Pruning's Effect on Generalization through the Lens of Training and Regularization

Tian Jin
MIT CSAIL
tianjin@csail.mit.edu

Daniel M. Roy
University of Toronto,
Vector Institute
droy@utstat.toronto.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

Jonathan Frankle
MosaicML
jonathan@mosaicml.com

Gintare Karolina Dziugaite
Google Research, Brain Team
Mila; McGill
gkdz@google.com

Abstract

Practitioners frequently observe that pruning improves model generalization. A long-standing hypothesis based on bias-variance trade-off attributes this generalization improvement to model size reduction. However, recent studies on over-parameterization characterize a new model size regime, in which larger models achieve better generalization. Pruning models in this over-parameterized regime leads to a contradiction – while theory predicts that reducing model size harms generalization, pruning to a range of sparsities nonetheless improves it. Motivated by this contradiction, we re-examine pruning's effect on generalization empirically. We show that weight removal cannot fully account for the generalization-improving effect of standard pruning algorithms. Instead, we find that pruning lead to better training at specific sparsities, improving model training loss over the dense model. We find that pruning also lead to additional regularization at other sparsities, reducing the accuracy degradation due to noisy examples over the dense model. Pruning extends model training time and reduces model size. These two factors improve training and add regularization respectively. We empirically demonstrate that both factors are essential to fully explaining pruning's impact on generalization.

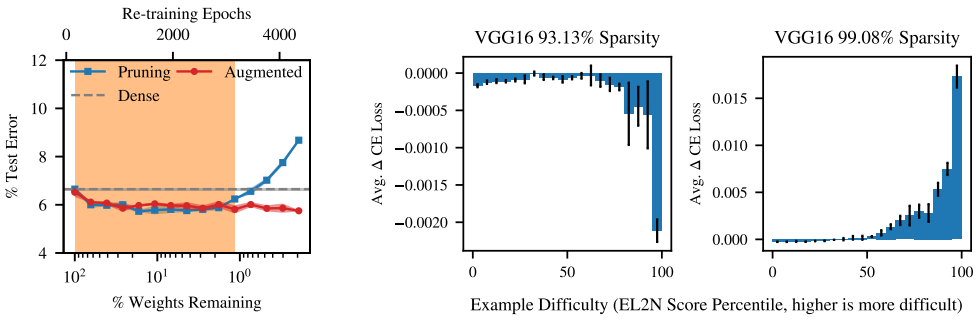
1 Introduction

Neural network pruning techniques remove unnecessary weights to reduce the memory and computational requirements of a model. Practitioners can remove a large fraction (often 80-90%) of weights without harming *generalization*, measured by test error [31, 19, 13]. While recent pruning research (e.g., [19, 13, 52, 7, 30, 43, 11, 66, 4, 34, 64, 53, 35, 37, 56, 54, 9, 67, 59]) focuses on reducing model footprint, improving generalization has been a core design objective for earlier work on pruning [31, 22]; recent pruning literature also frequently notes that it improves generalization [19, 13].

How does pruning affect generalization? An enduring hypothesis claims that pruning may benefit generalization by reducing *model size*, defined as the number of weights in a model.[1] We refer

[1] This hypothesis traces back to seminal work in pruning, such as Optimal Brain Surgeon [22]: "without such weight elimination, overfitting problems and thus poor generalization will result." And again in pioneering work [19] applying pruning to deep neural network models: "we believe this accuracy improvement is due to pruning finding the right capacity of the network and hence reducing overfitting." Because the latter hypothesis leaves the definition of network capacity unspecified, we examine an instantiation of it, measuring network capacity with model size.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) (b)

Figure 1: (a) Retraining the VGG-16 model on CIFAR-10 exactly as done in pruning, but without removing any weights produces models with similar generalization as pruning to the shaded range of weights remaining. We show the x-axis for pruning/augmented training below/above the plot respectively. (b) We show the loss difference between the sparse/dense model produced by pruning/standard training respectively for training examples grouped by EL2N score percentiles. Left: We find that pruning to a relatively low sparsity may improve generalization by decreasing training loss, with a particular decrease on difficult examples with a large influence on generalization. Right: We find that pruning to a relatively high sparsity may also improve generalization by increasing training loss, also with a particular increase on difficult examples with a large influence on generalization.

to this as the *size-reduction hypothesis*. However, algorithms for pruning neural networks and our understanding of the impact of model size on generalization has changed.

First, pruning algorithms have grown increasingly complex. Learning rate rewinding [52], a state-of-the-art algorithm prunes weights iteratively. An iteration consists of *weight removal*, where the algorithm removes a subset of remaining weights, followed by *retraining*, where the algorithm continues to train the model, replaying the original learning rate schedule. Across iterations, this retraining scheme effectively adopts a cyclic learning rate schedule [57], which may benefit generalization.

Second, emerging empirical and theoretical findings challenge our understanding of generalization for deep neural network models with ever-increasing number of weights [47, 12, 45, 68]. In particular, the size-reduction hypothesis builds on the classical generalization theory on *bias-variance trade-off*, which predicts that reducing model size improves generalization of an overfitted model [23]. However, recent work [5, 46] reveals that beyond this classical bias-variance trade-off regime lies an over-parameterized regime where models are large enough to achieve near-zero training error. In this regime, bigger models often achieve better generalization [5, 46, 29, 27, 58]. Practitioners often apply modern pruning algorithms to models in this over-parameterized regime. While double descent curves predict that reducing the size of an over-parameterized model should harm generalization, pruning to a range of sparsities nevertheless improves it. The size-reduction hypothesis is therefore inconsistent with our empirical observation about pruning.

