DYNAMIC SPARSE TRAINING VERSUS DENSE TRAINING: THE UNEXPECTED WINNER IN IMAGE CORRUPTION ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

It is generally perceived that Dynamic Sparse Training opens the door to a new era of scalability and efficiency for artificial neural networks at, perhaps, some costs in accuracy performance for the classification task. At the same time, Dense Training is widely accepted as being the "de facto" approach to train artificial neural networks if one would like to maximize their robustness against image corruption. In this paper, we question this general practice. Consequently, *we claim that*, contrary to what is commonly thought, the *Dynamic Sparse Training* methods can consistently *outperform Dense Training* in terms of *robustness accuracy*, particularly if the efficiency aspect is not considered as a main objective (i.e., sparsity levels between 10% and up to 50%), without adding (or even reducing) resource cost. We validate our claim on two types of data, images and videos, using several traditional and modern deep learning architectures for computer vision and three widely studied Dynamic Sparse Training algorithms. Our findings reveal a new yet-unknown benefit of Dynamic Sparse Training and open new possibilities in improving deep learning robustness beyond the current state of the art.

1 INTRODUCTION

If one would like to maximize as much as possible the robustness of a deep learning model on noisy more realistic data (e.g., image corruption, out of distribution), what would be the typical approach? Would one train directly a dense neural network (referred further as Dense Training) or a sparse neural network (referred further as Sparse Training)? Of course, there are many techniques to enhance the model robustness, but still they would be applied on top of a basic optimization process which usually requires training an artificial neural network. In order to give an answer to the previously posed question, we would need to quantify the research and practitioner community perception, which is a hard problem in itself. Yet, given the large body of literature on the topic, it is safely to assume that the usual answer would be "Yes, Dense Training is the typical solution to maximize the model robustness.". Nevertheless, the reader can answer to this question by themselves, while further in this manuscript we would like to bring more clarity and some new perspectives on this topic.

With respect to Dense Training, recent studies have demonstrated that over-parameterized neural networks, can achieve impressive generalizability on test data even without explicit regularizers Zhang et al. (2021). This phenomenon is partly explained by the implicit regularization[1] induced by the optimization algorithm used during training, such as stochastic gradient descent (SGD) Zhang et al. (2021); Keskar et al. (2017); Li & Liang (2018). Despite ongoing efforts to understand the robustness behavior of deep neural network models, researchers always favor larger models, trained densely, due to their perceived robustness to unexpected variations in input data, such as additional noise, contrast changes, and weather conditions, to a certain extent Hendrycks & Dietterich (2019); Hendrycks et al. (2021). However, larger over-parameterized models do not necessarily gain more robustness Algan & Ulusoy (2021); Wang et al. (2023).

[1] In older neural network literature, regularization referred specifically to *a penalty term added to the loss function to prevent overfitting*. Recently, as in Goodfellow et al. (2016), regularization is broadly defined as "*any modification to a learning algorithm intended to reduce test error without increasing training error.*"

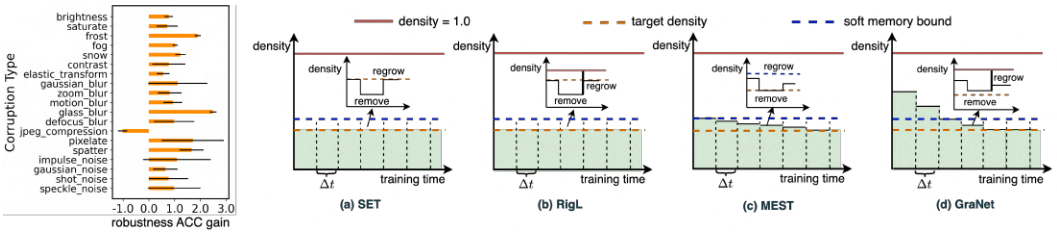


Figure 1: (Left) Robustness accuracy gain (%) in a sparsified ResNet34 trained with SET (sparsity ratio=0.5) compared to its dense counterpart, tested on CIFAR100-C with various corruption types shown on the Y-axis. Positive values indicate better performances of SET method. (Right) conceptual representation of model density during training across different DST algorithms: (a) SET, (b) RigL (Note: In the regrow step, RigL requires full gradient calculations.), (c) MEST and (d) GraNet.

With respect to sparse training, there are static sparse training Liu & Wang (2023) and Dynamic Sparse Training (DST) Mocanu et al. (2018); Bellec et al. (2018); Mostafa & Wang (2019); Evci et al. (2020); Yuan et al. (2021); Liu et al. (2021a), which maintain just a fraction of the parameters' budget throughout the entire training process. Static sparse training involves sparsifying the network before training and keeps the topology fixed during training. DST, a recent and efficient sparse-to-sparse training approach, starts with a sparse neural network and simultaneously optimizes both its weight values and connectivity. This optimization process is known as dynamic sparsity. Consequently, it is mainly seen as an approach that can maximize deep learning scalability and efficiency, at least in theory, as in practice, it is a difficult research problem to fully utilize sparsity during training Nowak et al. (2023). Nevertheless, it is often overlooked that DST frequently achieves performance comparable to, or even surpassing, that of its dense or static sparse counterparts Chen et al. (2022b); Xiao et al. (2022); Graesser et al. (2022); Sokar et al. (2022); Atashgahi et al. (2022); Liu et al. (2023); Sokar et al. (2023).

By connecting the dots, an intuitive research question arises: *Is somehow DST implicitly improving model robustness, and no one knew about this yet?*

Although several studies have investigated the impact of pruning model from Dense Training, lottery ticket hypothesis methods Frankle & Carbin (2019); Frankle et al. (2020); Renda et al. (2020) on image corruption robustness Liebenwein et al. (2021); Diffenderfer et al. (2021) and adversarial robustness Liao et al. (2022), other research shows that static sparsified networks, when increased width and depth to maintain capacity, can also improve robustness Timpl et al. (2022). However, a closer look at current DST research reveals a literature gap: the overlooked evaluation of DST models under various unexpected image corruptions. In an attempt to respond to this, we examined one of the classical DST algorithms, Sparse Evolutionary Training (SET) Mocanu et al. (2018), as illustrated in Figure 1 (Left). We find that SET (with a sparsity ratio of 0.5) can markedly improve model robustness against 18 out of 19 image corruptions, compared to Dense Training. In light of the strong performance of the DST method (e.g., SET) against common corruptions , we propose the Dynamic Sparsity Corruption Robustness (DSCR) Hypothesis.

DSCR Hypothesis. *Dynamic Sparse Training at low sparsity levels helps to improve model robustness against image corruption in comparison with Dense Training.*

We summarize our main contributions as follows:

- This study questions the typical approach of using Dense Training to maximize model robustness and proposes Dynamic Sparse Training as a more capable alternative.
- Formally, we propose the DSCR hypothesis which is validated in a wide range of corruption scenarios (i.e., corruption type and severity), using different types of data (i.e., image and video), DST algorithms (i.e., SET, RigL, MEST and GraNet) and deep learning architectures (i.e., VGG, ResNet, EfficientNet, DeiT and 3D ConvNet) (Section 4).
- We give insights, from both spatial and spectral perspectives, on the underlying factors that explain the superior performance of DST. Our analysis shows that the dynamic sparsity process in DST acts as an implicit regularization mechanism, enabling the model to automatically focus on more important features (e.g., low-frequency components) while reducing its attention to less important ones (e.g., high-frequency components) (Section 5).

2 Related Work

2.1 Relevant methods for corruption robustness

Deep neural networks (DNNs) are susceptible to unseen changes in their inputs, a scenario commonly encountered in practical applications, such as common corruptions like Gaussian noise, defocus blur, etc Hendrycks & Dietterich (2019). To enhance robustness against these common corruptions, many methods have been developed, primarily focusing on data augmentation, for example, AugMix Hendrycks* et al. (2020), AugMax Wang et al. (2021), PRIME Modas et al. (2022). These approaches aim to reduce the distribution gap between the training and testing datasets. Alternatively, other strategies to enhance corruption robustness concentrate on developing resilient learning approaches, such as contrastive learning Chen et al. (2020), or on designing specialized network components, like the push-pull layer Strisciuglio et al. (2020). Recent research has revealed that LTH-based sparse models can provide enhanced robustness to common corruptions compared to their denser counterparts Diffenderfer et al. (2021); Chen et al. (2022a). Additionally, Bair et al. (2024) introduces an adaptive sharpness-aware pruning method, which regularizes the model towards flatter regions, thereby improving robustness against image corruption. However, these methods are typically built on dense models either at the start or throughout training, often overlooking the aspect of resource-aware training.

2.2 Dynamic Sparse Training

Dynamic Sparse Training (DST) Mocanu et al. (2018); Bellec et al. (2018); Mostafa & Wang (2019); Evci et al. (2020); Chen et al. (2021); Yuan et al. (2021); Liu et al. (2021a), as a sparse-to-sparse training paradigm, starts from scratch and maintains partial parameters throughout training, providing a promising approach to significantly improve training and inference efficiency while still preserving accuracy. Recent research has focused on deciphering the mechanisms of DST Liu et al. (2021c); Evci et al. (2022; 2019), and has broadened its application in various domains, such as continuous learning Sokar et al. (2023), reinforcement learning Graesser et al. (2022); Tan et al. (2023), feature selection Sokar et al. (2022); Atashgahi et al. (2022), time series classification Xiao et al. (2022), network architecture design Liu et al. (2023), and vision transformers Chen et al. (2021).

