# *A Winning Hand*: Compressing Deep Networks Can Improve Out-Of-Distribution Robustness

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Two crucial requirements for a successful adoption of deep learning (DL) in the wild are: (1) robustness to distributional shifts, and (2) model compactness for achieving efficiency. Unfortunately, efforts towards simultaneously achieving Out-of-Distribution (OOD) robustness and extreme model compactness without sacrificing accuracy have mostly been unsuccessful. This raises an important question: *"Is the inability to create compact, accurate, and robust deep neural networks (CARDs) fundamental?"* To answer this question, we perform a large-scale analysis for a range of popular model compression techniques which uncovers several intriguing patterns. Notably, in contrast to traditional pruning approaches (e.g., fine tuning and gradual magnitude pruning), we find that "lottery ticket-style" pruning approaches can surprisingly be used to create high performing CARDs. Specifically, we are able to create extremely compact CARDs that are dramatically more robust than their significantly larger and full-precision counterparts while matching (or beating) their test accuracy, simply by pruning and/or quantizing. To better understand these differences, we perform sensitivity analysis in the Fourier domain for CARDs trained using different data augmentation methods. Motivated by our analysis, we develop a simple domain-adaptive test-time ensembling approach (CARD-Deck) that uses a gating module to dynamically choose an appropriate CARD from the CARD-Deck for each test sample based on their spectral-similarity with test samples. By leveraging complementary frequency biases of different pruned models, the proposed approach builds a "winning hand" of CARDs that establishes a new state-of-the-art for robustness (and accuracy) on CIFAR-10-C with dramatically better memory usage than non-compressed baselines. We also present some theoretical evidences supporting our empirical findings.

## 1 Introduction

Deep Neural Networks (DNNs) have achieved unprecedented success in a wide range of applications due to their remarkably high accuracy [12]. These advances predominantly come from the significant growth in the model size due to massive over-parameterization. Furthermore, these highly overparameterized models are known to be susceptible to the out-of-distribution (OOD) shifts encountered during deployment in the wild [3]. Resource-inefficiency and OOD brittleness of state-of-the-art (SOTA) DNNs severely limits the potential applications they can make an impact on. For example, consider the real-world use case of the "Mars rover mission" that uses laser-induced breakdown spectroscopy (LIBS) to search for indicators of microbial life. It is well accepted that endowing Perseverance rover with DNNs to analyze high-dimensional and complex LIBS spectra offers a huge potential to make scientific breakthroughs [1]. However, such an effort is non-existent because: 1) as these devices are battery operated, the model has to be lightweight so it consumes less memory with reduced power consumption; and 2) the model must be able to efficiently handle domain shifts

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

in spectral data caused by dynamic environmental or sensor noise. Both of these requirements are not specific to just the planetary exploration use case but arise in virtually any resource-limited application using machine learning (ML) in the wild. The fact that existing SOTA DNNs do not satisfy *model compactness* and *OOD robustness* requirements is holding us back from leveraging the advances in DL to enable game changing scientific discoveries.

In this paper, we are driven by two questions around this crucial problem. **Q1.** *Can we show the existence of compact, accurate, and robust DNNs (`CARDs`)?* **Q2.** If yes, *can we develop efficient algorithms to design `CARDs` that match (or beat) the performance of SOTA robust-DNNs?*

There have been some recent successes in addressing each of the challenges `CARDs` present in isolation. The authors in [20, 23] developed data augmentation methods for achieving high OOD robustness without sacrificing accuracy on the clean data. The authors in [9, 7] developed pruning and quantization approaches for achieving high accuracy at extreme levels of model compression. However, efforts towards achieving model compactness, high accuracy, and OOD robustness simultaneously have mostly been unsuccessful. For example, [22] and [28] recently showed that compressed (or pruned) DNNs achieve accuracy as high as the original network's, but are far more brittle when faced with OOD data. Perhaps surprisingly, the current solution in the robust ML community to improve the OOD robustness (and accuracy) is to increase the model size (e.g., [21, 13] and [6]).

In this paper, we argue that these negative results are a byproduct of inapt compression strategies, and the inability to create `CARDs` is not fundamental (answering **Q1** in the affirmative). Specifically, to answer **Q1**, we perform a large-scale comparison by varying architectures, training methods, and pruning rates for a range of popular model compression techniques. Our empirical study uncovers several intriguing patterns. We find that in contrast to traditional pruning methods analyzed in the past studies (e.g., fine tuning [15] and gradual magnitude pruning [47]), we find that "lottery ticket-style" compression approaches [9, 37, 35, 7] can surprisingly be used to create high performing `CARDs`. Specifically, we are able to create extremely compact (i.e., sparse and/or binary) `CARDs` that are dramatically more robust compared to their significantly larger and full-precision counterparts while matching (or beating) their test accuracy. Our results are in sharp contrast to the existing observations that compression is harmful to OOD robustness. In fact, we show that compression if done properly improves OOD robustness.

To better understand these differences and why they occur, we analyze these compressed models through the lens of Fourier sensitivity analysis in the spectral domain. Our analysis highlights that the model robustness on OOD data is dependent on the interaction between which frequencies the model is bias towards and which frequencies the test-time corruption is concentrated on.

Motivated by this, we develop a simple test-time domain-adaptive ensemble approach (`CARD-Deck`) that uses a gating module to dynamically chooses an appropriate `CARD` for each test sample by maximizing their spectral-similarity w.r.t. test data. By leveraging the complementary frequency biases of different `CARDs`, the proposed test-time ensembling approach build a "winning hand" of `CARDs` that establishes a new state-of-the-art robustness (and accuracy) on the popular OOD benchmark dataset CIFAR-10-C [24] with an extremely compact ensemble (answering **Q2** in the affirmative). We also present some theoretical evidences supporting our empirical findings. Finally, we provide general guidelines on choosing an appropriate compression methods for the design and deployment of DNNs in the wild (see Appendix A).