A new direction. We first confirm that the size-reduction hypothesis does not fully explain pruning's effect on generalization. Figure 1a compares the generalization of a family of sparse models of different sizes generated by a state-of-the-art pruning algorithm [52], and the generalization of a family of models generated by the same algorithm, except modified to no longer remove weights. This algorithm, which we refer to as the *augmented training algorithm*, differs from standard training in two ways: (1) the number of training epochs is larger; (2) the learning rate schedule is cyclic.

The results show that for any sparse VGG-16 model generated by the pruning algorithm with more than 1% of its original weights, its generalization is indistinguishable from that of the model generated by the augmented training algorithm that does not remove weights but trains the model for the same number of epochs and with the same learning rate schedule as pruning algorithm does.[2] For a range of sparsities, pruning's effect on generalization remains unchanged in the absence of weight removal, illustrating that the size-reduction hypothesis cannot fully explain pruning's effect on generalization.

[2]In Appendix C we show this phenomenon for 4 additional benchmarks, and other pruning algorithms.

Our approach. We instead develop an explanation for pruning's impact on generalization through an analysis of its effect on each training example, measured by the difference in its training loss between the sparse model produced by pruning and the dense model produced by standard training. We interpret pruning's effect on each example in relation to the example's influence on generalization. We measure this influence with the absolute change to generalization due to leaving out an example, which Paul et al. [50] approximates with the L2 distance between the predicted probabilities and one hot labels early in training, called the EL2N score. We refer to high EL2N examples as *difficult examples*, as initially the model predict their labels poorly. Our analysis shows the following results.

Better training. We prune a VGG-16 model on the CIFAR-10 dataset to the *optimal sparsity*, the sparsity with the best validation error. At this sparsity, we find that the pruned model achieves better training loss than the dense model. In the left plot in Figure 1b, we present the difference in loss between the sparse and dense model for the training examples grouped by their EL2N score percentiles. Difficult examples, which influence generalization the most, show the greatest training loss improvement. Pruning may therefore lead to *better training*, improving the training loss of the sparse model over the dense model, particularly on examples with large influence on generalization.

Additional regularization. We prune the same VGG-16 model on CIFAR-10 dataset with the pruning algorithm configured to produce the sparsest model that still improves generalization over the dense model. At this sparsity, we find that the pruned model achieves worse training loss than the dense model. In the right plot in Figure 1b, we present the difference in loss between the sparse and dense model for the training examples grouped by their EL2N score percentiles. The results show that pruning may also improve generalization while increasing the training loss of the sparse model over the dense model, particularly on difficult examples with large influence on generalization.

While it is perhaps counter-intuitive that increasing training loss, particularly on the most difficult examples may improve generalization, examining these pruning-affected examples leads us to hypothesize that it is due to an observation that Paul et al. [50] made: in image classification datasets, a small fraction of difficult and influential training examples have ambiguous or erroneous labels, training on which impairs generalization. Therefore, it is instead better for the model to not fit them.

We examine this hypothesis by introducing random label noise into the training dataset. We find that on such datasets, pruning may improve generalization by increasing the training loss on the noisy examples, while still preserving the training loss of examples from the original training dataset. The result is an overall generalization improvement for the sparse model. Notably, reducing model width has a similar effect on training loss and generalization, suggesting reduced number of weights in the model as the underlying cause for generalization improvement in the presence of noisy examples. We refer to the effect of removing weights as *additional regularization*, which reduces the accuracy degradation due to noisy examples over the original dense model.

Implications. It has been a long-standing hypothesis that pruning improves generalization by reducing the number of weights (via minimum description length principle [18] or Vapnik–Chervonenkis theory [62]). Our empirical observations suggest that the theory and practice of pruning should instead focus on the two effects of modern pruning algorithms: better training and additional regularization, both of which are indispensable to explaining pruning's benefits to generalization fully. Our results also suggest that quantifying pruning's heterogenous effect across training examples is key to understanding pruning's influence on generalization.

Contributions. We present the following contributions:

1. We find that pruning may lead to *better training*, improving the training loss, as well as generalization over the dense model at the optimal sparsity.
2. We find that pruning may also lead to *additional regularization*, reducing the accuracy degradation due to noisy examples, and improving generalization over the dense model at the optimal sparsity.
3. We demonstrate that our deconstruction of pruning's effect into training and regularization cannot be further simplified — alternative explanations such as increased training time and reduced model size only partially explain the effects of pruning on generalization.

2 Preliminaries

We study *Iterative Magnitude Pruning (IMP) with learning-rate rewinding* [52]. This algorithm consists of four steps: (1) Train a neural network to completion. (2) Remove a specified fraction of the smallest-magnitude remaining weights. (3) Reset the learning rate to its value in an earlier epoch t and (4) Repeat steps 1-3 iteratively until the model reaches the desired overall sparsity.

