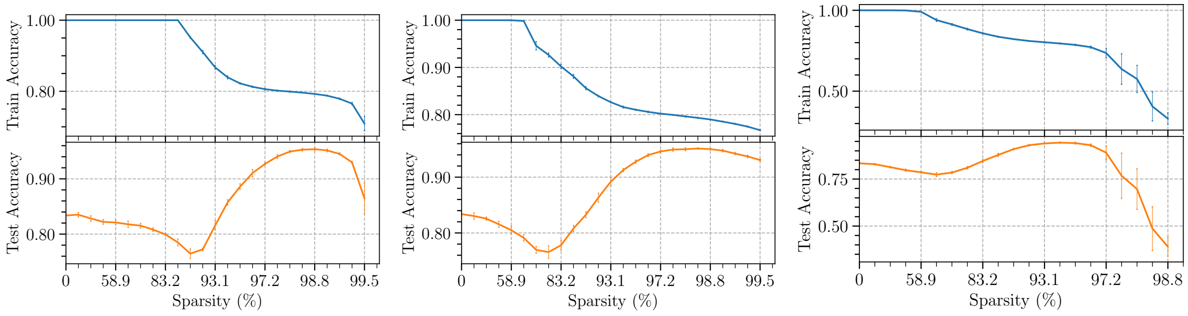


Figure 13. Sparse Double descend phenomenon with different pruning strategies for LeNet-300-100 on MNIST with symmetric label noise, $\epsilon = 20\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

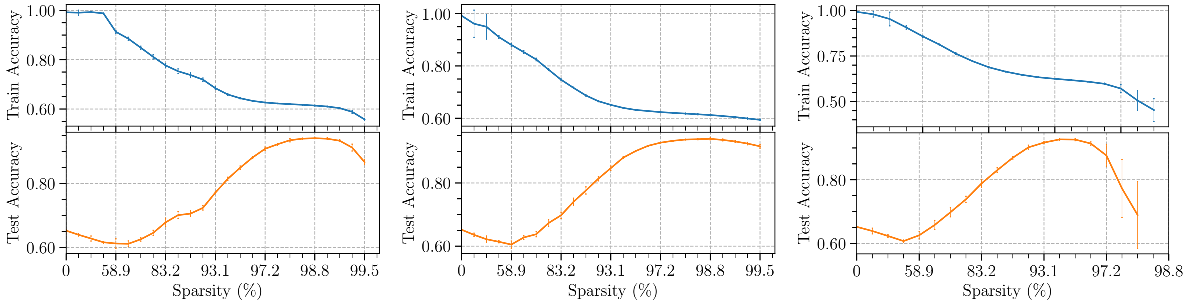


Figure 14. Sparse Double descend phenomenon with different pruning strategies for LeNet-300-100 on MNIST with symmetric label noise, $\epsilon = 40\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

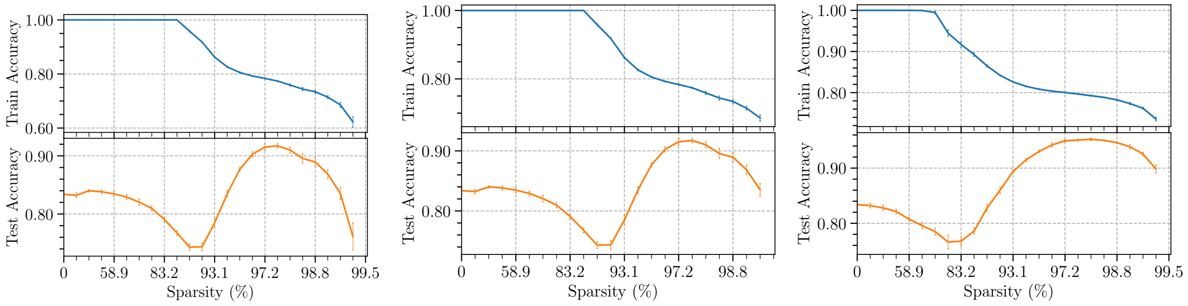


Figure 15. Sparse Double descend phenomenon with different retraining methods for LeNet-300-100 on MNIST with symmetric label noise, $\epsilon = 20\%$. **Left**: Finetuning. **Middle**: Learning Rate Rewinding. **Right**: Scratch retraining.

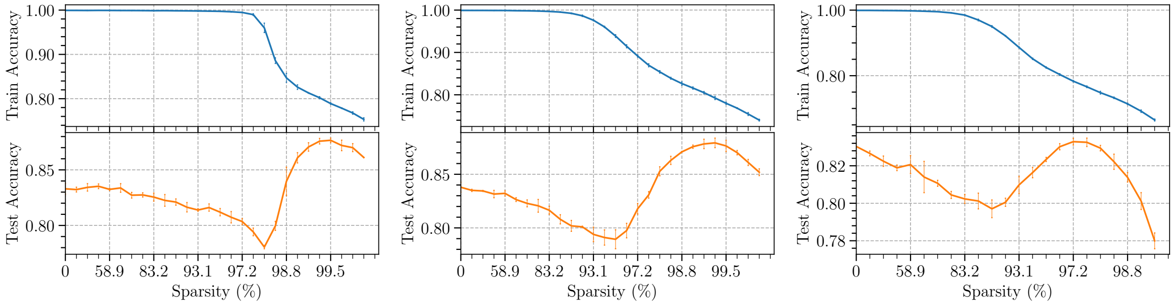


Figure 16. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-10 with symmetric label noise, $\epsilon = 20\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

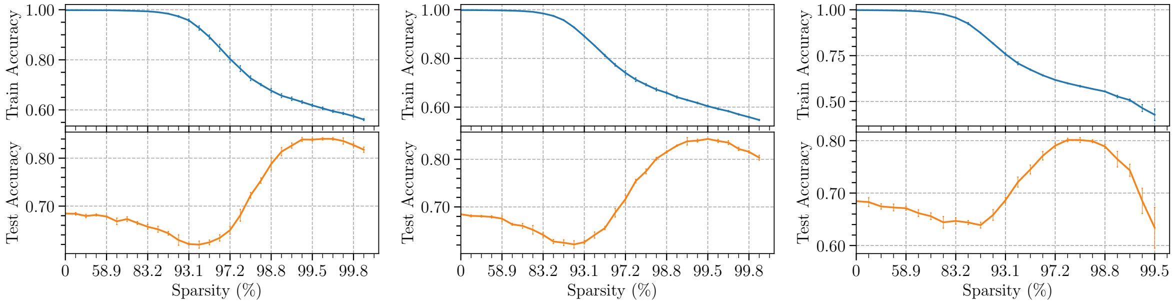


Figure 17. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-10 with symmetric label noise, $\epsilon = 40\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