Moreover, recent studies have concentrated on developing improved DST algorithms Dettmers & Zettlemoyer (2019); Mostafa & Wang (2019); Yuan et al. (2021); Jayakumar et al. (2020); Liu et al. (2021b). For instance, Dettmers & Zettlemoyer (2019) introduced a technique to regrow weights based on the momentum of existing weights, while Yuan et al. (2021) proposed a method that utilizes both weight and gradient information for pruning. More recently, it has been demonstrated that DST can enhance the performance of DNNs on biased data Zhao et al. (2023) and adversarial samples Chen et al. (2022b). However, the majority of DST research relies on pristine benchmark datasets, leaving the robustness of DST models against image corruption relatively unexplored. This paper aims to systematically investigate the robustness behavior of DST when exposed to various image corruptions.

3 Preliminaries

3.1 Problem Definition

Given the i.i.d. training samples $\{(\boldsymbol{x}_i^{\text{tr}}, y_i^{\text{tr}})\}_{i=1}^{n_{\text{tr}}}$ from a training distribution $p_{\text{tr}}(\boldsymbol{x}, y)$, our goal is to predict the labels $\hat{y}_i^{\text{te}}$ of test examples $\{(\boldsymbol{x}_i^{\text{te}}, y_i^{\text{te}})\}_{i=1}^{n_{\text{te}}}$ drawn from a test distribution $p_{\text{te}}(\boldsymbol{x}, y)$. Here i identifies each individual sample in both the training and test sets. We decompose $p_{\text{tr}}(\boldsymbol{x}, y)$ and $p_{\text{te}}(\boldsymbol{x}, y)$ into the marginal distribution and the conditional probability distribution, that is:

$$p_{\text{tr}}(\boldsymbol{x}, y) = p_{\text{tr}}(\boldsymbol{x}) p_{\text{tr}}(y \mid \boldsymbol{x}),$$

$$p_{\text{te}}(\boldsymbol{x}, y) = p_{\text{te}}(\boldsymbol{x}) p_{\text{te}}(y \mid \boldsymbol{x}).$$

$p_{\text{tr}}(\boldsymbol{x}) \neq p_{\text{te}}(\boldsymbol{x})$, and $p_{\text{tr}}(y \mid \boldsymbol{x}) = p_{\text{te}}(y \mid \boldsymbol{x})$, in our setting. To evaluate the performance of the models in the aforementioned problem setting, we utilize standard image benchmark datasets such as CIFAR10 Krizhevsky et al. (2009), CIFAR100 Krizhevsky et al. (2009), TinyImageNet Le & Yang (2015) and ImageNet Deng et al. (2009) for training. Then, we evaluate the performance

(robustness accuracy) of the models on the corresponding corruption benchmark datasets, including CIFAR10-C, CIFAR100-C, TinyImageNet-C and ImageNet- C/$\bar{\text{C}}$/3DCC. More details of datasets can be found in Appendix A.2.

3.2 Dynamic Sparse Training

Consider a dense neural network $f_{\boldsymbol{\theta}}$, the sparsification of $f_{\boldsymbol{\theta}}$ can be formulated as: $f_{\boldsymbol{\theta}\odot\mathbf{M}} \mapsto f_{\boldsymbol{\theta}'}$, where $\mathbf{M}$ is a set of binary masks and $\odot$ denotes element-wise multiplication. The sparsity ratio S of the resulting subnetwork, $f_{\boldsymbol{\theta}'}$, is defined as $S = 1 - \frac{\|\boldsymbol{\theta}'\|_0}{\|\boldsymbol{\theta}\|_0}$, where $\|\cdot\|_0$ represents the L_0 norm, counting the number of non-zero elements. The target memory budget (or density) ratio b is given by $\frac{\|\boldsymbol{\theta}'\|_0}{\|\boldsymbol{\theta}\|_0}$. The topology of the sparse network is dynamically updated during training every Δt iterations through a parameters remove-regrow process. Specifically, connections with low importance are removed, and new connections are regrown based on criteria like gradient information Evci et al. (2020) or randomly Mocanu et al. (2018). Dynamic sparse models are the resulting models from DST. In our study, we explore various DST approaches as follows:

- DST with **fixed memory budget in forward**: To the best of our knowledge, SET is pioneering work in DST. SET follows a magnitude-based removal and random regrow strategy, maintaining a fixed the memory budget, that is $\frac{\|\boldsymbol{\theta}'\|_0}{\|\boldsymbol{\theta}\|_0} \equiv b$, throughout the entire training process, as shown in Figure 1 (right (a)). RigL Evci et al. (2020) regrows weights based on the gradient information of all weights, while maintaining a constant memory budget during the forward phase of training, as shown in Figure 1 (right (b)). In both approaches, the mask $\mathbf{M}$ is updated every Δt iterations.
- DST with **soft memory budget in forward**: Memory-Economic Sparse Training (MEST) Yuan et al. (2021), with Soft Memory Bound Elastic Mutation, starts with a sparse model having a memory budget of $\frac{\|\boldsymbol{\theta}'\|_0}{\|\boldsymbol{\theta}\|_0} = b + b_s$, where $b_s > 0$ represents the soft memory bound that decays during training. Subsequently, at intervals of Δt, weights are removed based on magnitude and gradient information, reducing the memory budget to b, and then regrown it to $b + b_s$, as illustrated in Figure 1 (right (c)). The soft memory constraint b_s, gradually decays to zero, allowing the model to reach the target memory budget b by the end of training.
- DST with **flexible memory budget in forward**: GraNet Liu et al. (2021a) starts with a denser neural network (i.e., $\frac{\|\boldsymbol{\theta}'\|_0}{\|\boldsymbol{\theta}\|_0} < 1$). During training, at intervals of Δt, it removes weights based on magnitude and regrows connections using gradient information, similar to RigL. The main difference is that fewer connections are regrown than those removed. This process dynamically updates the mask $\mathbf{M}$ throughout training, gradually reaching the target memory budget ratio b by the end, as shown in Figure 1 (right (d)).

In this paper, for simplicity in analysis, we use a unified magnitude-based removal approach and implement either random ($(\cdot)_r$) or gradient-based (i.e., re-growing the weights with the highest magnitude gradients)($(\cdot)_g$) regrow strategy across different DST methods. Unlike their original papers Yuan et al. (2021); Liu et al. (2021a), where MEST utilized random regrow and GraNet employed gradient-based regrow, we explore both strategies for each model. Thus, for random regrow DST, our analysis includes SET, MEST$_r$, and GraNet$_r$, while for gradient-based regrow DST, we examine RigL, MEST$_g$, and GraNet$_g$. This approach allows us to consistently evaluate the robustness of DST models under varying memory budget constraints, such as fixed, soft, or flexible, during training.

4 Can DST Improve Robustness Against Image Corruption?

In this section, we aim to carry out extensive experiments to verify the DSCR Hypothesis across a wide variety of DST methods, models and data. To this end, we conduct a comprehensive set of experiments to evaluate the robustness accuracy of DST models with different sparsity ratios in the presence of image corruption.

To comprehensively validate the DSCR Hypothesis, our study embarks on an extensive analysis evaluating the robustness behavior of sparsely trained models. We train using several representative DST algorithms, including SET, Rigging the Lottery (RigL) Evci et al. (2020), Memory-Economic Sparse Training (MEST) Yuan et al. (2021), and Gradual Pruning with zero-cost Neuroregeneration (GraNet) Liu et al. (2021a), across a wide range of model architectures (i.e., VGG Simonyan & Zisserman (2015), 2D/3D ResNet He et al. (2016), EfficientNet Tan & Le (2019), DeiT Touvron et al. (2021) and Two-Stream Inflated 3D ConvNet (I3D) Carreira & Zisserman (2017)).

Our analysis spans a range of image corruption benchmarks—CIFAR10-C Hendrycks & Dietterich (2019), CIFAR100-C Hendrycks & Dietterich (2019), TinyImageNet-C Hendrycks & Dietterich (2019), ImageNet-C Hendrycks & Dietterich (2019), ImageNet-$\bar{\text{C}}$ Mintun et al. (2021), ImageNet-3DCC Kar et al. (2022) and corrupted UCF101 Soomro et al. (2012) —covering 19 types of image corruptions for CIFAR10/100-C, 15 types for TinyImageNet-C and ImageNet-C, 10 types for ImageNet-$\bar{\text{C}}$ and 11 types for ImageNet-3DCC, with each corruption tested across five severity levels. The corrupted UCF101 dataset includes 16 corruption types with four severity levels. This analysis provides a comprehensive perspective on the robustness evaluation of sparsely trained models. Detailed dataset descriptions and implementation setup can be found in Appendix A.1 and A.3.1.