Terminology. *Dense models* refer to models without any removed weights. *Sparse models* refer to models with removed weights as a result of pruning. The *optimally sparse model* is the model pruned to the sparsity at which it achieves the best validation error. The *optimal sparsity* is the sparsity of the optimally sparse model. *Generalization* refers to the classification error on the test set.

Experimental methodology. We use standard architectures: LeNet [32], VGG-16 [55], ResNet-20, ResNet-32 and ResNet-50 [24], and train on benchmarks (MNIST, CIFAR-10, CIFAR-100, ImageNet) using standard hyperparameter settings and standard cross-entropy loss function [13, 14, 65]. Following Frankle and Carbin [13], Frankle et al. [14], we set the t hyperparameter of IMP to $t = 0$ for MNIST-LeNet benchmark and $t = 10$ for the others. Appendix B shows further details.

We report all results by running the same experiment 3 times with distinct random seeds. We conclude that any real-valued results of two experiments differ if and only if the mean difference between three independent runs of respective experiments is at least one standard deviation away from zero, otherwise we say that the results *match*. We use PyTorch [49] on TPUs with OpenLTH library [13].

Limitations. Our work is empirical in nature. Though we validate each of our claims with extensive experimental results on 18 independent benchmarks with different architectures, datasets and noise levels, we recognize that our claims may still not generalize due to the empirical nature of our study.

3 Generalization Improvement from Better Training

Not all training examples have the same effect on training and generalization [61, 50, 8, 21, 25]. In this section, we first identify which training examples are most affected by pruning, where we measure pruning's effect on each training example by the difference in its training loss between the sparse and dense model. We find that pruning most significantly affect examples that are most difficult for the dense model to learn early in training. We then connect these examples to their effects on generalization: the most affected examples also tend to have the highest impact on generalization.

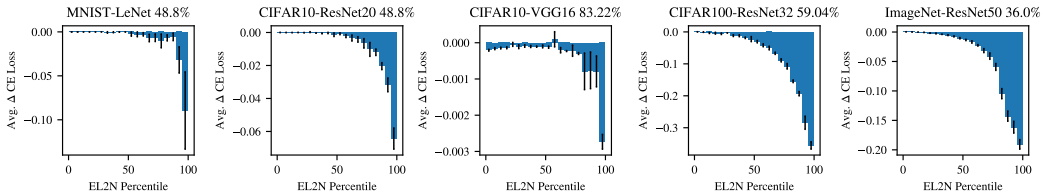
Method. Paul et al. [50] propose to estimate an example's influence on model generalization with the EL2N (**E**rror **L2** **N**orm) score – the L2 distance between the predicted probability and the one hot label of the example early in training. Details for computing the EL2N score is available in Appendix B.3. They show that an example with high EL2N score may cause large weight update, thereby substantially changing the predictions and loss of other examples. Paul et al. [50] rely on this relationship between EL2N score and generalization to remove examples with little influence on generalization from the dataset to accelerate training without affecting model generalization. We similarly use the EL2N score to assess the impact of pruning on generalization.

To measure the difference in training loss on examples between the sparse and dense model, we partition the training set into M subgroups, each with a different range of EL2N scores. We then compute the average training loss difference on examples in each subgroup. We compute this difference by subtracting the training loss of the dense model from that of the sparse model. A negative value indicates that pruning improves the subgroup training loss. We pick $M = 20$ because it is the largest value of M that enables us to clearly present per-subgroup training loss difference.

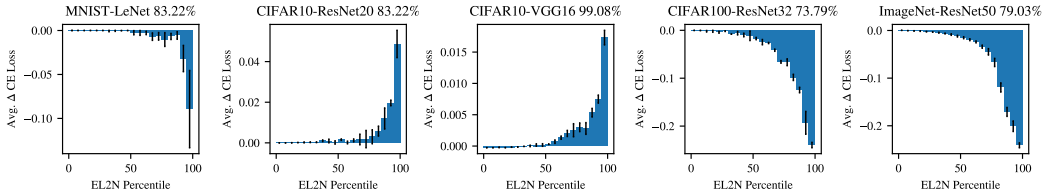
We measure training loss at two sparsities of interest: the sparsity that achieves the best generalization, and the highest sparsity that still attains better generalization than the dense model. In Appendix L, we present pruning's effect on subgroup training loss at a wider range of sparsities.

Results. Figure 2a presents pruning's effect on subgroup training loss. Our results show that at the sparsity that achieves the best generalization, for all subgroups, the average training loss of the pruned model either matches or improves over the dense model, indicating better training.

Figure 2a also shows that pruning's effect on training example subgroups is nonuniform. Pruning most significantly affects the most difficult examples – the ones with the highest range of EL2N scores. Specifically, the average magnitude of pruning's effect on examples with top 20% EL2N



(a) At sparsity achieving the best generalization



(b) At highest sparsity with generalization exceeding that of the dense counterpart

Figure 2: Difference between the average training loss of the sparse/dense model on example subgroups with distinct EL2N percentile range. Negative values indicate that pruning improves loss.

scores is 3.5-71x that on examples with bottom 80% EL2N scores. Pruning to the said sparsity thus improves generalization while improving the training loss of example subgroups nonuniformly, with an emphasis on difficult examples that are most influential to generalization. In Appendix D, we validate the effect of pruning-improved examples on generalization: excluding 20% of pruning-improved examples hurts generalization more than excluding a random subset of the same size.