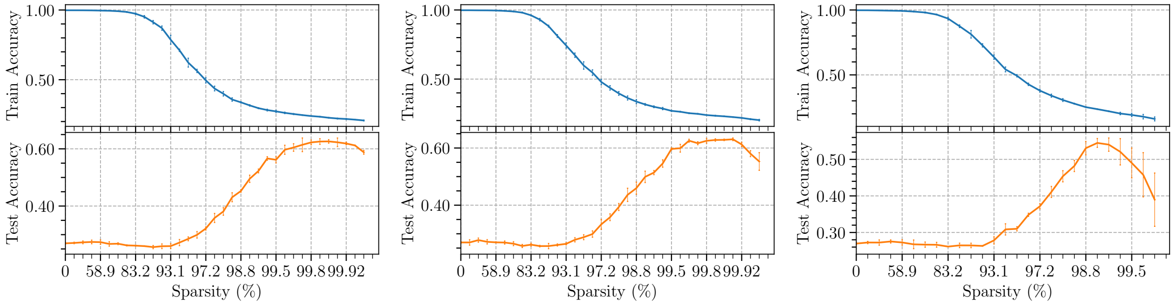


Figure 18. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-10 with symmetric label noise, $\epsilon = 80\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

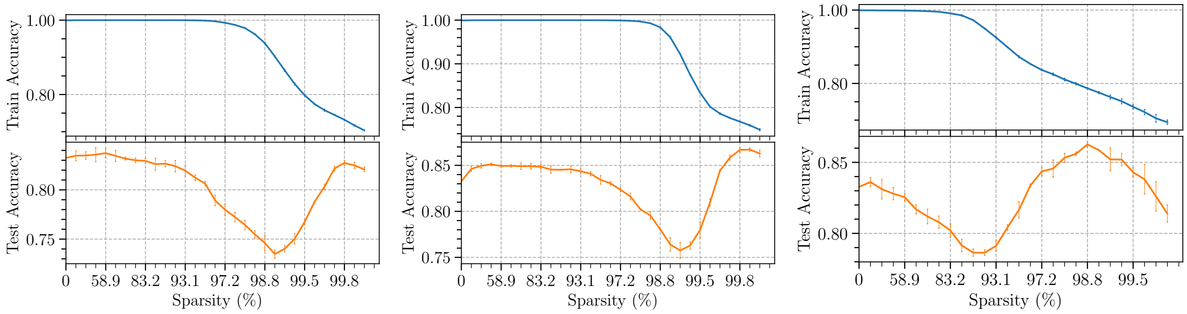


Figure 19. Sparse Double descend phenomenon with different retraining methods for ResNet-18 on CIFFAR-10 with symmetric label noise, $\epsilon = 20\%$. **Left**: Finetuning. **Middle**: Learning Rate Rewinding. **Right**: Scratch retraining.

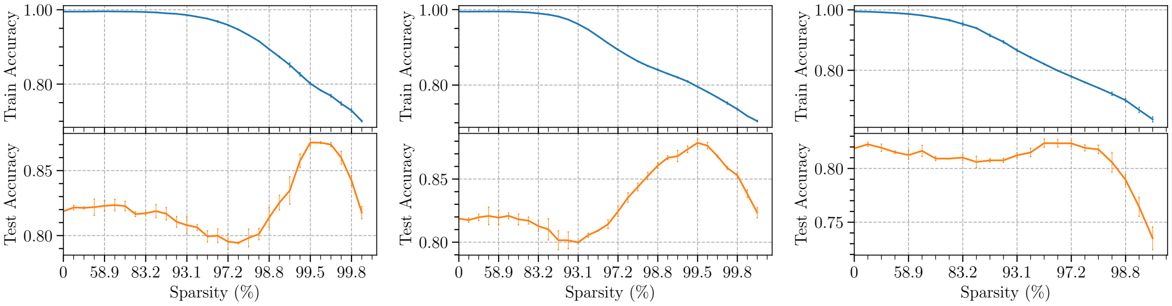


Figure 20. Sparse Double descend phenomenon with different pruning strategies for VGG-16 on CIFAR-10 with symmetric label noise, $\epsilon = 20\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

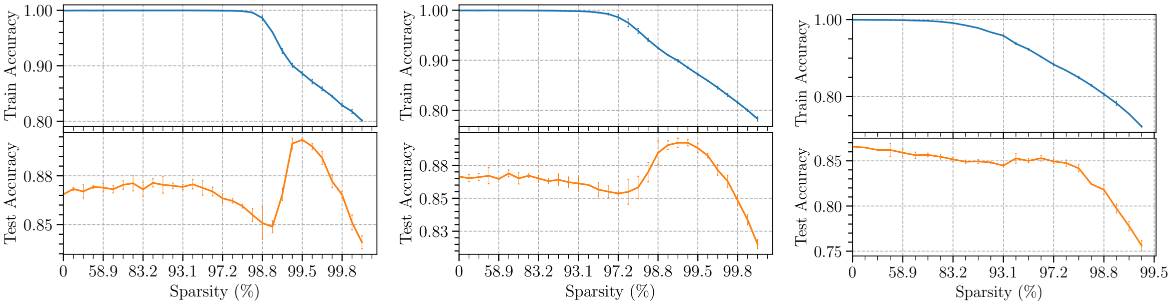


Figure 21. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-10 with asymmetric label noise, $\epsilon = 20\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

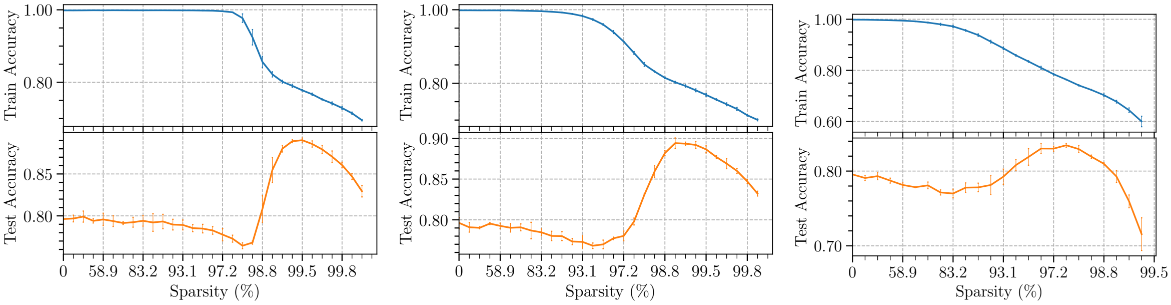


Figure 22. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-10 with pairflip label noise, $\epsilon = 20\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