Our main contributions are summarized as follows[1]:

- We present the first positive result on simultaneously achieving high accuracy and OOD robustness at extreme levels of model compression.
- Our study reveals an intriguing finding – "lottery ticket-style" pruned models are dramatically more robust compared to their significantly larger and full-precision counterparts.
- For a deeper understanding of these patterns, we analyze these compressed models through the lens of Fourier sensitivity analysis. This offers the much needed explanation for the apparent robustness difference between different model compression methods.
- Leveraging spectral behavior of test data and different frequency bias of models at hand, we develop a domain-adaptive ensemble `CARD-Deck` that outperforms the prior SOTA at a fraction of its memory footprint.

[1] A detailed discussion on related work is provided in Appendix B

## 2 Is the inability to create CARDs fundamental?

To motivate the scientific question of interest, we propose the following CARD hypothesis which we then test through a comprehensive empirical testing of existing model compression techniques.

**CARD Hypothesis.** *Given a sufficiently overparameterized neural network, a suitable model compression (i.e., pruning and binarization) scheme can yield a compact network with comparable (or higher) accuracy and robustness than the same network when it is trained without compression.*

### 2.1 Methodology

For a comprehensive analysis, we introduce a framework inspired by those in [37, 41] that accommodates pruning methods from traditional-through-emerging paradigms. Specifically, we define the trained subnetwork created by one training-pruning-retraining cycle as: $W_{\text{sparse}} = F_1(W; k_1, \mathcal{D}) \odot F_2(W; k_2, \mathcal{D})$, where $\mathcal{D}$ denotes the training dataset, $W$ denotes the weight vector, $F_1$ represents the function to find the masks, $F_2$ represents the function to train the weights after pruning, $k_i$ is the earliest training iteration that $F_i$ requires information from, and $\odot$ is the Hadamard (element-wise) product. Using this, the pruning paradigms considered in this paper can be specified as follows. Traditional: $k_1 = k_2$ and $k_1 > 0$; Rewinding-based Lottery Tickets: $k_1 = T$ and $k_2 = r, r < T$; and Initialization-based (Strong) Lottery Tickets: $k_1 = k_2 = 0$ and $F_2 = \mathbb{I}$ (identity).

As opposed to traditional and rewinding schemes, strong lottery ticket [35] schemes do not require any weight training after pruning—a performant network is found via $F_1$. In other words, learning occurs simply by pruning a randomly initialized neural network. As accuracy is of importance for CARDs, we consider representative techniques from these categories that have been demonstrated to deliver accuracy comparable to that of the unpruned dense model. In particular, we make use of: Fine-Tuning (FT) [15] and Gradual Magnitude Pruning (GMP) [47], which are traditional methods; Weight Rewinding (LTH) [9, 10] and Learning Rate Rewinding (LRR) [37], rewinding-based methods; and Biprop (BP) [7] and Edgepopup (EP) [35], initialization-based methods. See Appendix B.1 for a discussion of the differences between methods in a single class. In all of these methods, we make use of global unstructured pruning which allows for different pruning percentages at each layer of the network. We elected to use the hyperparameters specifically tuned for each approach by the developers of each method. Details of this can be found in Appendix C. For a reliable comparison, we make use of convolutional [9], VGG [39] and ResNet [16] style architectures of varying size and train them on CIFAR-10 [24]. In particular, we make use of the small Conv-4/6/8 models [9], VGG-19, and ResNet-18. The robustness of each model is determined by computing the Top-1 accuracy on CIFAR-10-C [19]. As a baseline, we train 5 realizations of each model without pruning. For each pruning technique, we train 5 realizations at each sparsity level from the set {50%, 60%, 80%, 90%, 95%} to cover the spectrum of low to high pruning percentages.

### 2.2 Our findings

**Accuracy-robustness comparison of global pruning methods.** In Figure 1, we plot our experimental results for different architectures. For each pruning method, we plot the mean accuracy or robustness value over all realizations relative to the mean of the dense baseline realizations with error bars extending to the minimum and maximum realization value at each of the aforementioned sparsity levels. The first row of plots in each figure corresponds to Top-1 accuracy on CIFAR-10 while the second row corresponds to Top-1 accuracy on CIFAR-10-C (both relative to the baseline dense/non-compressed models). The mean baseline accuracy and robustness for each architecture is listed as a reference accuracy in the corresponding plot.

Our results for traditional methods, i.e., Fine-Tuning and Gradual Magnitude Pruning, are consistent with previous works [22, 28] as the robustness of models pruned using these methods degrades relative to the dense models', particularly in higher pruning regimes. However, we find that rewinding and initialization based pruning approaches consistently produce dramatic gains in robustness relative to dense baselines while matching (and sometimes surpassing) the accuracy of the dense baseline. In particular, the rewinding class of methods provide a consistent, moderate improvement to both accuracy and robustness while the initialization class provides more substantial gains in robustness even when the accuracy is slightly below the baseline accuracy. The significance of overparameterization to finding highly compact CARDs using initialization methods is evident for all architecture