Conclusion. On standard datasets, pruning to the sparsity that achieves the best generalization leads to better training – the training loss of most example subgroups improves over the dense model. The subgroup with the most influence on generalization sees the largest training loss improvement.

4 Generalization Improvement from Additional Regularization

Pruning to the highest sparsity that improves generalization (Figure 2b) displays a similar effect as pruning to the sparsity that attains the best generalization (Figure 2a) – generalization improves with an overall improved training loss over the dense model, except for CIFAR-10, where pruning improves generalization while worsening training loss on most subgroups of examples. Specifically, for ResNet20 on CIFAR-10 at 83.22% sparsity (Figure 2b, second from left) and VGG-16 on CIFAR-10 at 99.08% sparsity (Figure 2b, third from left), the training loss of the sparse model is worse than the dense model. While pruning always increases training loss when it removes a large enough fraction of weights, the accompanied generalization improvement does not always occur in general.

The second and third plots of Figure 2b show that the loss increase is especially pronounced on particular subgroups of difficult examples, which have a substantial influence on generalization. Examining the training examples whose training loss increases the most due to pruning in Appendix K.1 leads us to hypothesize that it is due to an observation Paul et al. [50] made: although difficult examples are predominately beneficial for generalization, there exist a small fraction of noisy examples, such as ones with ambiguous or erroneous labels, that are difficult and exert a significant influence on generalization, albeit in a harmful way.[3]

To precisely characterize pruning's effect on noisy examples, we inject random label noise to the training dataset. Doing so enables comparison of pruning's effect on noisy data and original data.

Method. We inject $p\%$ random label noise by selecting $p\%$ examples uniformly at random and changing the label of each example to one of the other labels in the dataset sampled uniformly at random. We refer to the data that is not affected as original data. We sample and fix the random labels before each independent run of an experiment.

[3]We present leave-subgroup-out retraining experiments for these examples in Appendix K.

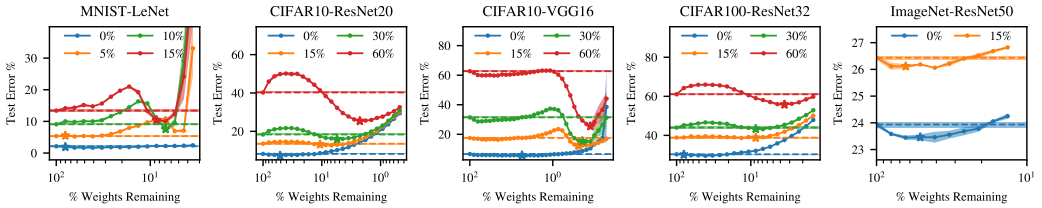


Figure 3: Pruning improves generalization over the dense model in the presence of random label noise. We show the generalization of the sparse models as a function of percentage weights remaining. Horizontal dashed lines indicate the baseline generalization of the dense models. We mark the optimal sparsity with a star. Legend shows the fraction of the training examples with random label noise.

We report the following results for models trained on datasets with and without random label noise: (1) pruning's impact on generalization versus sparsity (Figure 3); (2) pruning's impact on training loss for the original data and random label data versus sparsity (Figure 4);[4] (3) pruning's impact on training loss for subgroups of examples, each with a different range of EL2N scores (Figure 5).

Generalization results. Figure 3 shows that, in the presence of random label noise, the generalization of the optimally sparse models is better than that of the dense models on 11 out of 13 benchmarks and matches that of the dense models on the rest of the benchmarks. Our results provide evidence that pruning to the optimal sparsity improves generalization in the presence of random label noise.

The extent of pruning's generalization improvement at the optimal sparsity grows as the portion of label noise increases. For example, for VGG-16 on CIFAR-10 benchmark, pruning to the optimal sparsity improves generalization by 5.2%, 16.1% and 34.6% on 15%, 30% and 60% label noise.

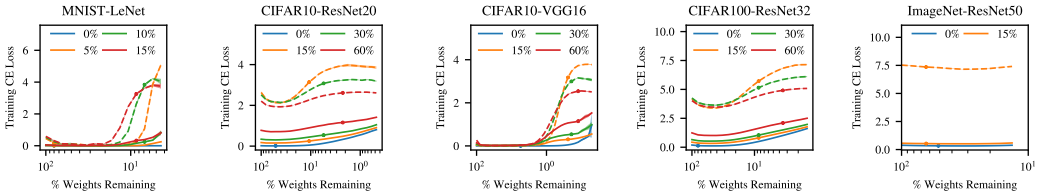


Figure 4: Pruning to the optimal sparsity increases training loss over the dense model in the presence of random label noise. Noisy examples see particular large loss increase. We show the training loss on the original and noisy data with dashed and solid lines respectively, as functions of % weights remaining. We mark the optimal sparsity with a star. Legend shows the random label noise level.

Original versus noisy data. Figure 4 shows that pruning initially reduces the average training loss on noisy examples: for instance, training loss on noisy examples is slightly lower for the sparse ResNet20 models on CIFAR-10 with many weights remaining (i.e., more than 10%) than for the dense model. Since noisy examples are difficult examples [50], these results indicate better training, which improves model training loss with a particular emphasis on difficult examples.