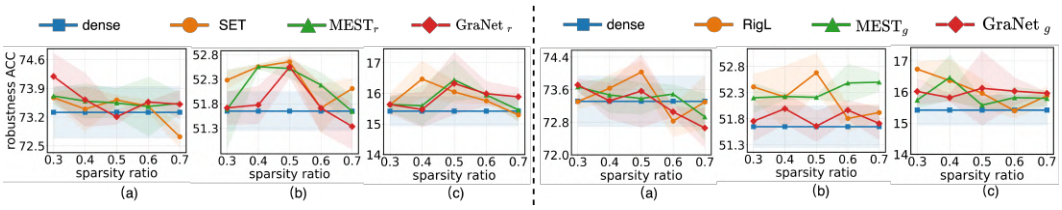


Figure 2: Robustness accuracy (%) for (a) VGG16 on CIFAR10-C, (b) ResNet34 on CIFAR100-C and (c) EfficientNet-B0 on TinyImageNet-C, comparing different DST algorithms with dense training. Left: Results for DST algorithms with random regrow strategy, including SET, MEST$_r$, and GraNet$_r$. Right: Results for DST algorithms with gradient-based regrow strategy, such as RigL, MEST$_g$, and GraNet$_g$.

4.1 Overall Performance

Figure 2 showcases the average robust accuracy across various types of corruption, with severity levels ranging from 1 to 5, on the CIFAR10/100-C and TinyImageNet-C datasets. We can find that DST models outperform their Dense training counterparts (as indicated by the blue line in Figure 2) in terms of robustness accuracy across multiple sparsity ratios. This trend is evident in both the random regrow (Figure 2, left) and gradient-based regrow (Figure 2, right) strategies. Specifically, for CIFAR100-C and TinyImageNet- C, DST models with sparsity ratios ranging from 0.3 to 0.7 generally show encouraging improvements in robustness. For CIFAR10-C, at certain sparsity ratios, such as 0.4, the overall DST methods achieve decent robust performance against image corruptions. Moreover, based on our experimental setup, at least 40% of both training computational and memory costs can be saved while simultaneously enhancing corruption robustness. Details on FLOPs and parameter counts are available in the Appendix A.3.2. This finding substantiates our DSCR Hypothesis across different DST methods, varying sparsity ratios, and architectures (i.e., VGG, ResNet, and EfficientNet).

4.2 Performance on Corruption Types

We further evaluate the DST methods across various corruption types with different severities. Figure 3 provides detailed relative robustness gains [2] for each type of corruption in ImageNet-C and ImageNet-$\bar{\text{C}}$ Mintun et al. (2021), compared to their dense model counterparts. It is worth mentioning that the X-axis in Figure 3 represents different types of corruption, which are ordered according to the amount of high-frequency information they contain, as in Saikia et al. (2021). Intriguingly, on ImageNet-C and ImageNet-$\bar{\text{C}}$ [3], DST models show significant potential in enhancing robustness

[2] Defined as $(\texttt{Acc}_{\texttt{s,sl}} - \texttt{Acc}_{\texttt{d,sl}})/\texttt{Acc}_{\texttt{d,sl}}$, where $\texttt{Acc}_{\texttt{s,sl}}$ and $\texttt{Acc}_{\texttt{d,sl}}$ represent the accuracies of models trained using DST and dense training, respectively, at a severity level of $\texttt{sl}$.

[3] Our experiments primarily focus on DST with a gradient-based regrow strategy when training on ImageNet. As our implementation is based on PyTorch, the gradient-based regrow approach tends to achieve

against specific corruptions, such as impulse, Gaussian, and shot noise on ImageNet-C, as well as brownish, Perlin, plasma, and blue noise on ImageNet-$\bar{\text{C}}$. It is worth noting that these corruptions are typically characterized by relatively high-frequency information.

This finding is consistent on CIFAR10-C, CIFAR100-C, and TinyImageNet-C, and due to page limitations, the results for these datasets are detailed in Appendix A.4.

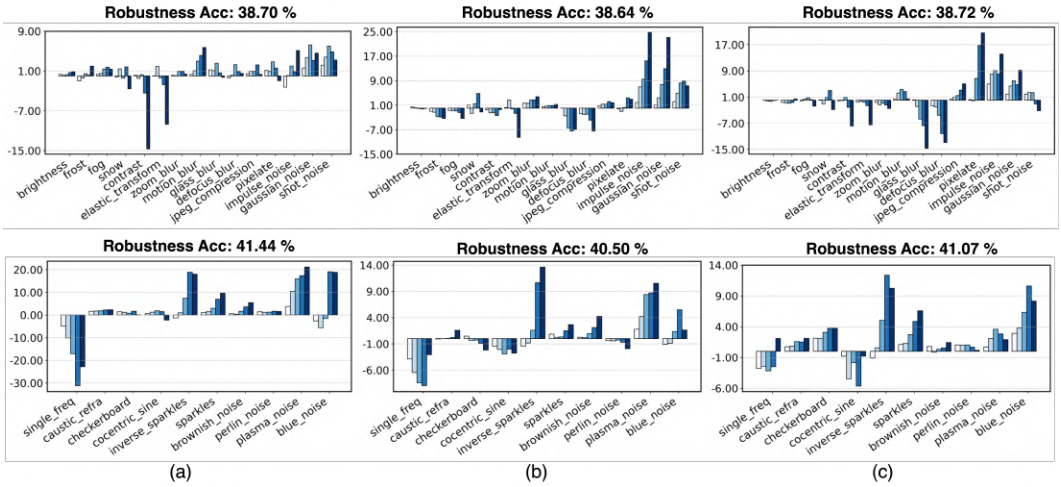


Figure 3: Relative robustness accuracy gain (%) on ImageNet-C (top) and ImageNet-$\bar{\text{C}}$ (bottom) for ResNet50, trained using DST with gradient-based strategies (i.e., (a) RigL, (b) MEST$_g$, (c) GraNet$_g$) at a sparsity ratio of 0.1, compared to a dense baseline (which has a mean robustness accuracy of 38.38% on ImageNet-C and 40.38% on ImageNet-$\bar{\text{C}}$). Positive values reflect better performance by the sparse models compared to the dense baseline. The five bars, ranging from light to dark shades, correspond to corruption severity levels from 1 to 5 for each corruption type. The title indicates the mean robustness accuracy for each corresponding DST method.

Additionally, we observe that DST models exhibit a more pronounced advantage in robustness over Dense Training models under high severity levels, particularly with noise corruptions. Specifically, as shown in Figure 3 (top (b)), the robustness performance of MEST$_g$ at severity level 5 for impulse noise and Gaussian noise demonstrates nearly a 25% improvement. This finding is also evident in CIFAR10-C, CIFAR100-C, and TinyImageNet-C; more details can be found in Appendix A.4. This observation highlights the potential of models trained with DST to tackle high-frequency corruptions, especially in harsher scenarios, offering the potential to enhance DNNs' robustness in complex corruption-prone environments.

4.3 Robustness of DST models: is true for other datasets, tasks, and architectures?

Herein, we evaluate the robustness of DST models using a more realistic corruption dataset (e.g., ImageNet-3DCC Kar et al. (2022)), and across different tasks (e.g., video classification), with modern architectures (transformer-based, e.g., DeiT-base Touvron et al. (2021)).

Evaluation on ImageNet-3DCC. We evaluate DST models on the ImageNet-3DCC, a challenging dataset with realistic corruptions. The ImageNet-3DCC dataset introduces corruptions aligned with 3D scene geometry, simulating real-world shifts including camera motion, weather conditions, occlusions, depth of field, and lighting variations. The improved average test accuracy on ImageNet-3DCC, as shown in Table 1, indicates that DST models excel not only at handling synthetic distortions, as discussed in Section 4.1 and Section 4.2, but also at adapting to practical, real-world corruptions.

strong performance on ImageNet more quickly with this framework Liu et al. (2021a); Nowak et al. (2023). However, it is worth noting that the random-based approach has also shown significant improvements when implemented in JAX Lee et al. (2023), which we plan to explore in future work.

Evaluation on video classification. We extended our exploration to video data using the UCF101 dataset Soomro et al. (2012), an action recognition dataset containing 101 categories and approximately 13K videos. Following the settings described in Schiappa et al. (2023), we compared the performance of dense training and DST at a sparsity level of 0.1 on two architectures: 3D ResNet50 and Two-Stream Inflated 3D ConvNet (I3D) Carreira & Zisserman (2017). Further details are provided in Appendix A.3.1. Both models were trained from scratch for 100 epochs on UCF101. Our analysis, based on a test set featuring 16 corruptions across severity levels 1 to 4, as shown in Table 1, revealed that DST models consistently achieved superior robust accuracy compared to their dense counterparts, even within the context of video data.

Table 1: The robustness accuracy (%) of dense and DST models on ImageNet-3DCC and UFC101 datasets.

Dataset	Model	Dense	RigL	MEST_g	GraNet_g
ImageNet-3DCC	ResNet50	43.62	**44.08**	**43.87**	**44.46**
UCF101	3D ResNet50	51.14	**52.71**	**53.00**	**53.57**
	I3D	53.58	**55.00**	**56.30**	**55.21**

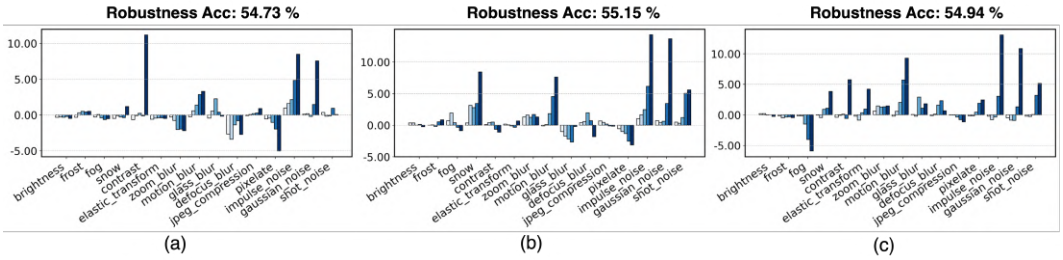


Figure 4: Relative robustness accuracy gain (%) on ImageNet-C for a DeiT-base, dynamically trained with gradient-based methods (i.e., (a) RigL, (b) MEST_g, (c) GraNet_g) at a sparsity ratio of 0.1, compared to a dense baseline (with a robustness accuracy of 54.68% on ImageNet-C). Positive values indicate superior performance by the sparse models.