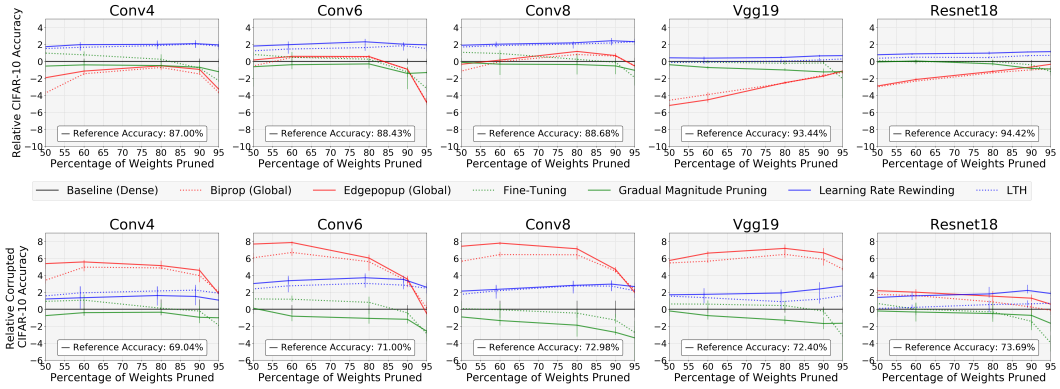


Figure 1: **Suitable pruning approaches dramatically improve robustness over dense models**: Comparing the Top-1 accuracy of pruned models relative to a dense baseline on CIFAR-10 and CIFAR-10-C demonstrates the existence of LRR, LTH, BP, and EP `CARDs`.

types, as the robustness of these models in higher pruning regimes improves at increasing levels of parameterization. However, even in models with fewer parameters, we find that both rewinding and initialization are able to provide robustness gains. Lastly, the significant robustness gains provided by EP- and BP-pruned models is a feature of initialization pruning methods and not due to weight binarization. We provide a comparison to EP using a full-precision initialization in Appendix D.

To summarize, we have empirically verified our hypothesis by demonstrating that "lottery ticket-style" pruning methods can identify `CARDs`. Further, these pruning methods are capable of providing both accuracy and robustness gains over dense models of the same architecture.

**Effect of global vs. layerwise pruning in lottery ticket initialization methods.** The lottery ticket initialization methods analyzed in the previous section were originally designed to prune a user-specified uniform percentage across all layers of the network. In contrast, global pruning methods are considered to be more flexible as they can prune some layers more heavily than others while still meeting a user-specified sparsity level for the entire network. By analyzing these initialization methods using both layerwise and global pruning, we notice certain peculiar patterns. Figure 2 provides the accuracy and robustness of models trained with BP and EP using global and layerwise pruning. While layerwise pruned models consistently achieved the maximum accuracy (across prune percentages), the globally pruned models consistently provided significant robustness gains over the layerwise models. Furthermore, the globally-pruned models typically achieve higher or comparable accuracy at higher sparsity levels, indicating that using initialization methods with global pruning can yield `CARDs` with better accuracy and robustness at the same level of compactness.

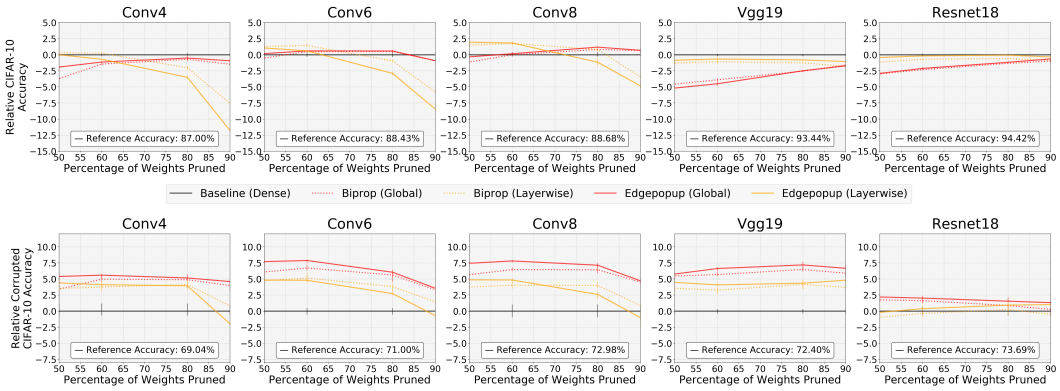


Figure 2: **Global pruning in lottery ticket Initialization methods provides greater robustness gains**: While layerwise pruning is able to achieve the highest accuracy across all sparsity levels in initialization methods, global pruning provides more significant robustness gains at all sparsity levels.

## 3 A deeper understanding of the effect of compression via a spectral lens

In order to understand what causes pruning methods to have drastically different levels of OOD robustness, we leverage the Fourier sensitivity method [43] to perform a frequency-domain analysis of models trained using various pruning techniques. First, we provide a brief summary of the Fourier sensitivity method. Given a model and a test dataset, we perturb each image in the test dataset with additive Fourier basis noise. Specifically, let $X$ be an image and $U_{i,j} \in \mathbb{R}^{d1 \times d2}$ be the 2D Fourier basis matrices. We then compute a perturbed image $X_{i,j} = X + r\varepsilon U_{i,j}$, where $r$ is chosen uniformly at random from $\{-1, 1\}$, and $\varepsilon > 0$ is the norm of the perturbation. Note that each channel of the image is perturbed independently. After applying this perturbation, we measure the test error for the model under Fourier basis noise and visualize how the test error changes as a function of frequencies $(i, j)$, thereby yielding the Fourier error heatmap of a model. The heatmap plots the models sensitivity to different frequency/orientation perturbations in the Fourier domain. The center of the heat map contains perturbations corresponding to the lowest-frequency Fourier bases (usually corresponding to clean data) and the edges correspond to the highest frequencies (high frequency noise/corruption).