As the fraction of weights remaining drops further, the loss of the sparse model on noisy data grows. For instance, the training loss on noisy examples increases for the sparse ResNet20 models on CIFAR-10 with few weights remaining (i.e., less than 10%). Moreover, the gap between the average training loss of the sparse model on original versus noisy examples widens beyond the dense model, indicating that pruning leads to additional regularization. In particular, for a range of sparsities where generalization improves over the dense model, the sparse model fits original examples significantly better than those with randomized labels. Taking the dense VGG-16 model trained on the CIFAR-10 dataset with 15% - 60% random label noise as an example, we observe that this model incurs 0.03 - 0.13 higher training loss on the partition with injected random label noise than on the partition without. Pruning to the optimal sparsity widens this training loss gap to 1.38 - 2.97. Furthermore, the gap between the average training loss of the sparse models on original versus noisy examples eventually diminishes as the sparsity increases even further.

[4]Numerical results available in Table 11 in Appendix M

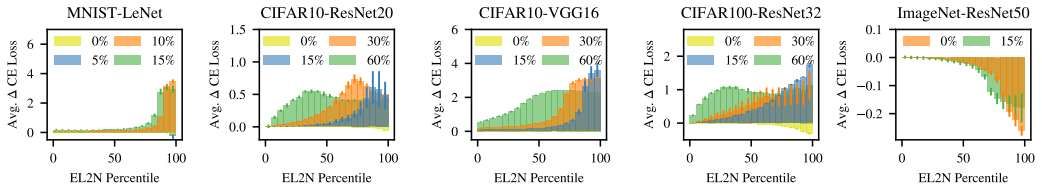


Figure 5: In the presence of random label noise, pruning to the optimal sparsity often increases training loss (except for ResNet50) over the dense model, especially to highly influential examples. For example subgroups with distinct EL2N percentile range, we show the difference in average training loss between dense and optimally sparse models. Legend indicates label noise level. We do not show LeNet with 5% noise because the dense model achieves the best generalization. For ResNet20 and ResNet32 with 60% noise, the training loss for high EL2N examples is already very high for dense models to begin with, leaving little room for their loss to increase further by pruning.

Difference in loss by subgroups. Figure 5 shows the difference between loss of the dense and optimally sparse model on subgroups of examples with different EL2N scores, but this time in the presence of random label noise. The results show that pruning to the optimal sparsity often increases the loss of difficult examples, contrary to Figure 2 shows when training on original data only where pruning decreases the training loss of difficult examples.

Since random label examples are difficult and harm generalization [50], increasing the training loss on difficult examples may therefore improve generalization. In Appendix E, we validate that excluding examples misclassified by the optimally sparse models improves generalization similar to pruning.

Conclusion. We study pruning's effects on training and regularization by connecting them to noisy examples. By introducing random label noise, we show the two effects of pruning more clearly. At relatively low sparsities, we observe better training because the training loss of sparse models is often lower than the dense model, especially on difficult examples highly influential to generalization. As sparsity increases, regularization effects dominate, because the sparse models see a larger gap between training loss on the noisy versus original data than the dense model. We note that across all sparsities, pruning has a highly nonuniform effect across example subgroups with different difficulties (EL2N scores) on both standard datasets and datasets with random label noise.

Crucially, the effect of pruning at the optimal sparsity – be it better training or additional regularization – depends on the noise levels in the dataset. On datasets with few noisy examples, Figure 2 shows that at the optimal sparsity, better training is the key to good generalization, because training loss reduces. On datasets with injected random label noise, Figure 5 shows that the optimal sparsity is the one at which pruning-induced regularization effects increase noisy data loss.

5 Isolating the Effects of Extending Training Time and Reducing Model Size

We identified two effects of pruning on generalization: better training and additional regularization. They intersect with two components of the pruning algorithm: extended training time and model size reduction, respectively. In this section, we study these two effects in isolation by extending dense model training time and reducing dense model size by scaling down model width.[5]. We confirm that both effects of pruning are necessary to fully explain pruning's impact on generalization. Thus our deconstruction of pruning's effect in terms of training and regularization cannot be further simplified.

5.1 Extended Training Time

What happens to generalization if the weight removal part is removed from pruning, and only the retraining part remains? To answer, we train dense models exactly as done in the pruning algorithm, for the same number of gradient steps and learning rate schedule, but without removing weights.

[5]To isolate other effects of pruning, we also evaluate the contribution of design choices of pruning algorithms, such as the heuristic selecting which weights to prune, to generalization via ablation studies in Appendix J

Method. We study the effect of training dense models for extra gradient steps on generalization, without removing any weights. When training the dense models, we use the same learning rate schedule as pruning, replaying the original learning rate schedule designed for the dense model multiple times, effectively adopting a cyclic learning rate schedule. We refer to this training algorithm as *extended dense training*. We report and compare the generalization of models that pruning and extended dense training algorithm produce.

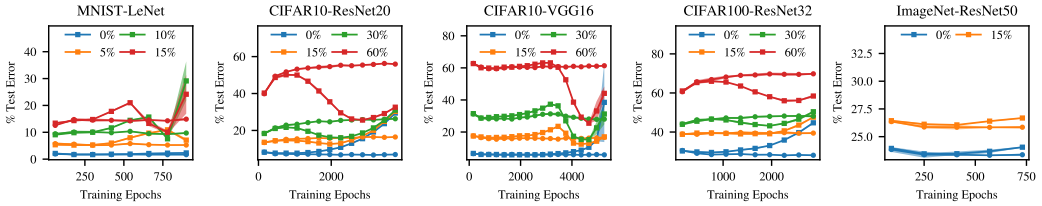


Figure 6: Extended training time does not explain the full extent of pruning's benefits to generalization. We show the test errors of the dense and sparse models with circle and square markers respectively as a function of the training epochs. Legend shows random label noise level.