Evaluation on transformer-based architecture. We evaluate the relative robustness accuracy gain of various sparse and dense DeiT-base models on ImageNet-C across different corruption types, as shown in Figure 4. Following the experimental settings in Chen et al. (2021), we conducted training from scratch over 200 epochs. The results show that DST models generally outperform their dense counterparts, with a particularly better advantage in handling noise corruptions. This finding, consistent with our observations in CNNs discussed in previous sections, suggests that the robustness benefits of DST models are applicable to alternative architectures, including transformer-based model.

In summary, our empirical findings indicate that *DST methods can surpass Dense Training models in robustness performance across various datasets at specific sparsity ratios, without (or even reduce) extra resource cost* [4]. Furthermore, DST models tend to perform better against different types of corruption, particularly those with more high-frequency information. The mystery behind these observations will be further explored in the following sections.

5 What is Behind the Corruption Robustness Gain in DST?

Building on the previous empirical results, where DST showed improved robustness against common corruptions, we are motivated to explore the inner reasons for the robustness of DST models. Our focus is on analyzing how DST and Dense Training models utilize their parameters in the spatial domain and process the input in the spectral domain.

5.1 Analysis Through Spatial Domain

Traditionally, the constraints on weight values, such as L_2 regularization, during dense training have been an effective method to mitigate the problem of overfitting. In contrast, DST imposes sparsity constraints, enabling the selective removal of parameters. In this section, we aim to intuitively

[4] We use a binary masking operation to simulate the sparisty, the computation cost of binary masking operation is negligible or even reduced when the hardware support sparse matrix product.

investigate how DST affects the location of non-zero weights. To achieve this, we performed a visualization analysis to pinpoint the locations of sparse parameters for DST models. Figure 5 showcases the non-zero weight counts and the sum of weight magnitudes (i.e., absolute values) for each kernel in a specific layer.

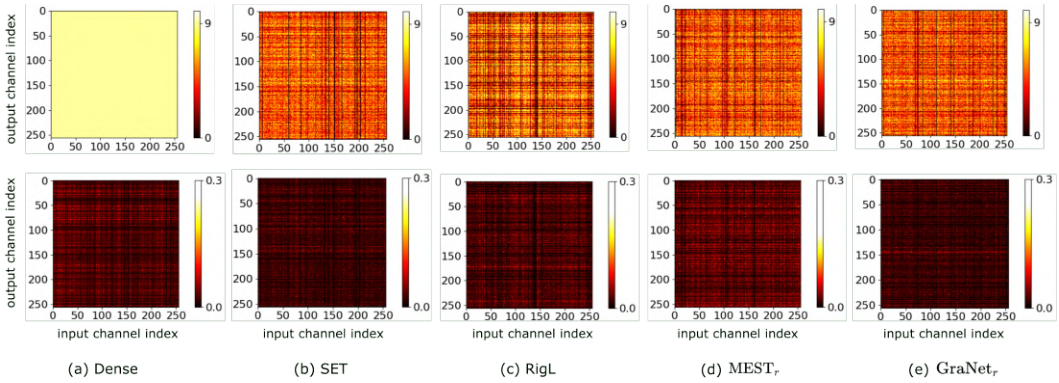


Figure 5: Visualization of the non-zero weight count (1st row) and the sum of weight magnitudes (2nd row) within a 3×3 kernel from the #layer3.3.conv2 of ResNet34 after training on the CIFAR100 using dense training or different DST methods (with a sparsity ratio of 0.5). Each value in the figure is derived from calculations within the kernel, with lighter colors indicating larger values.

Interestingly, the heatmaps in Figure 5 show dark vertical and horizontal lines, indicating a concentration of either sparsified weights or weights with smaller magnitudes in the filters corresponding to specific input or output channels (features). While both dense and DST models exhibit such a concentration phenomenon, in the case of the dense model, this concentration is solely due to the adjustment of weight magnitudes. These magnitudes tend to converge near, but not exactly at, zero, as indicated by the consistently non-zero weight count of 9 in Figure 5 (1st row (a)). In contrast, the dynamic sparse model attributes this phenomenon to both the adjustment of weight magnitudes and the allocation of sparsified weights, as shown in Figure 5 (1st row (b, c, d, e)), where the non-zero weight count approaches zero. The findings are consistently evident in other layers, particularly in deeper layers, and across different models. More visualizations can be found in the Appendix A.5. Compared to dense training, we suggest that dynamic sparsity in DST introduces a form of implicit regularization. This leads to the prominence of sparsified weights on particular filters, indicating a **reduced connectivity to specific features that are determined to be less crucial**.

Furthermore, based on the empirical results in Section 4.2, we observed that DST models are surprisingly more effective at countering corruptions rich in high-frequency information compared to their dense training counterparts. We assume that the improved robustness of DST models against these corruptions stems from their inherent tendency to de-emphasize high-frequency features. This selective focus on specific frequency ranges enhances the network's robustness against high-frequency corruptions, thereby providing a distinct advantage in handling noise corruptions rich in high-frequency information. Further analysis will be provided in the following section.

5.2 Analysis From Spectral Domain

Motivated by previous work Yin et al. (2019), which examines ImageNet accuracy after low-pass and high-pass filtering to understand how neural networks utilize frequency information for image recognition, and building on the observations and assumptions from the previous section, we will further explore dynamic sparse models from a spectral perspective.

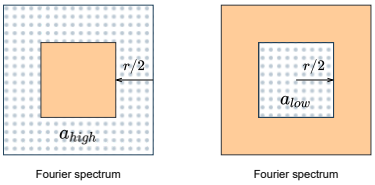


Figure 6: Illustration of high-frequency a_{high}, and low-frequency a_{low} attenuation. Gray areas indicate the removed frequency components, with low frequencies near the center.

In this section, we explore how the test accuracy of both DST and Dense Training models is affected by the removal of high and low-frequency components from the test data after training. In detail, the test images are first transformed into the Fourier spectrum, followed by a frequency attenuation operation. The attenuation is applied in two forms: high-frequency

attenuation a_{high} and low-frequency attenuation a_{low}, each controlled by an attenuation radius r. A larger r leads to more information being filtered out, as depicted in Figure 6.

We define $\texttt{Acc}_{\texttt{high}}(M, r)$ and $\texttt{Acc}_{\texttt{low}}(M, r)$ as the test accuracies of a model $M \in \mathcal{M}$, when tested on datasets processed with high-frequency attenuation a_{high} and low-frequency attenuation a_{low} respectively, using an attenuation radius r. Here, $\mathcal{M}$ represents the set of all considered models. The Radius-Accuracy (RA) curves for a_{high} and a_{low} illustrate the relationship between frequency attenuation range and test accuracy:

$$\mathcal{C}_{M,\texttt{high}} := \{(r, \texttt{Acc}_{\texttt{high}}(M, r)) \mid M \in \mathcal{M}, r \in \mathcal{R}\}, \tag{1}$$

$$\mathcal{C}_{M,\texttt{low}} := \{(r, \texttt{Acc}_{\texttt{low}}(M, r)) \mid M \in \mathcal{M}, r \in \mathcal{R}\}, \tag{2}$$

where $\mathcal{R}$ denotes the set of attenuation radius. Figure 7 (left) illustrates a toy example of ImageNet with attenuated high and low-frequency information at different attenuation radius, and provides the RA Curves $\mathcal{C}_{M,\texttt{high}}$ and $\mathcal{C}_{M,\texttt{low}}$ colored with orange and blue, respectively. In detail, with high-frequency attenuation, images appear blurred but remain recognizable to the human. In contrast, low-frequency attenuation leaves only the basic outlines of objects visible. Figure 7 (right) shows the detailed RA curves for both dense and dynamic sparse models across different datasets. These models are tested on the CIFAR100, TinyImageNet, and ImageNet test sets, each processed with high- and low-frequency attenuation at varying radius.