We generate heatmaps for a trained model corresponding to each pruning method used in Section 2. We also vary the norm of the perturbation, $\varepsilon$, over the set $\{3, 4, 6\}$ representing low, medium, and high levels of perturbation severity. Additionally, we include heatmaps for the dense (non-compressed) baseline model as a reference. Generated heatmaps for the Conv8 architecture at 80% prune percentage are provided in Figure 3; the remaining heatmaps can be found in Section E.

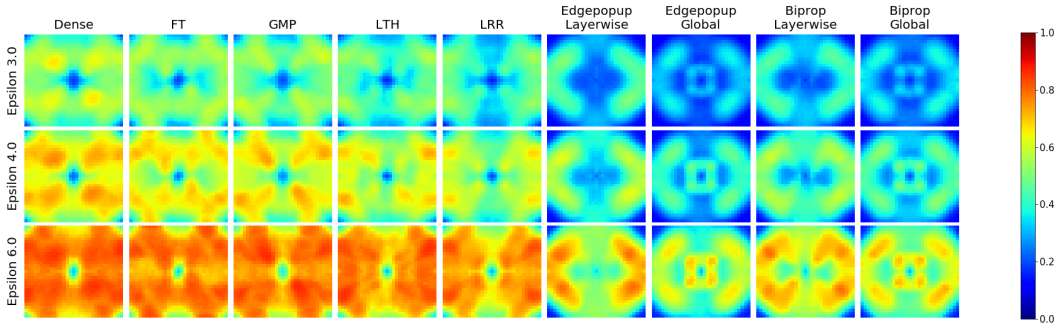


Figure 3: **Visualizing resilience of compressed models to perturbations at different frequencies**: Fourier heatmaps for error rate of Conv8 trained on CIFAR-10 with 80% pruned weights.

**Exploring the cause of robustness differences among different compression methods.** Figure 3 illustrate how rewinding and initialization pruning methods reduce the error rate across the whole frequency spectrum of perturbations when compared to the dense model. The error rate in initialization based approaches is lower than the baseline in nearly every heatmap region, aligning with the strong robustness performance of these pruning methods observed in Figure 1. Further, the results for layerwise and global heatmaps indicate that these pruning strategies result in networks that are robust to perturbations of different frequencies. In particular, by comparing the center of the heatmaps for layerwise and global methods, it follows that initialization methods leveraging layerwise pruning are more robust to low frequency perturbations, which translates to a higher accuracy on the clean data. On the other hand, global pruning in initialization methods yields `CARDs` that are less sensitive to mid to high frequency perturbations when compared to layerwise pruned `CARDs`, which translates to a higher robustness to the OOD data. Finally, traditional pruning methods yield a Fourier heatmap with a slightly altered structure when compared to the dense baseline and it slightly resembles a higher error rate version of heatmaps for rewinding methods. To investigate this, we now provide a more detailed analysis of traditional and rewinding methods.

**What separates rewinding methods from traditional pruning methods?** While LTH [9] and LRR [37] offer unsurpassed performance, such approaches also greatly extend the training duration, pruning just 20% of the remaining weights every $T - r$ epochs, where $T$ is the initial training duration and $r$ is the epoch the weights/learning-rate are rewound to after each pruning event (here, $r = 12$ and $T = 160$). This raises the question: is this longer training and multi-shot pruning procedure critical to the robustness improvements LTH/LRR offer relative to FT/GMP?

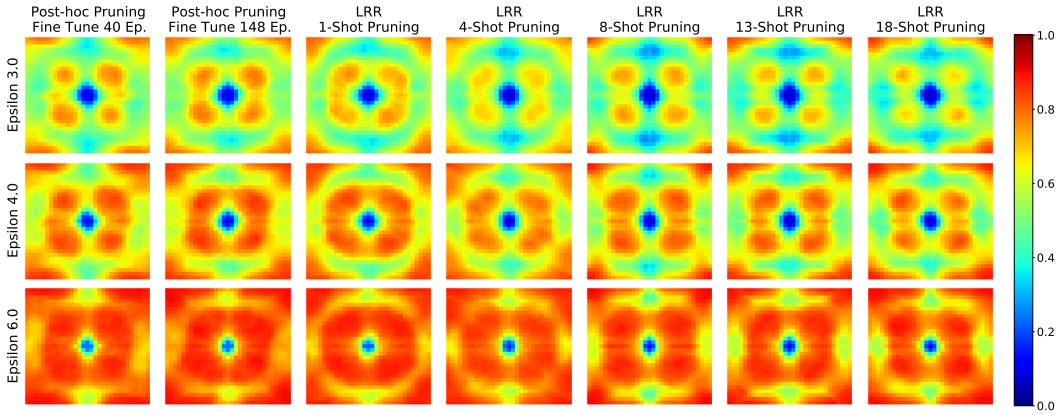


Figure 4: **Comparing resiliencies of Rewinding and Traditional methods to perturbations at different frequencies**: Fourier heatmaps for error rate of ResNet18 models trained on CIFAR-10 in which 95% of the weights are pruned. CIFAR-10-C accuracies from left to right: (70.2, 70.9, 72.6, 75.8, 75.2, 75.7, 76.2). CIFAR-10 accuracies: (94.6, 94.7, 94.5, 95.1, 95.5, 95.7, 95.6).