Results. Figure 6 shows test errors of models that extended dense training and pruning produce as a function of training epochs.[6] The two algorithms produce models with similar generalization on standard datasets without random label noise, where pruning improves training at optimal sparsity, as shown in Section 3. Indeed, in Appendix L, we show that longer training time similarly leads to better training. However, on benchmarks with random label noise, extended dense training underperforms pruning on 9 out of the 13 benchmarks. On LeNet, ResNet20, VGG-16, ResNet32 and ResNet50 benchmarks with noise, extended dense training achieves test errors that are worse than pruning by 0 (matching) to 2.6%, 0.4 to 14.5%, 3.6 to 34.2%, 0 (matching) to 4.8% and -0.3%, respectively.

Conclusion. On standard datasets without random label noise, the extended dense training algorithm produces models with generalization that matches or exceeds the optimally sparse model. However, with random label noise, the extended dense training algorithm can no longer produce models that matches the generalization of the optimally sparse model.

5.2 Size Reduction

What happens to generalization if the retraining part is removed from pruning, and only the weight removal part remains? To answer, we compare pruning with model width down-scaling, since both techniques reduce the number of weights in a model.

At optimal sparsities on datasets without random label noise, in Section 1 we show that size reduction is not necessary to replicate generalization improvements onto dense models; in this section, we show that size reduction is also not sufficient to replicate generalization improvement onto dense models.

Method. We reduce model size by down-scaling the width of a dense model: we train a sequence of dense models using the standard number of training epochs, where the next model in the sequence has 80% of the width of the model preceding it. This width scaling ratio mimics our sparsity schedule, where the pruned model always have 80% of the remaining weights in the previous iteration. We then compare the generalization of models that pruning and width down-scaling produce.

Results. Figure 7 presents the test errors of models that pruning and width down-scaling produce, using square and circle markers respectively. Width down-scaling under-performs pruning on 4 out of 5 benchmarks on datasets without random label noise. When training with random label noise, width down-scaling has notable regularization effects, and significantly improves generalization, similar to pruning. In Appendix I, we show that the regularization effects of both algorithms manifest similarly across example subgroups, suggesting that pruning's regularization effect is a consequence of model size reduction in general. However, despite this improvement, a gap remains between the minimum test error of models that pruning and width down-scaling produce achieve: pruning does better on LeNet, ResNet20, VGG-16, ResNet32 and ResNet50 benchmarks by -4 (pruning does worse) to 0.4%, 0% (matching), 1.5 to 1.8%, 0 (matching) to 2.1%, and 0.36% in test error, respectively.

[6] Numerical results available in Table 5 in Appendix M.

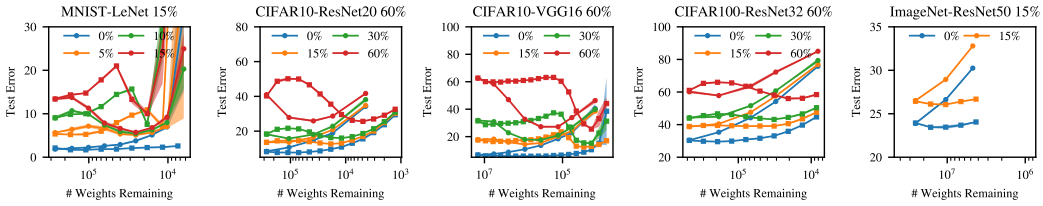


Figure 7: Size reduction does not explain the full extent of pruning's benefits to generalization. We show the test errors of the dense/sparse models with round/square markers respectively as a function of the weights remaining. Legend shows label noise level. Appendix F include detailed views.

Conclusion. On standard datasets with random label noise, width down-scaling produces models with similar generalization (no worse by more than 2.1%) as the optimally sparse models. However, without random label noise, width down-scaling under-performs pruning.

5.3 Conclusion

By removing the weight removal part from pruning in Section 5.1, we show the importance of pruning-induced additional regularization for achieving good generalization in the presence of label noise more clearly than Section 4. Similarly, by removing the retraining part from pruning in Section 5.2, we show the importance of pruning-induced better training for achieving good generalization using the original dataset without random label noise more clearly than Section 3. We confirm that both effects of pruning are required to fully account for its impact on generalization. Our deconstruction of pruning's effect in terms of training and regularization is therefore minimum.

6 Related Work

Pruning has an extended history: early work starting from the 1980s found that pruning enhances model interpretability and generalization [44, 31, 22]. More recently, the advent of deep neural network models motivates the adoption of pruning to reduce the storage and computational demand of deep models [19, 20, 30, 39, 7, 43, 11, 66, 4, 34, 64, 53, 35, 37, 56, 54, 9, 67, 59]. In this section, we describe several branches of pruning research pertinent to our work.