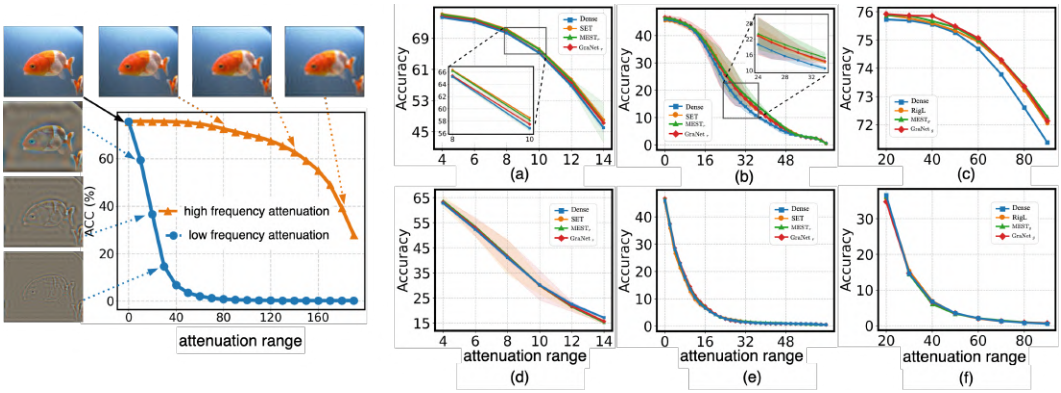


Figure 7: (Left) A toy example demonstrating the impact of high- and low-frequency attenuation on dense ResNet50 model using the ImageNet test set. (Right-top) High-frequency attenuation performance: (a) ResNet34 (sparsity 0.5) on CIFAR100, (b) EfficientNet-B0 (sparsity 0.5) on Tiny-ImageNet, and (c) ResNet50 (sparsity 0.1) on ImageNet. (Right-bottom) Low-frequency attenuation performance for the same models and datasets. The x-axis represents the attenuation radius, and the y-axis shows test accuracy.

Observation 1: Equal attention for low-frequency information. Given models $M \in \mathcal{M}$ and the set of attenuation radius $r \in \mathcal{R}$, it is observed that $\mathcal{C}_{M,\texttt{low}}$ lie below the $\mathcal{C}_{M,\texttt{high}}$. This reveals that attenuating low-frequency information significantly degrades classification accuracy, highlighting the crucial role of low-frequency information in image recognition tasks, consistent with findings in Yin et al. (2019). Under low-frequency attenuation a_{low}, both DST and Dense Training models exhibit similar performance, as shown in Figure 7 (right, 2nd row), suggesting that DST models allocate attention to low-frequency information in a manner comparable to their Dense Training counterparts.

Observation 2: Less attention to high-frequency information: When subjected to high-frequency attenuation a_{high}, DST models consistently outperform Dense Training models across all evaluated datasets, as shown in Figure 7 (right, 1st row). This suggests that high-frequency attenuation has less impact on DST models, indicating they are less sensitive to high-frequency information. This finding further supports the robustness of dynamic sparse models against common image corruptions rich in high-frequency content, as discussed in Section 5.1.

Based on the above observation, we conclude that DST introduces a form of implicit regularization, reducing the focus on high-frequency information. This selective attention in dynamic sparse models enhances their robustness, especially in handling corrupted images with high-frequency content, such as noise corruption.

6 FURTHER ANALYSIS AND DISCUSSION

6.1 DSCR HYPOTHESIS IN RECENT DST ALGORITHMS

Table 2: Robustness accuracy (%) on Fashion MNIST-C.

Dense Training	Dynamic Sparse Training				
	Sparsity	SET	RigL	CHTs+CSTI	CHTs+BSW
19.82 ± 0.32	97%	23.44 ± 1.79	25.23 ± 1.39	24.50 ± 1.03	**26.64 ± 1.06**
	95%	25.55 ± 0.32	25.03 ± 1.11	**27.38 ± 0.25**	27.13 ± 0.33

In this paper, we primarily investigate the robustness of fundamental DST models initialized from random sparse networks with the basic removal and regrowth criteria, to validate our DSCR hypothesis in commonly used settings. With the emergence of other recent and promising DST methods, we also include an initial exploration to further validate our hypothesis on these methods, such as Cannistraci-Hebb soft training (CHTs) Zhang et al. (2024b;a), which employ more complex growth criteria and topology initialization. We compare dense training with CHTs using Correlated Sparse Topological Initialization (CHTs+CSTI) and bipartite small-world networks (CHTs+BSW) on robustness accuracy, using the Fashion-MNIST corrupted dataset with an MLP model, as shown in Table 2. More training details are provided in Appendix A.3.1. The results indicate that the recent DST method (i.e., CHTs) achieves even more inspiring robustness performance, further validating our DSCR hypothesis on the robustness of DST.

6.2 DISCUSSION AND LIMITATION

Our study questions the conventional approach of using "de facto" dense training for model robustness in favor of sparse training. Due to the page constraint, we also analyze the complementarity between DST and data augmentation (e.g., Mixup Zhang et al. (2018)) in Appendix A.6, and further explore DST's robustness with Out-Of-Domain (OOD) data and Grad-CAM Selvaraju et al. (2017) in Appendix A.7 and A.8, respectively. In the future, it is worth moving beyond the basic fact revealed in this paper, i.e., vanilla DST outperforms vanilla Dense Training for model robustness, and to devise new DST methods for model robustness especially in the high sparsity regime, and to replace Dense Training with DST in the current state-of-the-art techniques for model robustness. Additionally, while we empirically analyze robustness behaviors between dense and sparse training and provide insights from both spatial and frequency domains, further work on the theoretical understanding of this phenomenon, along with a fine-grained exploration of different components of DST (e.g., topology initialization, regrowth methods and regrowth weight initialization), could provide deeper insights for the development of more robust and efficient networks. This would be a promising direction for future research.

7 CONCLUSION

Table 3: Summary of the main results presenting the 9 scenarios studied in this paper. The table takes a snapshot at a particular sparsity level and presents the robustness accuracy (%) of the gradient-based regrow DST methods. The bold values represent the best performer between Dense Training and each DST method taken separately.

Data type	Dataset	Model	Dense Training	Dynamic Sparse Training			
				Sparsity	RigL	MEST_g	GraNet_g
images	CIFAR10-C	VGG16	73.32	0.4	**73.64**	**73.47**	**73.32**
	CIFAR100-C	ResNet34	51.65	0.4	**52.56**	**52.56**	**51.78**
	TinyImageNet-C	EfficientNet-B0	15.42	0.4	**16.46**	**16.5**	**15.82**
	ImageNet-C	ResNet50	38.38	0.1	**38.70**	**38.64**	**38.72**
	ImageNet-$\bar{\text{C}}$	ResNet50	40.38	0.1	**41.44**	**40.50**	**41.07**
	ImageNet-3DCC	ResNet50	43.62	0.1	**44.08**	**43.87**	**44.46**
	ImageNet-C	DeiT	54.68	0.1	**54.73**	**55.15**	**54.94**
videos	UCF101	3D ResNet50	51.14	0.1	**52.71**	**53.00**	**53.57**
	UCF101	I3D	53.58	0.1	**55.00**	**56.3**	**55.21**
Total wins			0		9		

This paper questioned the typical approach of using "de facto" Dense Training to maximize model robustness in deep learning, and revealed an unexpected finding. That is: "*Dynamic Sparse Training at low sparsity levels can train more robust models against image corruption than Dense Training.*". This finding has been confined formally in the DSCR Hypothesis and rigorously validated on nine scenarios (as summarized in Table 3 where a striking observation can be made - all DST algorithms studied outperform Dense Training in all scenarios). To understand what factors contribute to this compelling and intriguing DST behaviour, to the end, we present a deeper analyze. We show that the dynamic sparsity process performs a form of implicit regularization which makes the model to automatically learn to focus on the more informative features (e.g., rich in low-frequency), while paying less attention to the less informative ones (e.g., rich in high-frequency).

REFERENCES

Görkem Algan and Ilkay Ulusoy. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, 215:106771, 2021.

Mike Ashby, Christiaan Baaij, Peter Baldwin, Martijn Bastiaan, Oliver Bunting, Aiken Cairncross, Christopher Chalmers, Liz Corrigan, Sam Davis, N. van Doorn, Jon Fowler, Graham Hazel, Basile Henry, David Page, Jonny Shipton, and S. C. Steenkamp. Exploiting unstructured sparsity on next-generation datacenter hardware. 2019.

Zahra Atashgahi, Ghada Sokar, Tim van der Lee, Elena Mocanu, Decebal Constantin Mocanu, Raymond Veldhuis, and Mykola Pechenizkiy. Quick and robust feature selection: the strength of energy-efficient sparse training for autoencoders. *Machine Learning*, pp. 1–38, 2022.

Anna Bair, Hongxu Yin, Maying Shen, Pavlo Molchanov, and José M. Álvarez. Adaptive sharpness-aware pruning for robust sparse networks. In *International Conference on Learning Representations (ICLR)*, May 2024.

Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. Deep rewiring: Training very sparse deep networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 4724–4733. IEEE Computer Society, 2017.

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Yang Zhang, Shiyu Chang, and Zhangyang Wang. Data-efficient double-win lottery tickets from robust pre-training. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3747–3759. PMLR, 17–23 Jul 2022a.

Tianlong Chen, Zhenyu Zhang, Pengjun Wang, Santosh Balachandra, Haoyu Ma, Zehao Wang, and Zhangyang Wang. Sparsity winning twice: Better robust generalization from more efficient training. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/chen20j.html`.

Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2):292–308, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009.

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

James Diffenderfer, Brian R. Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

Utku Evci, Fabian Pedregosa, Aidan Gomez, and Erich Elsen. The difficulty of training sparse neural networks. *arXiv preprint arXiv:1906.10732*, 2019.