To test this, we gradually build the LTH/LRR pruning approach used in this paper by starting from a fine-tuning approach and adding modifications until we produce the LTH/LRR method that prunes the network 13 times to reach 95% sparsity. The progress of this construction is illustrated in Figure 4, wherein we plot a column of Fourier heatmaps for each phase. Specifically, the first column is our FT approach, the second column extends the fine-tuning duration, the third column adds learning-rate rewinding to this fine-tuning period, the fourth column decreases the iterative prune rate to achieve 95% sparsity in 4 shots rather than 1, and subsequent columns continue to increase the number of pruning shots. In this construction process, we find a notable benefit of adding learning-rate rewinding (70.9% to 72.6% CIFAR-10-C accuracy moving from column 2 to column 3), but the biggest benefits of LTH/LRR come from combining this rewinding with iteration (i.e., all columns from 4 onward display at least 75% robust accuracy). Interestingly, our results also indicate that it may be possible to achieve the robustness benefits of LTH/LRR with a higher iterative pruning rate and thus fewer pruning shots/iterations than what is standard in the literature [9, 37]. Appendix F shows that LTH displays similar trends and adds more discussion of these results.

## 4 Playing the right CARD to improve accuracy-robustness performance

Our analysis in Section 3 highlights that the robustness of the compressed model on frequency-perturbed data is crucially dependent on the interaction between the frequencies the model is biased towards and the frequencies on which the test-time corruption is concentrated. Motivated by this insight and work on data augmentation methods for biasing the model to improve robustness, we exploit the compact nature of `CARDs` to design a data-adaptive ensembling approach based on spectral similarity of test data and training data.

**A spectral-similarity metric** Let $x_{train} \in \mathbb{R}^{D_1 \times D_2 \times N}$ denote the N unaugmented training images of dimension $D_1 \times D_2$, $A = \{a_k\}_{k=1}^{m}$ denote a set of $m$ different augmentation schemes for images of dimension $D_1 \times D_2$, and $\hat{S}_{a,P}(x)$ denote a sampling of $P$ images from $x$ where augmentation $a \in A$ has been applied to $x$. Motivated by our analysis using Fourier heatmaps, we use a spectral-similarity metric to compare representatives from augmented versions of the training sets, $\{\hat{S}_{a,P}(x_{train})\}_{a \in A}$, to the test data. Our spectral-similarity metric is a map $d_{ss} : \mathbb{R}^{D_1 \times D_2 \times P} \times \mathbb{R}^{D_1 \times D_2 \times M} \to \mathbb{R}$ defined by $d_{ss}(X, Y) = \min_{1 \leq i \leq P} \|(F(X_i)/\|F(X_i)\|) - \mathbb{E}[F(Y)/\|F(Y)\|]\|$, where $F(\cdot)$ computes the 1D radially-averaged power spectrum for images of dimension $D_1 \times D_2$ then takes the reciprocal of each component. The expectation, $\mathbb{E}$, is taken over the final dimension so as to average the 1D power spectrum information for each image in $Y$. In practice, we found that the 1D power spectra for different augmentation types were more separable in the higher frequencies of the power spectrum leading to the use of the reciprocal in the definition of $F(\cdot)$. Having proposed this metric, we now leverage it to design a data-adaptive ensembling method.

| | | Baseline | | CARD | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Dense | | Edgepopup | | | | | | LRR | | | LTH | | | Biprop | | | | | | |
| | | | | Layerwise | | | Global | | | Global | | | Global | | | Layerwise | | | Global | | | |
| | | Augmix | Clean | Gauss. | Augmix | Clean | Gauss. | Augmix | Clean | Gauss. | Augmix | Clean | Gauss. | Augmix | Clean | Gauss. | Augmix | Clean | Gaussian | Augmix | Clean | Gauss. | |
| 80% | Clean Acc. | 95.5 | 95.1 | 93.9 | 94.3 | 93.7 | 92.4 | 94.9 | 94.4 | 93 | 96.1 | 95.6 | 93.8 | 95.6 | 94.9 | 93.5 | 93.9 | 93.7 | 92.4 | 94.5 | 94.1 | 93.2 |
| | Robust Acc. | 89.2 | 73.7 | 85.6 | 87.8 | 76.7 | 85.1 | 88.4 | 74.4 | 85.7 | 89.8 | 76.3 | 86.4 | 89.4 | 74.4 | 85.8 | 87.5 | 76.1 | 85.1 | 87.8 | 74.3 | 85.3 |
| | Memory (Mbits) | 358 | 358 | 358 | 2.23 | 2.23 | 2.23 | 2.23 | 2.23 | 2.23 | 71.5 | 71.5 | 71.5 | 71.5 | 71.5 | 71.5 | 2.23 | 2.23 | 2.23 | 2.23 | 2.23 | 2.23 |
| 90% | Clean Acc. | 95.5 | 95.1 | 93.9 | 94.4 | 93.9 | 92.9 | 94.4 | 94.1 | 92.8 | 96.3 | 95.6 | 93.9 | 95.7 | 95.2 | 93.6 | 94.4 | 94.1 | 92.7 | 93.7 | 93.6 | 93 |
| | Robust Acc. | 89.2 | 73.7 | 85.6 | 88 | 75.6 | 85.2 | 87.9 | 76 | 85.4 | 89.8 | 75.7 | 86.3 | 89.7 | 76.4 | 86 | 87.8 | 75.1 | 85 | 87.1 | 74.6 | 84.2 |
| | Memory (Mbits) | 358 | 358 | 358 | 1.12 | 1.12 | 1.12 | 1.12 | 1.12 | 1.12 | 35.8 | 35.8 | 35.8 | 35.8 | 35.8 | 35.8 | 1.12 | 1.12 | 1.12 | 1.12 | 1.12 | 1.12 |
| 95% | Clean Acc. | 95.5 | 95.1 | 93.9 | 94.5 | 94 | 92.6 | 93.2 | 92.7 | 91.2 | 96.1 | 95.7 | 93.9 | 95.8 | 95.1 | 93.8 | 94.2 | 93.8 | 92.5 | 92.5 | 92.1 | 91.4 |
| | Robust Acc. | 89.2 | 73.7 | 85.6 | 87.8 | 73.1 | 84.3 | 85.7 | 73.4 | 83.9 | 89.6 | 75.6 | 86.3 | 89.7 | 76.5 | 86 | 87.5 | 73.8 | 84.4 | 84.7 | 73.4 | 83.1 |
| | Memory (Mbits) | 358 | 358 | 358 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 17.9 | 17.9 | 17.9 | 17.9 | 17.9 | 17.9 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 | 0.56 |

Table 1: Performance comparison between dense baselines and CARDs. Clean and Robust Acc. refer to accuracy on CIFAR-10 and CIFAR-10-C, respectively.