Pruning algorithm design. Recent pruning research often focuses on improving pruning algorithm design. An extended line of research studies the heuristics that determine which weights to remove [43, 11, 33] – the simplest heuristic is to remove weights with the smallest magnitude. For example, Molchanov et al. [42], Louizos et al. [41] propose to learn which weights to prune as part of the optimization process. Another line of research concerns the structure of the weights to remove to achieve computational speedup without significantly hurting model generalization. For example, Li et al. [36], Liu et al. [40] propose to prune weights in groups, or to prune neurons, convolutional filters, channels. Our work does not produce new pruning algorithm design. Rather, we study the existing pruning algorithms to understand their effect on model generalization.

Pruning's effect on generalization. There are many reported instances in which pruning benefits model generalization [31, 19, 13]. This effect has sparked interest in examining pruning as a technique to gain additional generalization benefits beyond common regularization techniques. *Weight decay* is a typical regularization technique that encourages model weights to have small L_2 norm as part of the training objective. Giles and Omlin [16] demonstrated that pruning can improve the generalization of recurrent neural network models better than weight decay. Thimm and Fiesler [60], Augasta and Kathirvalavakumar [1] compared the effect on the generalization of several contemporary pruning techniques. Recently, [3] discovered a positive correlation between *pruning instability*, defined as the drop in test accuracy immediately following weights removal, and model generalization. The authors found that higher pruning-induced instability leads to increased flatness of minima, which in turn improves generalization. Our work differs from prior work as we study pruning's effects on generalization through a novel perspective, examining pruning's impact on model training.

Pruning's effect beyond generalization. Our work contributes to a growing line of work [25, 38] investigating pruning's effect on examples beyond test error. Hooker et al. [25] discovered that model

compression can have a disproportionately large impact on predicting the under-represented long-tail of the data distribution. Liebenwein et al. [38] studied the performance of pruning using metrics such as out-of-distribution generalization and resilience to noise, and found that pruning may not preserve these alternative performance metrics even when it preserves test accuracy. To the best of our knowledge, our work is the first to examine pruning's impact on examples in the training set.

7 Closing Discussion

Importance of training improvement. Our work sheds light on an overlooked and underestimated effect of pruning that boosts generalization – training improvement. Seminal work on pruning [22, 19] focuses on its regularization effect and attributes pruning's beneficial influence on generalization to regularization that comes with model size reduction. Instead, our work discovers that, on standard image classification benchmarks, the state-of-the-art pruning algorithm attains the optimal generalization consistently by improving model training.

How does model size reduction impact generalization? The relationship between *model complexity* and generalization has long been a subject of immense interests [51, 23, 5, 46]. Informally, model complexity refers to the model's ability to fit a wide variety of functions [17]. For example, model size [5] and the norm of model weights [48] both provide meaningful measures of model complexity for the study of generalization. Classical theories (e.g., on VC dimension [63] and Rademacher complexity [2]) show that reducing model complexity decreases the likelihood for the learning algorithm to produce trained models with large gap between training and test loss, and may therefore improve generalization, We instead focus on learning dynamics of training example subgroups, and show that reducing model size improves generalization by mitigating the harmful effect of noisy examples on generalization. Our work therefore contributes a complementary perspective to the understanding of the relationship between model complexity and generalization.

Societal impact. Our work examines pruning's effect on generalization. While we measure generalization using test error, Hooker et al. [25] show that test error may be an imperfect characterization of model prediction quality due to a lack of consideration of fairness and equity. Concretely, our work does not address the potential disproportionate impact of model pruning on label categories. Our work must therefore be interpreted within the context of the community's current and future understanding of pruning's potential contribution to systemic bias, especially against minority groups that may be underrepresented in existing training datasets.

Implication for fairness. A growing line of work concerns the adverse impact of pruning on model fairness [25, 26, 6, 6] – they show that pruning disproportionately worsens the prediction accuracy on a small subgroup of examples. However, our results show that their work does not completely characterize pruning's effects pertinent to fairness: (1) We show that only at relatively high sparsities can pruning harm prediction accuracy of any example subgroup, where pruning's regularization effects dominate. (2) We show that size reduction in general, rather than pruning in particular, underpins the disproportionately worse accuracy of the pruned model on certain subgroups of examples. Our results call for further assessment and discussion of pruning's risk to model fairness.

Conclusion. We show that the long-standing size-reduction hypothesis attributing pruning's beneficial effect on generalization to its reduction of model size does not fully explain pruning's impact on generalization. Inspired by studies [61, 50] that show nonuniform effects of training examples on generalization, we develop an analysis to study pruning's impact on generalization by interpreting its effect on an example's training loss in relation to the example's influence on generalization.

With this novel analysis, we find that, at the optimal sparsity, pruning leads to either better training or additional regularization, which improves training loss over the dense model, and reduces the accuracy degradation due to noisy examples over the dense model, respectively. Both effects contribute to improving model generalization.

Our novel analysis adds to our empirical toolkit for studying the effect of a learning algorithm on generalization through its effect on the loss of training examples. Using our novel analysis, we derive findings that advance our empirical understanding of pruning as a learning algorithm.

References

[1] M. Augasta and T. Kathirvalavakumar. Pruning algorithms of neural networks — a comparative study. *Central European Journal of Computer Science*, 3, 2013.

[2] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3, 2003.

[3] B. Bartoldson, A. Morcos, A. Barbu, and G. Erlebacher. The generalization-stability tradeoff in neural network pruning. In *Advances in Neural Information Processing Systems*, 2020.