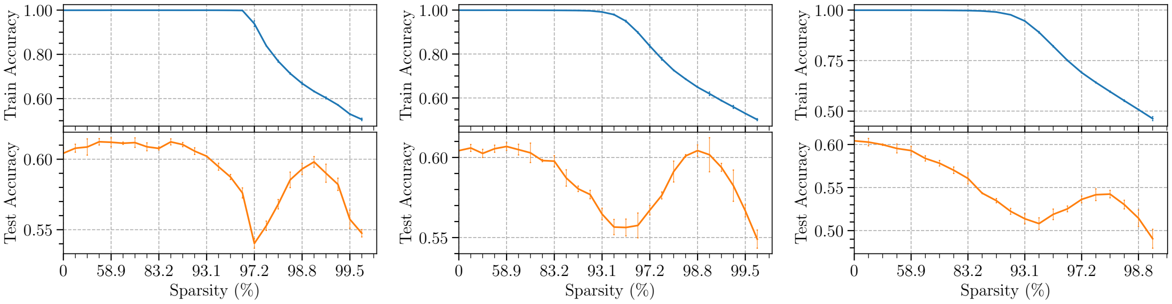


Figure 23. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-100 with symmetric label noise, $\epsilon = 20\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

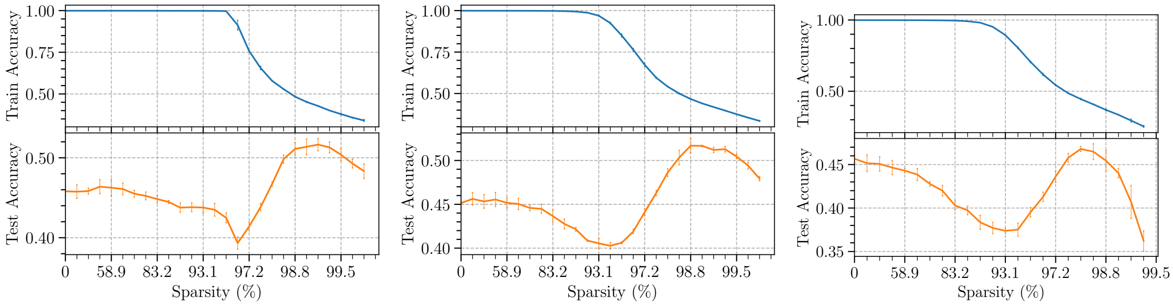


Figure 24. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-100 with symmetric label noise, $\epsilon = 40\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

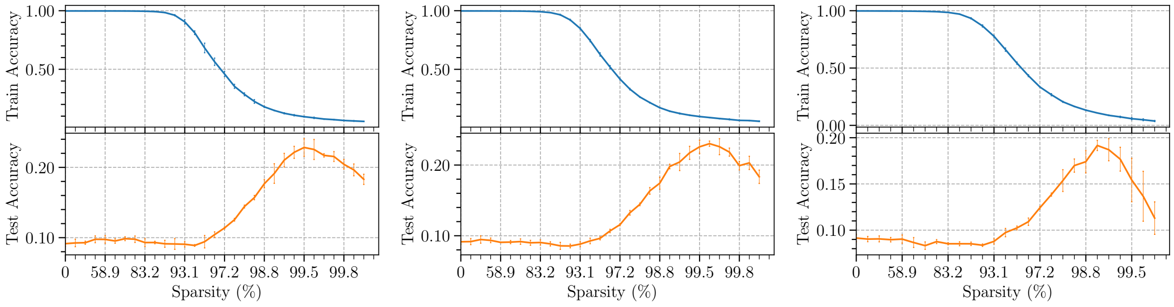


Figure 25. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-100 with symmetric label noise, $\epsilon = 80\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

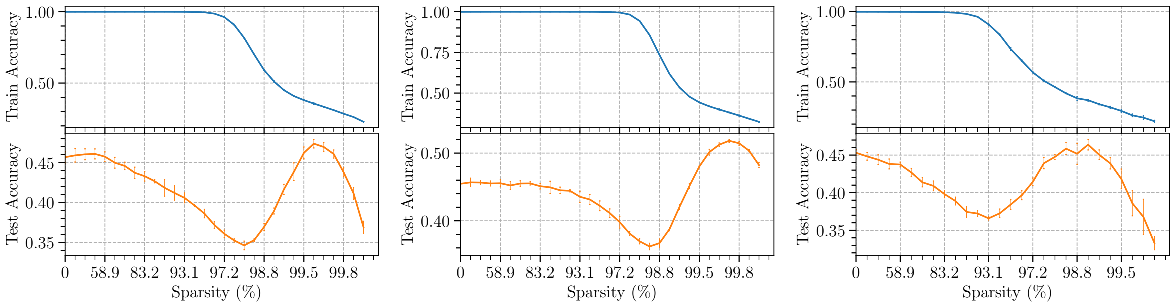


Figure 26. Sparse Double descend phenomenon with different retraining methods for ResNet-18 on CIFFAR-100 with symmetric label noise, $\epsilon = 40\%$. **Left**: Finetuning. **Middle**: Learning Rate Rewinding. **Right**: Scratch retraining.

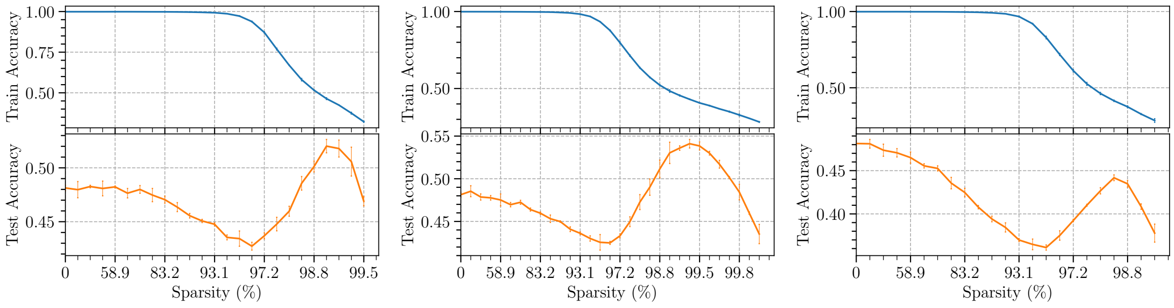


Figure 27. Sparse Double descend phenomenon with different pruning strategies for VGG-16 on CIFAR-100 with symmetric label noise, $\epsilon = 40\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