**A winning hand of CARDs by test-time ensembling** We now design an ensembling approach called CARD-Deck that leverages the proposed spectral-similarity metric to further improve the performance of CARDs while maintaining a small memory footprint. As a simple baseline, we use domain-agnostic ensemble that assigns the same weight to all the CARDs in a CARD-Deck. In both ensembling techniques, the ensemble consists of CARDs that have been trained on the same dataset where different augmentation schemes are applied during training. The CARD-Deck ensemble is then comprised of $n$ CARDs given by $f^{Deck} = \{f^{a_k}\}_{k=1}^{n}$ where $a_k$ is one of the $m$ augmentation schemes from $A$ and the superscript in $f^{a_k}$ denotes that this CARD was trained used data from the distribution $S_{a_k}(x_{train})$. When a batch of $M$ test images $x_{test} \in \mathbb{R}^{D_1 \times D_2 \times M}$ is provided to the network the CARD-Deck determines how to leverage the available CARDs in the CARD-Deck to improve network performance. We provide details on the two variants of CARD-Deck next. A figure illustrating the CARD-Deck design is provided in Appendix G.

Our domain-agnostic CARD-Deck approach makes use of all CARDs in the deck to make the prediction using prediction averaging. Supposing that the output of each CARD in $f^{Deck}$ is softmax vectors, then the output of the static CARD-Deck can be expressed as $\frac{1}{n}\sum_{k=1}^{n} f^{a_k}(x_{test})$. In our domain-adaptive, we use the spectral-similarity metric $d_{ss}$ to determine which augmentation is most similar to the incoming test images. When an augmentation scheme, say $a \in A$, is identified as most similar to the incoming test data using $d_{ss}$, the CARD-Deck selects only the CARDs that were trained using data from the distribution $S_a(x_{train})$. The most similar augmenation is given by $a^* \in \arg\min_{a \in A} d_{ss}(\hat{S}_{a,P}(x_{train}), x_{test})$. If the CARD-Deck contains multiple cards trained using the same data augmentation scheme, prediction averaging is used on these CARDs and returned as the CARD-Deck prediction. As computing the spectral-similarity scheme is independent of CARD evaluation, the CARD-Deck provides reduced inference time by only evaluating the CARDs necessary for prediction. Given $a^*$ and letting $\mathcal{I}(a) = \{k : a_k = a\}$, the output of the dynamic CARD-Deck can be expressed as $\frac{1}{|\mathcal{I}(a^*)|}\sum_{k \in \mathcal{I}(a^*)} f^{a_k}(x_{test})$.

### 4.1 Experimental results

For our experiments, we selected two augmentation schemes. The first is AugMix [20] which provides improved robustness without compromising clean accuracy. The second is a Gaussian noise augmentation where $\mathcal{N}(\mu = 0,\ \sigma = 0.1)$ is independently added to all the training image pixels with probability $p = 0.5$. We trained 3 realizations of CARDs using each pruning and augmentation scheme with the ResNet-18 architecture. For each pruning scheme, we formed CARD-Decks of size 2, 4, and 6 where half of the CARDs were trained using AugMix and the other half were trained using the Gaussian augmentation scheme. When using the spectral-similarity metric in data-adaptive CARD-Decks, we used $P = 5000$ to sample images from $\hat{S}_{AugMix,P}(x_{train})$ and $\hat{S}_{Gaussian,P}(x_{train})$ to use as the representatives for the AugMix and Gaussian datasets in the similarity metric, respectively. At test time, $M = 100$ was the number of corrupted test images used in the spectral-similarity metric to determine which augmentation representatives best represented the corrupted test data. Results for the average clean and robust accuracy across realizations and memory usage for CARDs of each