[4] C. Baykal, L. Liebenwein, I. Gilitschenski, D. Feldman, and D. Rus. Data-dependent coresets for compressing neural networks with applications to generalization bounds. In *International Conference on Learning Representations*, 2019.

[5] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116 (32), 2019.

[6] C. Blakeney, N. Huish, Y. Yan, and Z. Zong. Simon says: Evaluating and mitigating bias in pruned neural networks with knowledge distillation. *arXiv preprint arXiv:2106.07849*, 2021.

[7] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag. What is the state of neural network pruning? In *Conference on Machine Learning and Systems*, 2020.

[8] T. Calders, F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. In *IEEE International Conference on Data Mining Workshops*, 2009.

[9] T. Chen, B. Ji, T. Ding, B. Fang, G. Wang, Z. Zhu, L. Liang, Y. Shi, S. Yi, and X. Tu. Only train once: A one-shot neural network training and pruning framework. In *Advances in Neural Information Processing Systems*, 2021.

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

[11] X. Dong, S. Chen, and S. Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*, 2017.

[12] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

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

[14] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

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

[16] C. Giles and C. Omlin. Pruning recurrent neural networks for improved generalization performance. *IEEE Transactions on Neural Networks*, 5(5), 1994.

[17] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[18] P. D. Grünwald. *The minimum description length principle*. MIT press, 2007.

[19] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. In *Conference on Neural Information Processing Systems*, 2015.

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

[21] M. Hardt, E. Price, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Conference on Neural Information Processing Systems*, 2016.

[22] B. Hassibi, D. Stork, and G. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, 1993.

[23] T. Hastie, J. Friedman, and R. Tisbshirani. *2.9 Model Selection and Bias-Variance Tradeoff*, page 37–38. 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[25] S. Hooker, A. C. Courville, Y. N. Dauphin, and A. Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

[26] S. Hooker, N. Moorosi, G. Clark, S. Bengio, and E. Denton. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020.

[27] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and z. Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Conference on Neural Information Processing Systems*, 2019.

[28] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems*, 2012.

[30] V. Lebedev and V. Lempitsky. Fast convnets using group-wise brain damage. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[31] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In *Conference on Neural Information Processing Systems*, 1990.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[33] J. Lee, S. Park, S. Mo, S. Ahn, and J. Shin. Layer-adaptive sparsity for the magnitude-based pruning. In *International Conference on Learning Representations*, 2021.

[34] N. Lee, T. Ajanthan, and P. Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

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

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

[37] L. Liebenwein, C. Baykal, H. Lang, D. Feldman, and D. Rus. Provable filter pruning for efficient neural networks. In *International Conference on Learning Representations*, 2020.

[38] L. Liebenwein, C. Baykal, B. Carter, D. Gifford, and D. Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. In *Conference on Machine Learning and Systems*, 2021.

[39] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision*, 2017.

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

[41] C. Louizos, M. Welling, and D. P. Kingma. Learning sparse neural networks through l0 regularization. In *International Conference on Learning Representations*, 2018.

[42] D. Molchanov, A. Ashukha, and D. Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, 2017.

[43] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017.

[44] M. C. Mozer and P. Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *Conference on Neural Information Processing Systems*, 1988.

[45] V. Nagarajan and J. Z. Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Conference on Neural Information Processing Systems*, 2019.

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

[47] B. Neyshabur, R. Tomioka, and N. Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[48] B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, 2015.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Conference on Neural Information Processing Systems*, 2019.

[50] M. Paul, S. Ganguli, and G. K. Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Conference on Neural Information Processing Systems*, 2021.

[51] C. Rasmussen and Z. Ghahramani. Occam's razor. In *Conference on Neural Information Processing Systems*, 2000.

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

[53] T. Serra, A. Kumar, and S. Ramalingam. Lossless compression of deep neural networks. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 2020.

[54] T. Serra, X. Yu, A. Kumar, and S. Ramalingam. Scaling up exact neural network compression by relu stability. In *Conference on Neural Information Processing Systems*, 2021.

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

[56] S. P. Singh and D. Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. In *Conference on Neural Information Processing Systems*, 2020.

[57] L. N. Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision*, 2017.

[58] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[59] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Conference on Neural Information Processing Systems*, 2020.

[60] G. Thimm and E. Fiesler. Evaluating pruning methods. In *International Symposium on Artificial Neural Networks*, 1995.

[61] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

[62] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[63] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16, 1971.

[64] C. Wang, R. Grosse, S. Fidler, and G. Zhang. EigenDamage: Structured pruning in the Kronecker-factored eigenbasis. In *International Conference on Machine Learning*, 2019.

[65] C. Wang, G. Zhang, and R. Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020.

[66] R. Yu, A. Li, C. Chen, J. Lai, V. I. Morariu, X. Han, M. Gao, C. Lin, and L. S. Davis. NISP: pruning networks using neuron importance score propagation. *arXiv preprint arXiv:1711.05908*, 2017.

[67] X. Yu, T. Serra, S. Ramalingam, and S. Zhe. The combinatorial brain surgeon: Pruning weights that cancel one another in neural networks. In *International Conference on Machine Learning*, 2022.

[68] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3), 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section 7
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Section 7
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Appendix B
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Section 2
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix B
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Section 2
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]