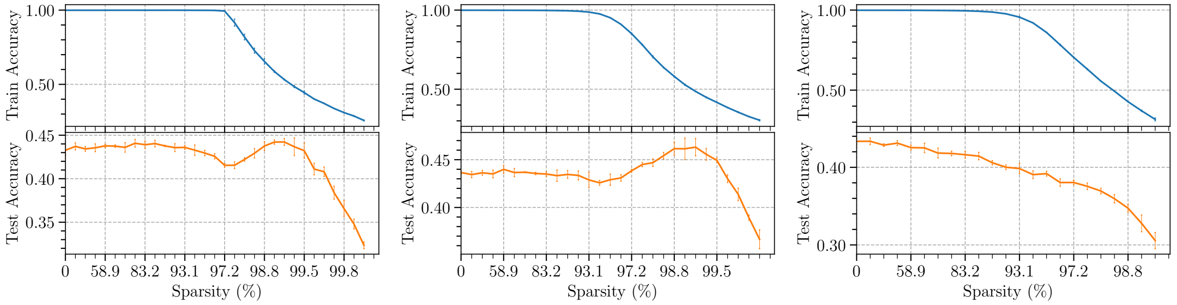


Figure 28. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-100 with asymmetric label noise, $\epsilon = 40\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

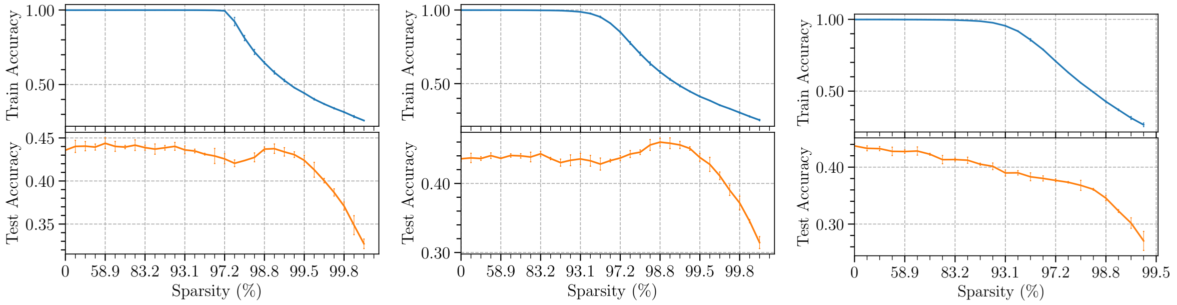


Figure 29. Sparse Double descend phenomenon with different pruning strategies for ResNet-18 on CIFAR-100 with pairflip label noise, $\epsilon = 40\%$. **Left**: Magnitude-based pruning. **Middle**: Gradient-based pruning. **Right**: Random pruning.

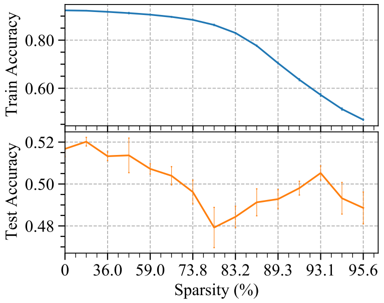


Figure 30. Sparse Double descend phenomenon with magnitude-based pruning for ResNet-101 on Tiny ImageNet with symmetric label noise, $\epsilon = 20\%$.

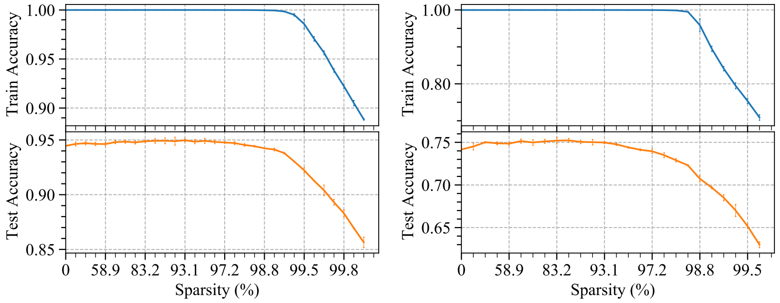


Figure 31. Results of magnitude-based pruning for ResNet-18 on CIFAR with no label noise. **Left**: CIFAR-10. **Right**: CIFAR-100.

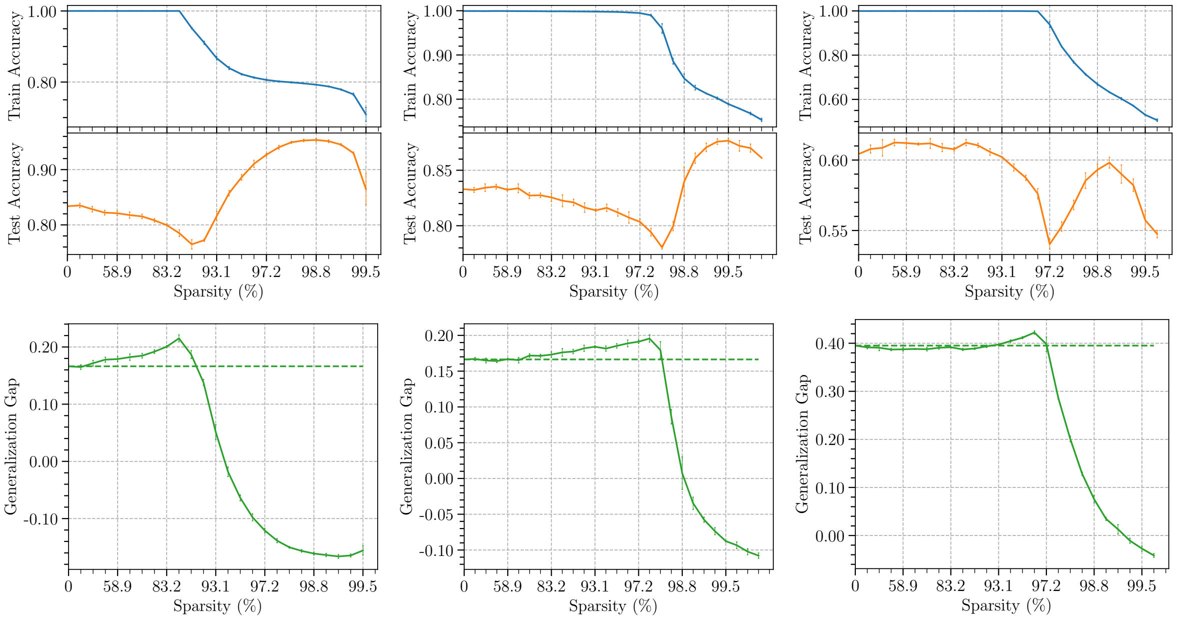


Figure 32. Sparse Double Descent of LeNet-300-100 on MNIST and ResNet-18 on CIFAR with 20% symmetric label noise, pruned with magnitude-based pruning, and retrained with LTR. We show how the generalization performance of networks depends on model sparsity, where generalization gap equals test error - train error. **Left**: MNIST. **Middle**: CIFAR-10. **Right**: CIFAR-100.