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

Utku Evci, Yani Ioannou, Cem Keskin, and Yann Dauphin. Gradient flow in sparse neural networks and how lottery tickets win. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6577–6586, 2022.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269. PMLR, 2020.

Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. ISBN 978-0-262-03561-3. URL http://www.deeplearningbook.org/.

Laura Graesser, Utku Evci, Erich Elsen, and Pablo Samuel Castro. The state of sparse training in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 7766–7792. PMLR, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*.

Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gmrxHFvB.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.

Siddhant M. Jayakumar, Razvan Pascanu, Jack W. Rae, Simon Osindero, and Erich Elsen. Top-kast: Top-k always sparse training. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Oguzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 18941–18952. IEEE, 2022.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Joo Hyung Lee, Wonpyo Park, Nicole Mitchell, Jonathan Pilault, Johan S. Obando-Ceron, Han-Byul Kim, Namhoon Lee, Elias Frantar, Yun Long, Amir Yazdanbakhsh, Shivani Agrawal, Suvinay Subramanian, Xin Wang, Sheng-Chun Kao, Xingyao Zhang, Trevor Gale, Aart Bik, Woohyun Han, Milen Ferev, Zhonglin Han, Hong-Seok Kim, Yann N. Dauphin, Karolina Dziugaite, Pablo Samuel Castro, and Utku Evci. Jaxpruner: A concise library for sparsity research. *CoRR*, abs/2304.14082, 2023.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31, 2018.

Ningyi Liao, Shufan Wang, Liyao Xiang, Nanyang Ye, Shuo Shao, and Pengzhi Chu. Achieving adversarial robustness via sparsity. *Mach. Learn.*, 111(2):685–711, 2022. URL `https://doi.org/10.1007/s10994-021-06049-9`.

Lucas Liebenwein, Cenk Baykal, Brandon Carter, David Gifford, and Daniela Rus. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *CoRR*, abs/2103.03014, 2021.

Shiwei Liu and Zhangyang Wang. Ten lessons we have learned in the new "sparseland": A short handbook for sparse neural network researchers. *CoRR*, abs/2302.02596, 2023. URL `https://doi.org/10.48550/arXiv.2302.02596`.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 9908–9922, 2021a.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. *Advances in Neural Information Processing Systems*, 34:9908–9922, 2021b.

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

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Constantin Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=bXNl-myZkJl`.

Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 3571–3583, 2021.

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

Apostolos Modas, Rahul Rade, Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Prime: A few primitives can boost robustness to common corruptions. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 623–640, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19806-9.

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

Aleksandra Nowak, Bram Grooten, Decebal Constantin Mocanu, and Jacek Tabor. Fantastic weights and how to find them: Where to prune in dynamic sparse training. *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2023*, 2023.

Alexandra Peste, Eugenia Iofinova, Adrian Vladu, and Dan Alistarh. AC/DC: alternating compressed/decompressed training of deep neural networks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 8557–8570, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=S1gSj0NKvB`.

Tonmoy Saikia, Cordelia Schmid, and Thomas Brox. Improving robustness against common corruptions with frequency biased models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10211–10220, 2021.

Madeline Chantry Schiappa, Naman Biyani, Prudvi Kamtam, Shruti Vyas, Hamid Palangi, Vibhav Vineet, and Yogesh S. Rawat. A large-scale robustness analysis of video action recognition models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 14698–14708. IEEE, 2023.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Ghada Sokar, Zahra Atashgahi, Mykola Pechenizkiy, and Decebal Constantin Mocanu. Where to pay attention in sparse training for feature selection? In *Advances in Neural Information Processing Systems*, 2022.

Ghada Sokar, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Avoiding forgetting and allowing forward transfer in continual learning via sparse networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part III*, pp. 85–101. Springer, 2023.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. URL `http://arxiv.org/abs/1212.0402`.

Nicola Strisciuglio, Manuel Lopez-Antequera, and Nicolai Petkov. Enhanced robustness of convolutional networks with a push–pull inhibition layer. *Neural Computing and Applications*, 32(24):17957–17971, 2020. ISSN 1433-3058. doi: 10.1007/s00521-020-04751-8. URL https://doi.org/10.1007/s00521-020-04751-8.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 2019.

Yiqin Tan, Pihe Hu, Ling Pan, Jiatai Huang, and Longbo Huang. RLx2: Training a sparse deep reinforcement learning model from scratch. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DJEEqoAq7to.

Lukas Timpl, Rahim Entezari, Hanie Sedghi, Behnam Neyshabur, and Olga Saukh. Understanding the effect of sparsity on neural networks robustness. *CoRR*, abs/2206.10915, 2022. URL https://doi.org/10.48550/arXiv.2206.10915.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021.

Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=P5MtdcVdFZ4.

Shunxin Wang, Raymond Veldhuis, and Nicola Strisciuglio. The robustness of computer vision models against common corruptions: a survey. *arXiv preprint arXiv:2305.06024*, 2023.

Qiao Xiao, Boqian Wu, Yu Zhang, Shiwei Liu, Mykola Pechenizkiy, Elena Mocanu, and Decebal Constantin Mocanu. Dynamic sparse network for time series classification: Learning what to "see". In *Advances in Neural Information Processing Systems*, 2022.

Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

Geng Yuan, Xiaolong Ma, Wei Niu, Zhengang Li, Zhenglun Kong, Ning Liu, Yifan Gong, Zheng Zhan, Chaoyang He, Qing Jin, et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. *Advances in Neural Information Processing Systems*, 34:20838–20850, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Yingtao Zhang, Jialin Zhao, Ziheng Liao, Wenjing Wu, Umberto Michieli, and Carlo Vittorio Cannistraci. Brain-inspired sparse training in mlp and transformers with network science modeling via cannistraci-hebb soft rule. *Preprints*, June 2024a. doi: 10.20944/preprints202406.1136.v1. URL https://doi.org/10.20944/preprints202406.1136.v1.

Yingtao Zhang, Jialin Zhao, Wenjing Wu, Alessandro Muscoloni, and Carlo Vittorio Cannistraci. Epitopological learning and cannistraci-hebb network shape intelligence brain-inspired theory for ultra-sparse advantage in deep learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024b.

Jiaxu Zhao, Lu Yin, Shiwei Liu, Meng Fang, and Mykola Pechenizkiy. REST: enhancing group robustness in dnns through reweighted sparse training. In *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part II*, 2023.

Aojun Zhou, Yukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. Learning n:m fine-grained structured sparse neural networks from scratch. In *International Conference on Learning Representations*, 2021.

A APPENDIX

A.1 DATASET INTRODUCTION

A.1.1 CORRUPTION DATASET

Image corruptions refer to visible distortions in images that lead to data distribution shifts from that of the original data. They are common in practical applications when models are deployed in the real world. Typical corruptions include Gaussian noise, impulse noise, defocus blur, etc. In Figure 8, we display the common image corruptions on the CIFAR-10 dataset.

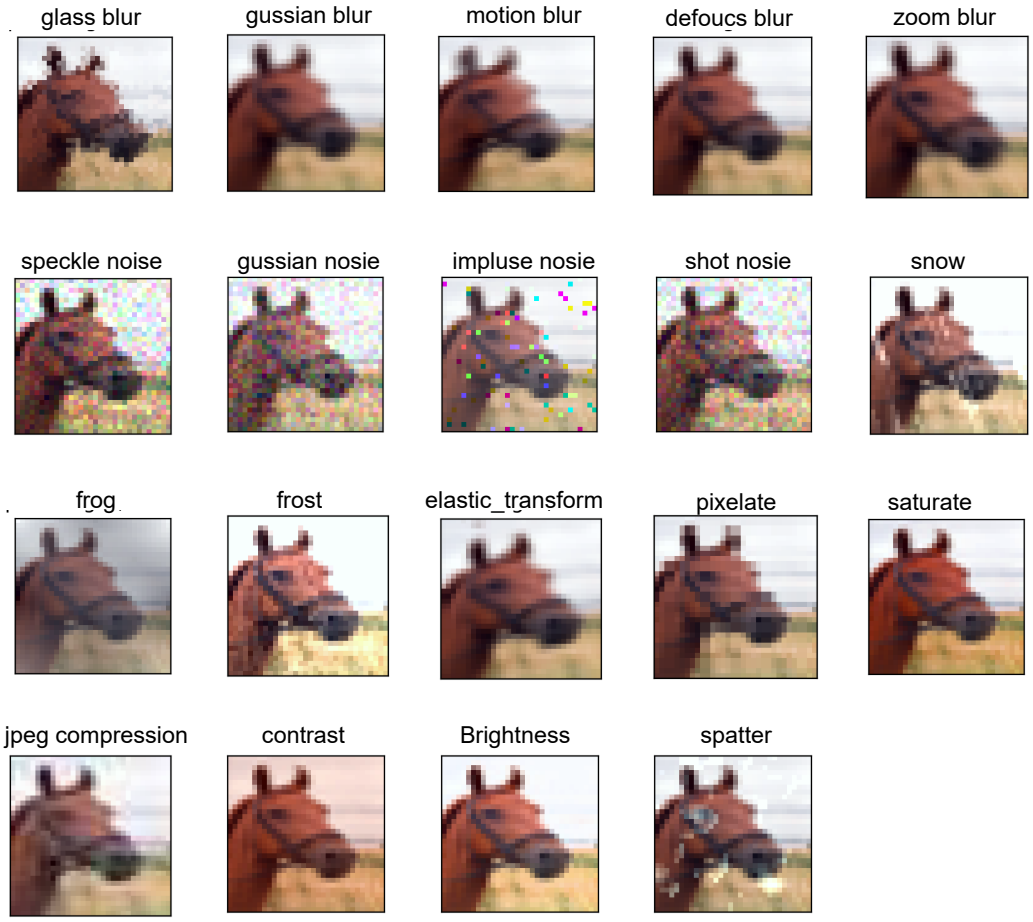


Figure 8: Visualization of the common image corruptions on CIFAR10 dataset.