| | | Baseline | | | CARD-Deck (Agnostic/Adaptive) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Dense | | | LRR | | | LTH | | | Biprop (Global) | | |
| | | Augmix | Clean | Gaussian | 2 | 4 | 6 | 2 | 4 | 6 | 2 | 4 | 6 |
| 80% | Clean Acc. | 95.5 | 95.1 | 93.9 | 96/96 | 96.3/96.4 | 96.4/96.6 | 95.5/95.5 | 96/96.1 | 96.1/96.1 | 93.8/93.8 | 94.3/94.3 | 94.3/94.4 |
| | Robust Acc. | 89.2 | 73.7 | 85.6 | 89.7/90.9 | 91.7/91.8 | 91.9/92 | 89.4/90.4 | 91/91.2 | 91.2/91.4 | 87.5/88.6 | 88.8/89.3 | 89/89.6 |
| | Memory | 358 Mb | 358 Mb | 358 Mb | 143 Mb | 286 Mb | 429 Mb | 143 Mb | 286 Mb | 429 Mb | 4.47 Mb | 8.94 Mb | 13.4 Mb |
| 90% | Clean Acc. | 95.5 | 95.1 | 93.9 | 96.3/96.3 | 96.4/96.4 | 96.4/96.6 | 95.7/95.7 | 95.9/95.7 | 96.2/96.2 | 94/94 | 94.6/94.4 | 94.5/94.6 |
| | Robust Acc. | 89.2 | 73.7 | 85.6 | 89.8/91.1 | 91.7/91.8 | 92/92.1 | 89.4/90.5 | 91/91.3 | 91.3/91.5 | 87.4/88.6 | 89.2/89.5 | 89.4/89.9 |
| | Memory | 358 Mb | 358 Mb | 358 Mb | 71.5 Mb | 143 Mb | 215 Mb | 71.5 Mb | 143 Mb | 215 Mb | 2.23 Mb | 4.47 Mb | 6.70 Mb |
| 95% | Clean Acc. | 95.5 | 95.1 | 93.9 | 96.1/96.1 | 96.3/96.4 | 96.3/96.5 | 95.8/95.8 | 96/96.1 | 96.1/96.2 | 94/94 | 94.7/94.5 | 94.7/94.6 |
| | Robust Acc. | 89.2 | 73.7 | 85.6 | 89.6/91 | 91.6/91.8 | 91.9/92 | 89/90.3 | 90.8/91.1 | 91.1/91.4 | 87.2/88.5 | 88.9/89.4 | 89.2/89.7 |
| | Memory | 358 Mb | 358 Mb | 358 Mb | 35.8 Mb | 71.5 Mb | 107 Mb | 35.8 Mb | 71.5 Mb | 107 Mb | 1.12 Mb | 2.23 Mb | 3.35 Mb |

Table 2: Performance gains of domain-agnostic and domain-adaptive CARD-Decks over dense baselines and CARDs. Clean and Robust Acc. refer CIFAR-10 and CIFAR-10-C accuracy, respectively.

pruning scheme are provided in Table 1. Performance of the domain-agnostic and domain-adaptive 2-, 4-, and 6-CARD-Decks are provided in Table 2. We find that both the data-agnostic and adaptive methods are capable of significantly improving the performance and are comparable to the current state-of-the-art performance on CIFAR-10-C which requires the use of ResNet-50. In comparison, our best performing CARD-Deck makes use of 6 LRR CARDs yielding a clean and robust accuracy of 96.4 and 92.1. Additional results are provided in Appendix G.

# 5 Theoretical justifications

## 5.1 Function approximation view of CARDs

By leveraging existing theoretical analyses of the Strong and Multi-Prize Lottery Ticket Hypotheses [34, 33, 7], we can provide theoretical guarantees on the existence and approximation potential of CARDs under certain assumptions. Note that this result does not guarantee the existence of CARDs within the size network as the dense network even though we were able to empirically demonstrate such CARDs exist. We state an informal version of this result for binary-weighted CARDs here. We note that this can be leveraged to establish approximation capabilities for CARD-Decks, which can be found in Appendix H.

**Theorem 1.** *(Informal Statement of CARD Approximation Theorem) Given a non-compressed network $F$ with depth $\ell$ and width $w$ with bounded weights that achieves a desired target accuracy and robustness, a random binary network of depth $2\ell$ and width $O\left((\ell w^{3/2}/\varepsilon) + \ell w \log(\ell w/\delta)\right)$ contains with probability $(1-\delta)$ a binary-weighted CARD that approximates the target non-compressed network with error at most $\varepsilon$, for any $\varepsilon, \delta > 0$.*

We can naturally extend the approximation result provided for CARDs in Theorem 1 to establish the effectiveness of CARD-Deck ensembles. In particular, we can establish that CARD-Decks provide $\varepsilon$-approximations of ensembles comprised of only non-compressed networks thereby achieving comparable accuracy and robustness to methods that do not leverage compression.

## 5.2 Robustness analysis of CARD-Deck

To provide the theoretical justification behind our CARD-Deck approach over a single classifier, we first define a robustness measure for a given classifier ensemble trained on a set of augmentations w.r.t. a corruption set encountered at the test-time. We assume that each test sample may encounter a specific corruption type from a given OOD set and be transformed to a corrupted test sample.

**Definition 1** (Average OOD Robustness)**.** *We define average robustness for a set of classifiers $f^{Deck} = \{f^a\}_{a=1}^{|A|}$ trained on augmentation type $a \in A$ on a distribution $\mathcal{D}$ and corruption type $c \in C$ as follows: $Rob(\mathcal{D}, C, f^{ens}) = \sum_{a=1}^{|A|} \sum_{c=1}^{|C|} Rob(\mathcal{D}|c, f^a) Pr(c, f^a)$, where $Rob(\mathcal{D}|c, f^a) = \mathbb{E}_{\{x,y\}\sim\mathcal{D}}\left[\inf_{f^a(x_c)\neq y} d(x_c, x)\right]$ with $x_c$ being a corrupted version of $x \sim \mathcal{D}$ and $d$ corresponds to a distance metric.*

This definition refers to the expectation of the distance to the closest misclassified corrupted sample for a given test sample. Having this definition, we can next explain the goal of our theoretical analysis

and information statement of our results. A formal statement with corresponding proof is provided in Appendix H.