C.2. Comparison between Lottery Tickets and Random Initializations

Here we compare the performance of lottery tickets and random initializations. Models at the same sparsity share the same mask structure.

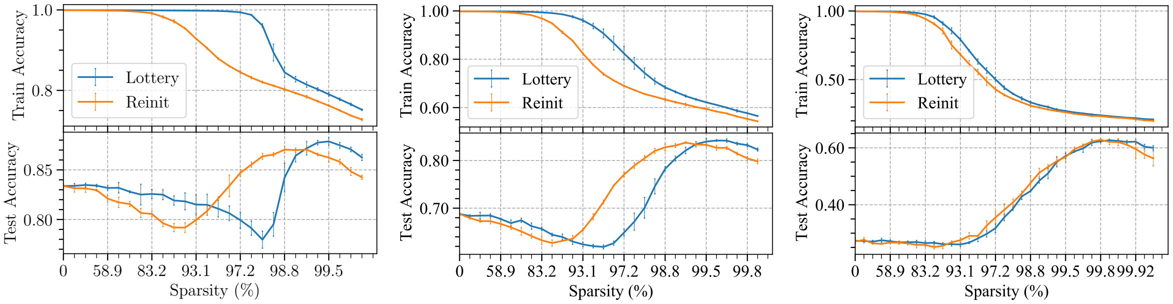


Figure 33. Performance of ResNet-18 on CIFAR-10 when retrained from either the original initialization (lottery tickets), or a random reinitialization. **Left**: $\epsilon = 20\%$. **Middle**: $\epsilon = 40\%$. **Right**: $\epsilon = 80\%$.

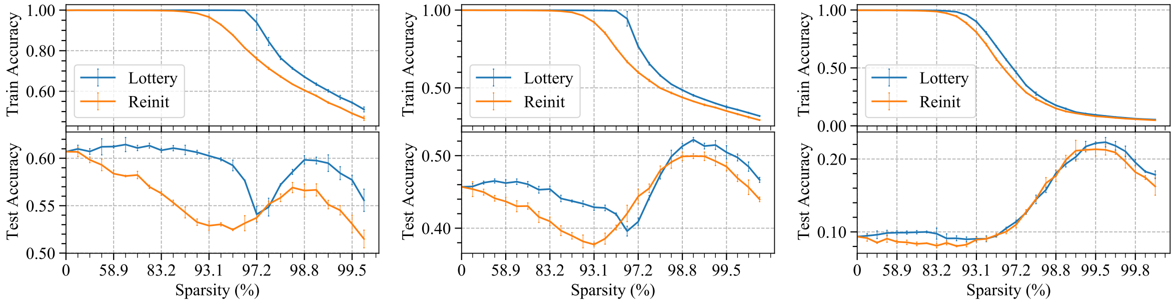


Figure 34. Performance of ResNet-18 on CIFAR-100 when retrained from either the original initialization (lottery tickets), or a random reinitialization. **Left**: $\epsilon = 20\%$. **Middle**: $\epsilon = 40\%$. **Right**: $\epsilon = 80\%$.

C.3. Training Dynamics of Sparse Models

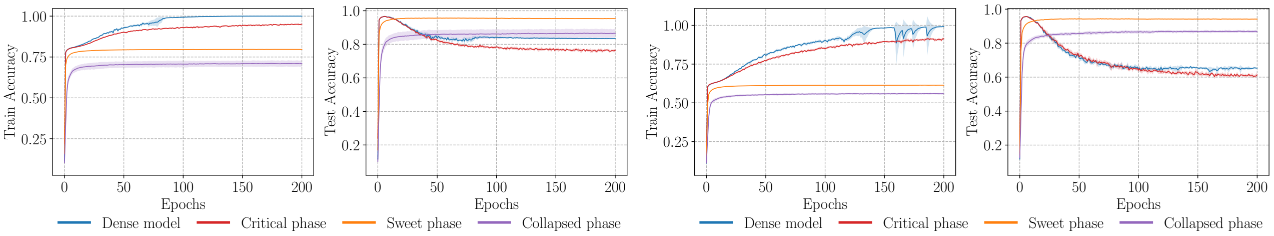
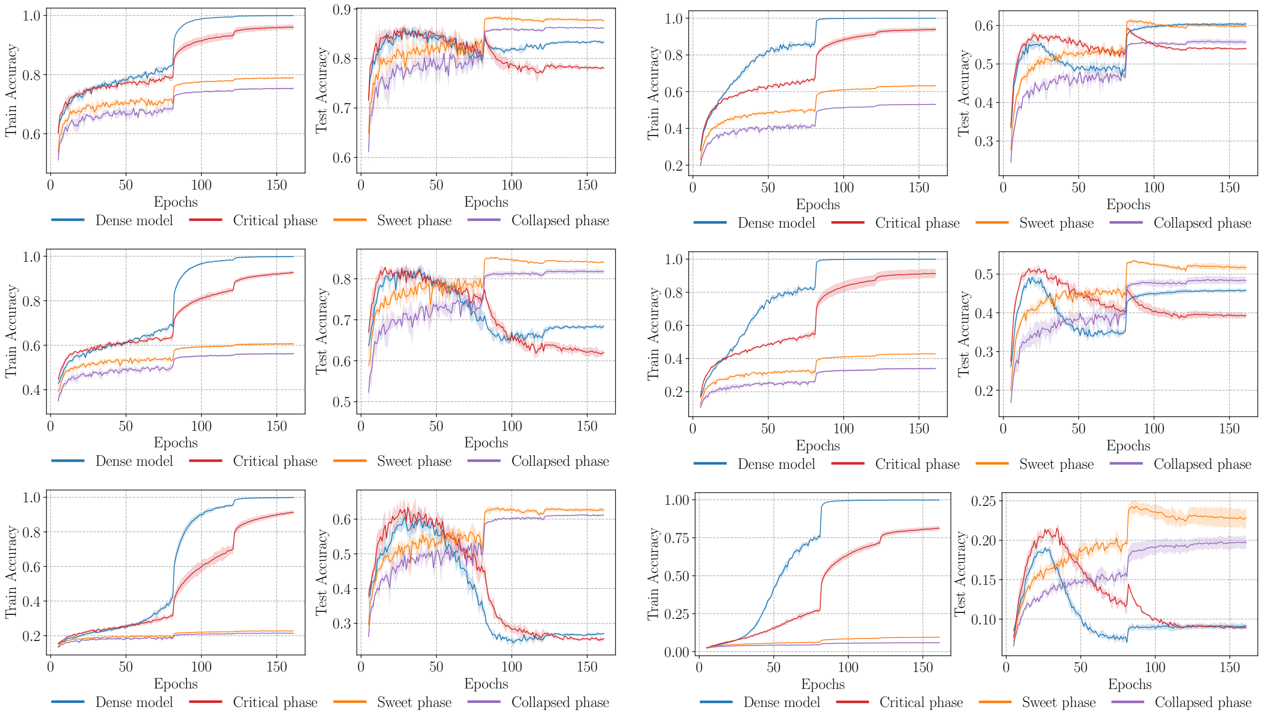