A.2 RELATED BENCHMARKS

- **CIFAR-10/100-C** Hendrycks & Dietterich (2019) is synthetically generated on top of the test set of CIFAR-10/100 dataset. It includes 19 sub-datasets, each corrupted with a type of image corruption (Gaussian noise, impulse noise, shot noise, speckle noise, defocus blur, Gaussian blur, glass blur, motion blur, zoom blur, brightness, fog, frost, snow, spatter, contrast, elastic transform, JPEG compression, pixelate, saturate). Each corruption dataset contains five subsets, which have images corrupted with five severity levels. The higher the severity, the more influence the corruption has on the test images.
- **Tiny-ImageNet-C** Hendrycks & Dietterich (2019) consists of 15 types of common image corruptions, and is synthetically generated from Tiny-ImageNet dataset as well. The types of corruption are Gaussian noise, impulse noise, shot noise, defocus blur, glass blur, motion

918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971

blur, zoom blur, brightness, fog, frost, snow, contrast, pixelate, elastic transform, and JPEG compression, with 5 severity levels.

- **ImageNet-C** Hendrycks & Dietterich (2019) consists of 15 types of common image corruptions, and is synthetically generated from ImageNet dataset as well. The types of corruption are Gaussian noise, impulse noise, shot noise, defocus blur, glass blur, motion blur, zoom blur, brightness, fog, frost, snow, contrast, pixelate, elastic transform, and JPEG compression, with 5 severity levels.
- **ImageNet-$\bar{\text{C}}$** Mintun et al. (2021) are perceptually dissimilar to ImageNet-C in our transform feature space, consists of 10 types of common image corruptions. The types of corruption are blue noise, brownish noise, caustic refraction, checkerboard cutout, cocentric sine waves, inverse sparkles, perlin noise, plasma noise, single frequency greyscale, sparkles, with 5 severity levels.
- **ImageNet-3DCC** Kar et al. (2022): addresses several aspects of the real world, such as camera motion, weather, occlusions, depth of field, and lighting. It consists of 11 types of common image corruptions: near focus, far focus, fog 3d, flash, color quant, low light, xy motion blur, z motion blur, iso noise, bit error, h265 abr, with 5 severity levels.
- **Corrupted UCF101** Soomro et al. (2012): a realistic action recognition dataset featuring videos collected from YouTube, spanning 101 different action categories. It consists of 16 corruptions: motion blur, compression, defocus blur, gaussian noise, shot noise, impulse noise, zoom blur, rotate, static rotate, speckle noise, sampling, reverse sampling, jumble, box jumble, freeze, translate, with 4 severity levels.

A.3 Experiment Setup

A.3.1 Implementation Details

For CIFAR10, CIFAR100, and TinyImageNet, the codes were implemented in Python using PyTorch and executed on a single NVIDIA A100 GPU. Each was run three times using three fixed random seeds. The ImageNet experiments were run using distributed computing on four NVIDIA A100 GPUs. A batch size of 64 was utilized for each task, resulting in a combined total batch size of 256. For the UCF101 dataset, we followed the settings described in Schiappa et al. (2023) utilizing the codebase [5]. For the FashionMNIST dataset, we followed the training settings detailed in Zhang et al. (2024b;a), using the codebase [6].

Table 4: The experiment hyperparameters setting. The hyperparameters include Learning Rate (LR), Batch Size (BS), LR Schedule (LRS), Optimizer (Opt), Weight Decay (WD), Sparsity Distribution (Sparsity Dist), Topology Update Interval (ΔT), Pruning Rate Schedule (Sched), Initial Pruning Rate (P).

Model	Data	# Epoch	LR	BS	LRS	Opt	WD	Sparsity Dist [2]	ΔT	Sched [3]	P
VGG16	CIFAR10	160	0.1	100	Cosine	SGD [4]	5e-4	Uniform	500	polynomial	0.1
ResNet34	CIFAR100	160	0.1	100	Cosine	SGD	5e-4	Uniform	500	polynomial	0.1
EfficientNet-B0	TinyImageNet	100	0.1	128	Cosine	SGD	5e-4	Uniform	100	polynomial	0.1
ResNet50	ImageNet	90	0.1	256	Step [1]	SGD	1e-4	ERK	2000	polynomial	0.1
3D ResNet50/I3D	UCF101	80	0.1	8	Cosine	SGD	1e-3	ERK	300	polynomial	0.1
MLP	Fashion-MNIST	100	0.025	8	Cosine	SGD	5e-4	ERK	300	polynomial	0.1

[1] Step denotes a schedule that the learning rate decayed by 10 at every 30 epochs until 90 epochs.

[2] For the uniform distribution, the sparsity ratio of each layer is the same (i.e. $S_i = 1 - \|\boldsymbol{\theta}'_i\|_0 / \|\boldsymbol{\theta}_i\|_0$), where $\boldsymbol{\theta}_i$ is the parameters in i-th layer. ERK distribution allocates higher sparsities to layers with more parameters, while assigning lower sparsities to smaller layers. In ERK, the number of parameters of the sparse convolutional layers is scaled proportionally to $1 - \frac{n_{i-1}+n_i+w_i+h_i}{n_{i-1}\times n_i \times w_i \times h_i}$, where n_{i-1} and n_i are the number of input channels and output channels in i-th layer, w_i and h_i are the width and the height of the kernel.

[3] The remove and regrow ratio is decayed according to a polynomial strategy: $p \times \left(1 - \frac{\text{step}_{\text{current}}}{\text{step}_{\text{total}}}\right)^{0.01}$, where $\text{step}_{\text{current}}$ is the current step and $\text{step}_{\text{total}}$ is the total number of training steps.

[4] SGD optimizer with momentum 0.9.

[5] https://github.com/Maddy12/ActionRecognitionRobustnessEval

[6] https://github.com/biomedical-cybernetics/Cannistraci-Hebb-training

A.3.2 THE COMPUTATIONAL AND MEMORY CONSUMPTION

In this section, we present the computational cost in terms of Floating-Point Operations (FLOPs) for training and inference, as well as the memory cost in terms of the number of parameters, for different models across different sparsity ratios.

CIFAR-10: For the CIFAR10 dataset, we train VGG16 for 160 epochs. The soft memory bound for MEST is set to 10% of the target density. For example, if the sparse network's density is 0.3, then the soft memory bound is 0.03, resulting in an initial model density of 0.33. This soft memory bound is decayed to 0 using a polynomial strategy. The initial density of GraNet is set at 0.8. Every 500 training steps, it prunes weights to achieve a density described by the formula: $d_i - (d_i - d_t)\left(1 - \frac{t-t_0}{n\Delta t}\right)^3$, where d_i is the initial density, d_t is the target density, and $n\Delta t$ is set to half of the total training time.

Table 5: Training FLOPs (TrFLOPs, $\times 10^{16}$), Inference FLOPs (InFLOPs, $\times 10^9$), and the number of model parameters (Param, $\times 10^6$) for VGG16 on CIFAR10 at various sparsity ratios (S) using different sparsified algorithms.

	TrFLOPs ($\times 10^{16}$)						TeFLOPs ($\times 10^8$)						Param ($\times 10^6$)					
density	0.3	0.4	0.5	0.6	0.7	1.0	0.3	0.4	0.5	0.6	0.7	1.0	0.3	0.4	0.5	0.6	0.7	1.0
Dense	-	-	-	-	-	1.51	-	-	-	-	-	6.30	-	-	-	-	-	15.25
SET / RigL	0.46	0.61	0.76	0.91	1.06	-	1.92	2.55	3.17	3.80	4.42	-	4.95	6.43	7.89	9.37	10.81	-
MEST_r / MEST_g	0.51	0.67	0.84	1.00	1.17	-	1.92	2.55	3.17	3.80	4.42	-	4.95	6.43	7.89	9.37	10.81	-
GraNet_r / GraNet_g	0.68	0.73	0.85	0.97	1.09	-	1.92	2.55	3.17	3.80	4.42	-	4.95	6.43	7.89	9.37	10.81	-
AC/DC	0.89	1.03	1.11	1.19	1.27	-	1.92	2.55	3.17	3.80	4.42	-	4.95	6.43	7.89	9.37	10.81	-

CIFAR-100: For the CIFAR100 dataset, we train ResNet-34 for 160 epochs. The soft memory bound for MEST is set to 10% of the target density. For example, if the sparse network's density is 0.3, then the soft memory bound is 0.03, resulting in an initial model density of 0.33. This soft memory bound is decayed to 0 using a polynomial strategy. The initial density of GraNet is set at 0.8. Every 500 training steps, it prunes weights to achieve a density described by the formula: $d_i - (d_i - d_t)\left(1 - \frac{t-t_0}{n\Delta t}\right)^3$, where d_i is the initial density, d_t is the target density, and $n\Delta t$ is set to half of the total training time.