Let us assume $f^a$ is learnt using a learning algorithm $L$ using the sampled augmented data $S_a$, thus, we have $f^{ens} = \{f^a = L(S_a)\}_{a=1}^{|A|}$. Let us denote by $\hat{S}_a$ an empirical distribution sampled from $S_a$. We want to provide a lower bound on the following quantity

$$\mathop{\mathbb{E}}_{\{S_a \sim D_a\}_{a=1}^{|A|}} Rob(\mathcal{D}, C, f^{ens}), \tag{1}$$

and show a better OOD robustness w.r.t. the case where we use just use a single classifier $f^a$. To understand this quantity better, using the linearity of expectation and triangle inequality, we derive a decomposition that can bound average OOD robustness below in terms of the average over the following three error terms for a classifier-corruption pair weighted by their gating (or selection) probability: 1) empirical robustness, 2) generalization penalty, and 3) distributional-shift penalty. This inequality suggests that in order to bound the average OOD robustness, we need to bound both the generalization penalty and the distributional-shift penalty. Next, we provide a bound on the distribution-shift penalty that is independent of the classifier $f^a$ and is only related to the closeness of the augmented data distribution and corrupted data distribution. Specifically, the closeness is defined in terms of Wasserstein distance as adopted in [38] but in a different setting. In summary, we prove that the difference between the robustness of a classifier trained with a specific augmentation method (e.g., Augmix or Gaussian) on test data transformed with a specific corruption type is upper bounded by the conditional Wasserstein distance between these datasets. An adaptive ensemble with a suitable gating function can achieve a minimum conditional Wasserstein distance (or best achievable OOD robustness) over the augmentation-corruption pair.

# 6 Limitations and future directions

In this paper, we showed that model compression and high robustness (and accuracy) are not necessarily conflicting objectives. In fact, we found that compression if done properly (e.g., using "lottery ticket-style" objectives) can dramatically improve OOD robustness compared to a non-compressed model. Leveraging this finding, we proposed a simple domain-adaptive ensemble of compressed models that outperformed existing SOTA in terms of accuracy and OOD robustness (at a fraction of the original memory usage).

Our results are consistent with past results in that we also show that use of test accuracy alone to evaluate the quality/deployability of a compressed model in the wild is not sufficient—one needs to adopt harder metrics such as OOD robustness. However, as opposed to the existing works in this direction, we present a construction that satisfies these "harder" requirements.

There are still many interesting questions that remain to be explored. First, spectral behavior of train and test data (as considered in this work) is not the only interaction determining the compressed model performance. It will be worthwhile to take a more holistic approach that also takes spectral behavior of the compressed model into account when building "a winning hand". Note that some analytical methodologies used in this paper are certainly exploitable for studying this. Next, our empirical study, though comprehensive in other aspects, was limited to CIFAR-10. We plan to extend this study to more complex datasets in the near future. We only derived an upper bound on the amount of overparameterization needed to approximate a target dense network in our current theoretical analysis; it will also be interesting to explore a lower bound on the same that may indicate scenarios where the proposed approach will not hold (e.g., underparameterized NNs). Also, "lottery ticket-style" networks in theory can be found more efficiently, which was not the focus of this work but is a very valuable direction. Further, achieving the theoretical memory savings obtained from `CARDs` would require their implementation on specialized hardware, e.g., FPGAs. Finally, our technique is compatible with most existing robust training methods and is expected to perform even better if we use bigger models, more advanced and end-to-end robust training with additional augmentation types. This was not considered in the paper as our primary goal was not to push SOTA but to show the existence and some initial constructions of `CARDs`.

We hope that our results, together with our analysis and new algorithmic tools, will help researchers better understand the limits of compressed neural networks and motivate future work in the direction of developing high performing `CARDs` and `CARD-Decks`, and their applications to impactful areas where DL struggles currently due to its parameter inefficiency and OOD brittleness.

# References


[1] Kshitij Bhardwaj and Maya Gokhale. Semi-supervised on-device neural network adaptation for remote and portable laser-induced breakdown spectroscopy. *arXiv preprint arXiv:2104.03439*, 2021.

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

[3] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

[4] Dan A Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, Andras Gyorgy, Timothy Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *arXiv preprint arXiv:2104.01086*, 2021.

[5] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

[6] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[7] James Diffenderfer and Bhavya Kailkhura. Multi-prize lottery ticket hypothesis: Finding accurate binary neural networks by pruning a randomly weighted network, 2021.

[8] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

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

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

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

[12] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[13] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[14] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[15] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[17] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[18] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018.

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

[20] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

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

[22] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

[23] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. *arXiv preprint arXiv:2103.02325*, 2021.

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

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

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

[27] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[28] Lucas Liebenwein, Cenk Baykal, Brandon Carter, David Gifford, and Daniela Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *arXiv preprint arXiv:2103.03014*, 2021.

[29] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *Advances in Neural Information Processing Systems*, pages 345–353, 2017.

[30] Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through $l_0$ regularization. *arXiv preprint arXiv:1712.01312*, 2017.

[31] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507. PMLR, 2017.

[32] Sharan Narang, Erich Elsen, Gregory Diamos, and Shubho Sengupta. Exploring sparsity in recurrent neural networks. *arXiv preprint arXiv:1704.05119*, 2017.

[33] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need, 2020.

[34] Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. Optimal lottery tickets via subsetsum: Logarithmic over-parameterization is sufficient, 2021.

[35] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11893–11902, 2020.

[36] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[37] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning, 2020.

[38] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Improving adversarial robustness using proxy distributions. *arXiv preprint arXiv:2104.09425*, 2021.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[40] Mingxing Tan and Quoc V Le. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021.

[41] Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. Emerging paradigms of neural network pruning. *arXiv preprint arXiv:2103.06460*, 2021.

[42] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *arXiv preprint arXiv:1608.03665*, 2016.

[43] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[44] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[45] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv preprint arXiv:1702.03044*, 2017.

[46] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[47] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017.

# Checklist

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] The discussion is included in Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work focuses on two well-defined scientific problems (model compression and out-of-distribution). Thus, we do not expect negative societal impacts from this work.
   (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? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
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)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The training details are reported in Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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] The computation resource details are reported in Appendix C.
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] We cited the datasets and assets.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (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]