Figure 35. Training dynamics w.r.t. epochs at four sparsities across different permuted fractions. Models are LeNet-300-100 for MNIST pruned with magnitude-based strategy. **Left**: $\epsilon = 20\%$. **Right**: $\epsilon = 40\%$.



(a) CIFAR-10

(b) CIFAR-100

Figure 36. Training dynamics w.r.t. epochs at four sparsities across different permuted fractions. Datasets are CIFAR-10/CIFAR-100 and models are ResNet-18 pruned with magnitude-based strategy. **Top**: $\epsilon = 20\%$. **Middle**: $\epsilon = 40\%$. **Bottom**: $\epsilon = 80\%$.

C.4. Re-dense Training Results

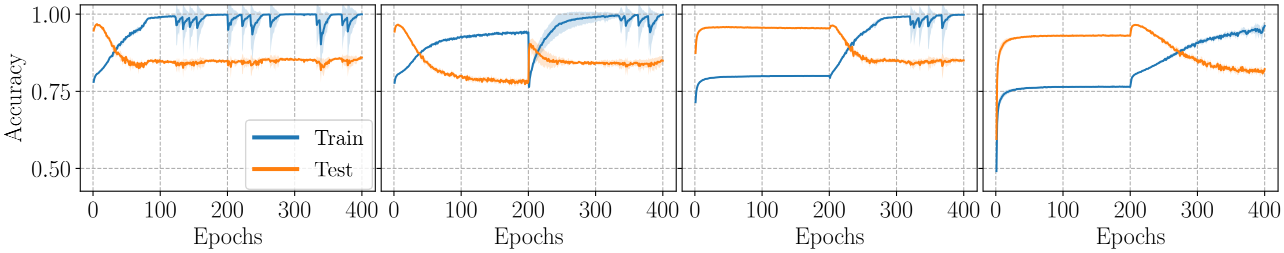


Figure 37. Accuracy curve of the sparse and re-dense training process. We recover pruned weights at epoch 200, and training them from value of zero for another 200 epochs using the last learning rate of sparse training, which is 0.1 for MNIST with $\epsilon = 20\%$, LeNet-300-100.

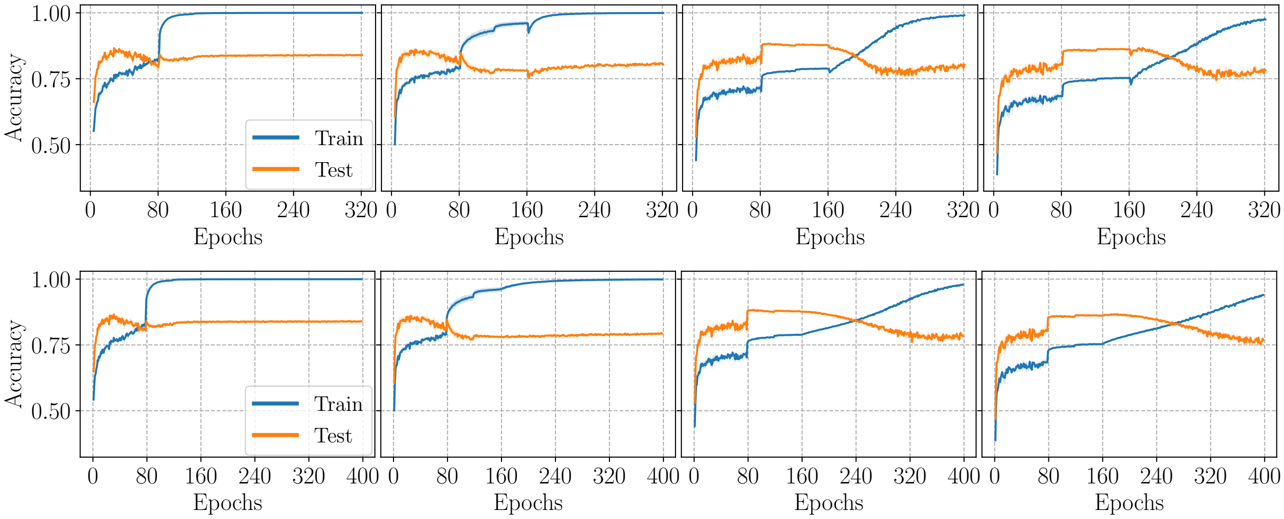


Figure 38. Accuracy curve of the sparse and re-dense training process. We recover pruned weights at epoch 160, and training them from value of zero for another 160 epochs using a small learning rate of sparse training, which is 0.001 as the last epoch learning rate (the top) and 0.0001 (the bottom) for CIFAR-10 with $\epsilon = 20\%$, ResNet-18.

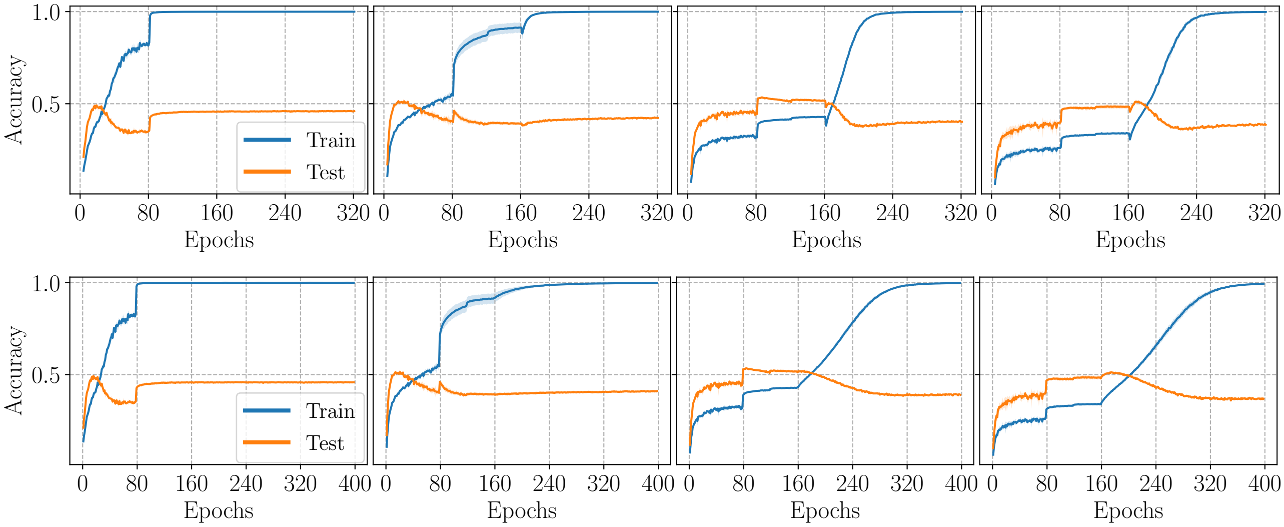
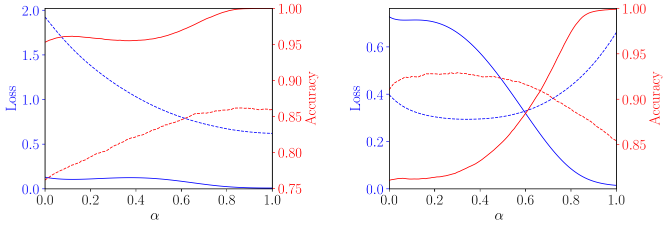


Figure 39. Accuracy curve of the sparse and re-dense training process. We recover pruned weights at epoch 160, and training them from value of zero for another 160 epochs using a small learning rate of sparse training, which is 0.001 as the last epoch learning rate (the top) and 0.0001 (the bottom) for CIFAR-100 with $\epsilon = 40\%$, ResNet-18.

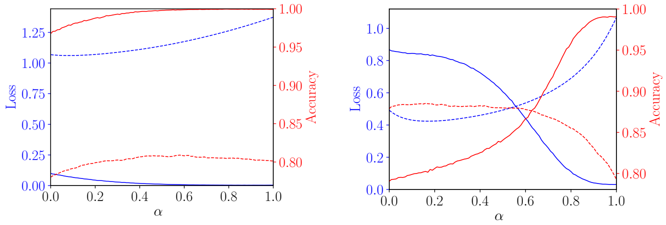
C.5. Linear Interpolation Results



(a) Critical Phase

(b) Sweet Phase

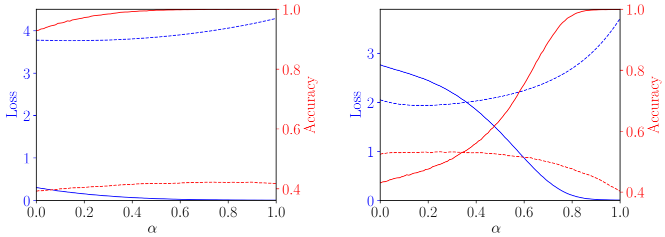
Figure 40. Linear interpolation plots. Models are LeNet-300-100 on MNIST with $\epsilon = 20\%$. $\alpha = 0$ corresponds to sparse solutions and $\alpha = 1$ corresponds to the re-dense solutions. The blue lines are loss curves and the red lines are accuracy curves; solid lines indicate training data set and dashed lines indicate testing data set. For re-dense models, sparsity is measured before recovering weights.



(a) Critical Phase

(b) Sweet Phase

Figure 41. Linear interpolation plots. Models are ResNet-18 on CIFAR-10 with $\epsilon = 20\%$. $\alpha = 0$ corresponds to sparse solutions and $\alpha = 1$ corresponds to the re-dense solutions. The blue lines are loss curves and the red lines are accuracy curves; solid lines indicate training data set and dashed lines indicate testing data set. For re-dense models, sparsity is measured before recovering weights.



(a) Critical Phase

(b) Sweet Phase

Figure 42. Linear interpolation plots. Models are ResNet-18 on CIFAR-100 with $\epsilon = 40\%$. $\alpha = 0$ corresponds to sparse solutions and $\alpha = 1$ corresponds to the re-dense solutions. The blue lines are loss curves and the red lines are accuracy curves; solid lines indicate training data set and dashed lines indicate testing data set. For re-dense models, sparsity is measured before recovering weights.

C.6. 1-D Visualization of Re-dense Solutions

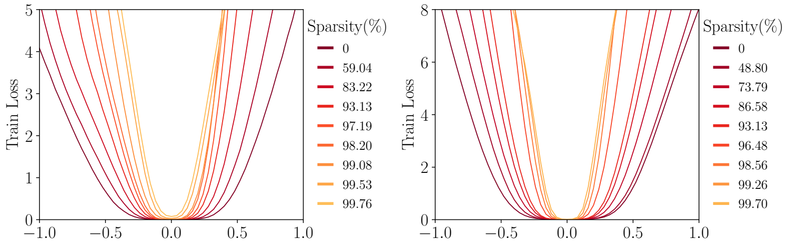


Figure 43. The 1-D loss visualization of minima found by re-dense training using filter normalization (Li et al., 2018).The sparsity of particular model is measured before the re-dense step. The sparser the original pruned network is, the sharper minima it might converge to after re-dense training. **Left**: ResNet-18 on CIFAR-10 with $\epsilon = 20\%$. **Right**: ResNet-18 on CIFAR-100 with $\epsilon = 40\%$.

C.7. Learning Distance Results

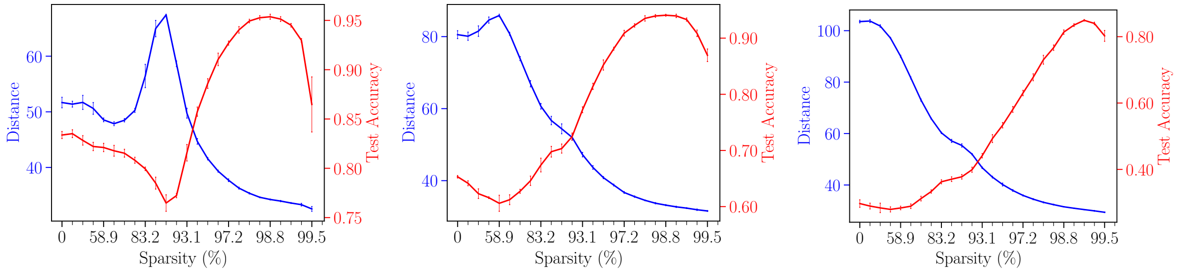


Figure 44. Learning distance for LeNet-300-100 on MNIST. The blue lines refer to ℓ_2 learning distance and the red lines are test accuracy. **Left**: $\epsilon = 20\%$. **Middle**: $\epsilon = 40\%$. **Right**: $\epsilon = 80\%$.

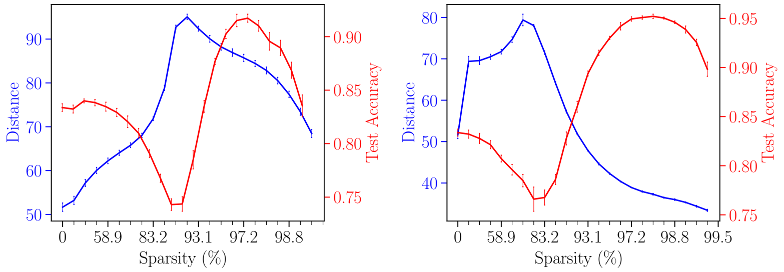


Figure 45. Learning distance for LeNet-300-100 on MNIST with $\epsilon = 20\%$. The blue lines refer to ℓ_2 learning distance and the red lines are test accuracy. **Left**: Retraining using learning rate rewinding. **Right**: Scratch retraining.

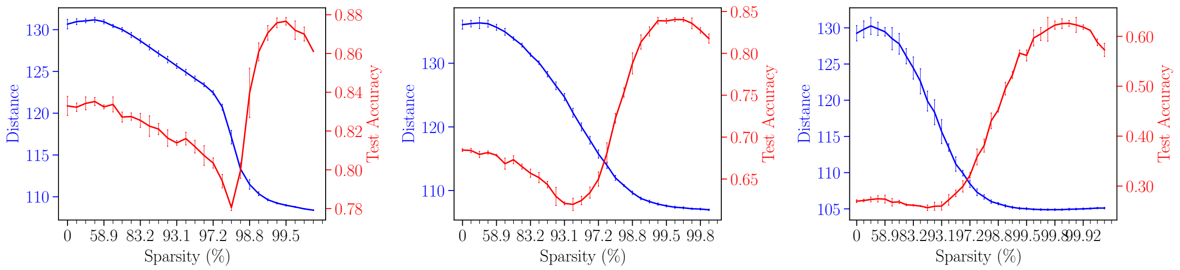


Figure 46. Learning distance for ResNet-18 on CIFAR-10. The blue lines refer to ℓ_2 learning distance and the red lines are test accuracy. **Left**: $\epsilon = 20\%$. **Middle**: $\epsilon = 40\%$. **Right**: $\epsilon = 80\%$

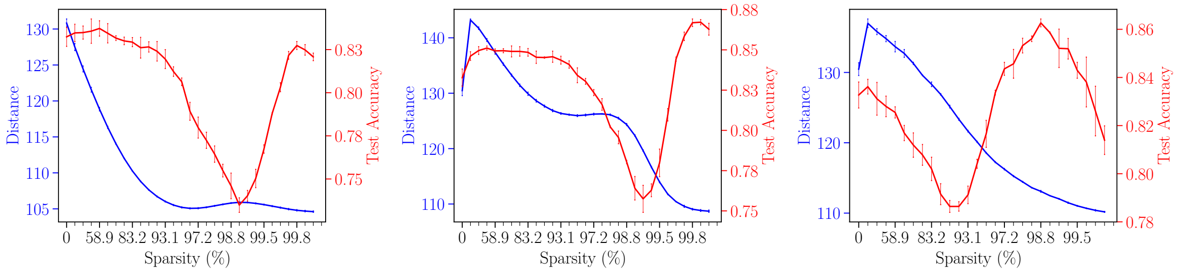


Figure 47. Learning distance for ResNet-18 on CIFAR-10 with different retraining methods. The blue lines refer to ℓ_2 learning distance and the red lines are test accuracy. **Left**: Finetuning. **Middle**: Learning rate rewinding. **Right**: Scratch retraining.

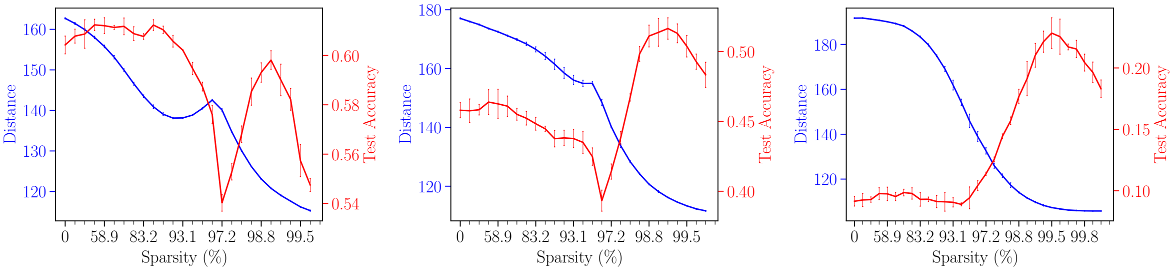


Figure 48. Learning distance for ResNet-18 on CIFAR-100. The blue lines refer to ℓ_2 learning distance and the red lines are test accuracy. **Left**: $\epsilon = 20\%$. **Middle**: $\epsilon = 40\%$. **Right**: $\epsilon = 80\%$

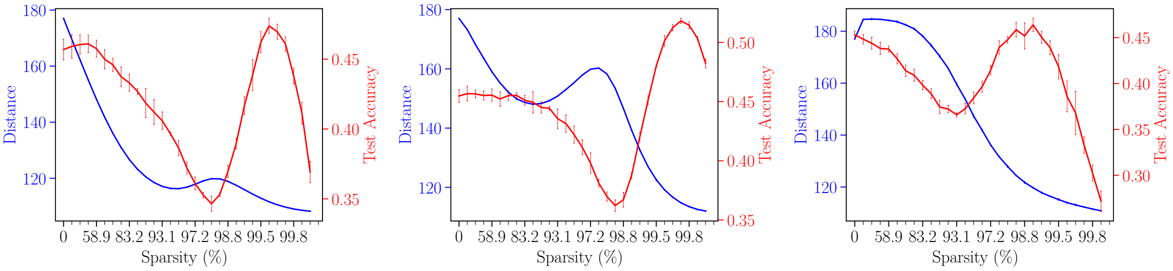


Figure 49. Learning distance for ResNet-18 on CIFAR-100 with different retraining methods. The blue lines refer to ℓ_2 learning distance and the red lines are test accuracy. **Left**: Finetuning. **Middle**: Learning rate rewinding. **Right**: Scratch retraining.