Table 6: Training FLOPs (TrFLOPs, $\times 10^{16}$), Inference FLOPs (InFLOPs, $\times 10^9$), and the number of model parameters (Param, $\times 10^6$) for ResNet34 on CIFAR100 at various sparsity ratios (S) using different sparsified algorithms.

	TrFLOPs ($\times 10^{16}$)						InFLOPs ($\times 10^9$)						Param ($\times 10^6$)					
density	0.3	0.4	0.5	0.6	0.7	1.0	0.3	0.4	0.5	0.6	0.7	1.0	0.3	0.4	0.5	0.6	0.7	1.0
dense	-	-	-	-	-	5.57	-	-	-	-	-	2.32	-	-	-	-	-	21.33
SET / RigL	1.68	2.24	2.79	3.35	3.90	-	0.67	0.93	1.18	1.40	1.62	-	6.45	8.57	10.70	12.82	14.95	-
MEST_r / MEST_g	1.85	2.46	3.07	3.68	4.29	-	0.67	0.93	1.18	1.40	1.62	-	6.45	8.57	10.70	12.82	14.95	-
GraNet_r / GraNet_g	2.51	2.68	3.13	3.63	4.02	-	0.67	0.93	1.18	1.40	1.62	-	6.45	8.57	10.70	12.82	14.95	-
AC/DC	3.51	3.80	4.10	4.39	4.68	-	0.67	0.93	1.18	1.40	1.62	-	6.45	8.57	10.70	12.82	14.95	-

TinyImageNet: For the TinyImageNet dataset, we train EfficientNet-B0 for 100 epochs. The soft memory bound for MEST is set to 10% of the target density. For example, if the sparse network's density is 0.3, then the soft memory bound is 0.03, resulting in an initial model density of 0.33. This soft memory bound is decayed to 0 using a polynomial strategy. The initial density of GraNet is set at 0.8. Every 500 training steps, it prunes weights to achieve a density described by the formula: $d_i - (d_i - d_t)\left(1 - \frac{t-t_0}{n\Delta t}\right)^3$, where d_i is the initial density, d_t is the target density, and $n\Delta t$ is set to half of the total training time.

ImageNet: For the ImageNet dataset, we train ResNet50 for 90 epochs. The soft memory bound for MEST is set to 0.2, and the sparse network's density is 0.9, then the soft memory bound is 0.92, resulting in an initial model density of 0.92. This soft memory bound is decayed to 0 using a polynomial strategy. The initial density of GraNet is set at 0.95. Every 2000 training steps, it prunes weights to achieve a density described by the formula: $d_i - (d_i - d_t)\left(1 - \frac{t-t_0}{n\Delta t}\right)^3$, where d_i is the initial density, d_t is the target density, and $n\Delta t$ is set to half of the total training time.

The Resource Consumption and Robustness. Even if it is outside of the scope of this paper, in the traditional DST style, we provide an overview of the relationship between resource efficiency

Table 7: Training FLOPs (TrFLOPs, $\times 10^{15}$), Inference FLOPs (InFLOPs, $\times 10^{9}$), and the number of model parameters (Param, $\times 10^{6}$) for EfficientNet-B0 on TinyImageNet at various sparsity ratios (S) using different sparsified algorithms.

	TrFLOPs ($\times 10^{15}$)						InFLOPs ($\times 10^{9}$)						Param ($\times 10^{6}$)					
density	0.3	0.4	0.5	0.6	0.7	1.0	0.3	0.4	0.5	0.6	0.7	1.0	0.3	0.4	0.5	0.6	0.7	1.0
Dense	-	-	-	-	-	1.98	-	-	-	-	-	6.63	-	-	-	-	-	4.26
SET/ RigL	0.66	0.85	1.04	1.23	1.42	-	2.21	2.85	3.48	4.12	4.75	-	1.89	2.29	2.68	2.68	3.08	-
MEST_r / MEST_g	0.72	0.92	1.13	1.34	1.55	-	2.21	2.85	3.48	4.12	4.75	-	1.89	2.29	2.68	2.68	3.08	-
GraNet_r / GraNet_g	0.89	1.02	1.15	1.32	1.46	-	2.21	2.85	3.48	4.12	4.75	-	1.89	2.29	2.68	2.68	3.08	-
AC/DC	1.19	1.30	1.42	1.53	1.65	-	2.21	2.85	3.48	4.12	4.75	-	1.89	2.29	2.68	2.68	3.08	-

Table 8: Training FLOPs (TrFLOPs, $\times 10^{18}$), Inference FLOPs (InFLOPs, $\times 10^{9}$), and the number of model parameters (Param, $\times 10^{6}$) for ResNet50 on ImageNet at various sparsity ratios (S) using different sparsification algorithms.

	TrFLOPs ($\times 10^{18}$)	InFLOPs ($\times 10^{9}$)	Param ($\times 10^{6}$)	Init density $\rightarrow$ Final density
Dense	2.28	8.21	25.56	1.0
RigL	2.17	7.80	23.00	$0.9 \rightarrow 0.9$
MEST_g	2.18	7.80	23.00	$0.92 \rightarrow 0.9$
GraNet_g	2.20	7.80	23.00	$0.95 \rightarrow 0.9$
AC/DC	2.26	7.80	23.00	$1.0 \rightarrow 0.9$

and the model's robustness against common corruption. Figure 9 showcases the computational cost (i.e. training FLOPs and model sizes) and robustness of dense and dynamic sparse models on TinyImageNet-C (results for other datasets can be found in the above.

We can find that the dense model, with the highest resource usage (e.g., FLOPs and parameter count), does not necessarily lead to better robustness. In the end, we also investigate other training paradigms: iterative dense and sparse training, represented by AC/DC Peste et al. (2021) algorithm. AC/DC begins with a dense warm-up, then starts alternating dense and sparse training, returning accurate sparse-dense model pairs at the end of the training process. We find that AC/DC also exhibits decent robustness, as in Figure 9. However, in most cases, AC/DC tends to require more training FLOPs compared with DST methods.

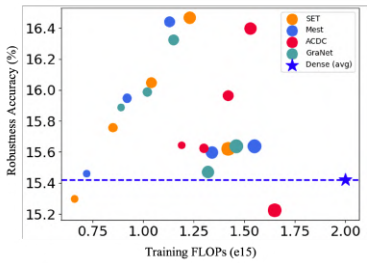


Figure 9: The comparison of training FLOPs, parameter count (represented by the area of each circle), and robustness accuracy of the dense and dynamic sparse EfficientNet-B0 on TinyImageNet. Each point represents the result for a specific sparsity ratio.

It is worth noting that the computational costs in this work are theoretical and not fully realized gains on current hardware infrastructures. This is a common limitation in the literature on sparse training methods, as most current hardware and software systems are optimized for dense matrix operations rather than sparse ones. However, recent advancements in model sparsity are increasingly aligning with hardware and software developments to fully leverage the benefits of sparsity. For example, NVIDIA's A100 GPU supports 2:4 sparsity Zhou et al. (2021), and other innovations are making strides toward efficient sparse implementations Chen et al. (2019); Ashby et al. (2019). Simultaneously, software libraries such as Liu et al. (2021b); Mocanu et al. (2018) are emerging to enable truly sparse network implementations. These advancements are paving the way for future deep neural networks to achieve greater efficiency in terms of computation, memory, and energy use.

A.4 THE ROBUSTNESS ACCURACY FOR DIFFERENT CORRUPTION TYPES AND SERVERITIES

Figure 10, 11, 12, 13 14 and 15 showcase the relative gains[7] in robustness at each of these severity levels. We observe that dynamic sparse models exhibit a more pronounced advantage in robustness over dense models, particularly under high severity levels of high-frequency corruption.

[7]Defined as $(Acc_{s,l} - Acc_{dense})/Acc_{d,l}$, where $Acc_{s,l}$ and $Acc_{d,l}$ represent the accuracies of dynamic sparse and dense models, respectively, under severity level l.

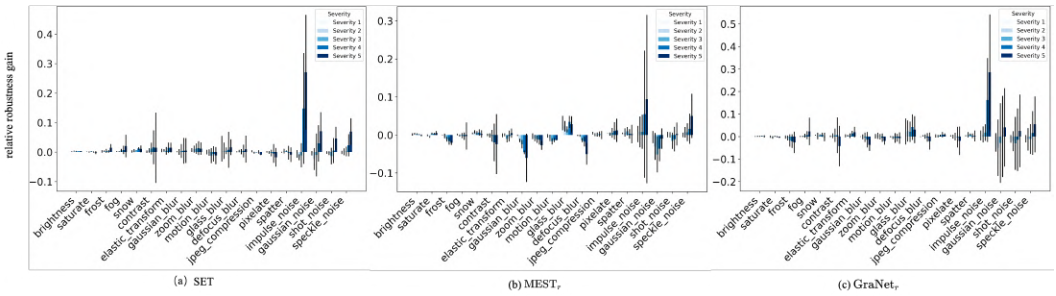


Figure 10: Compared with the the dense model, the relative robustness gain of different dynamic sparse models with sparsity 0.5 trained by (a) RigL, (b) MEST$_r$, (c) GraNet$_r$ under different severities of CIFAR10-C.

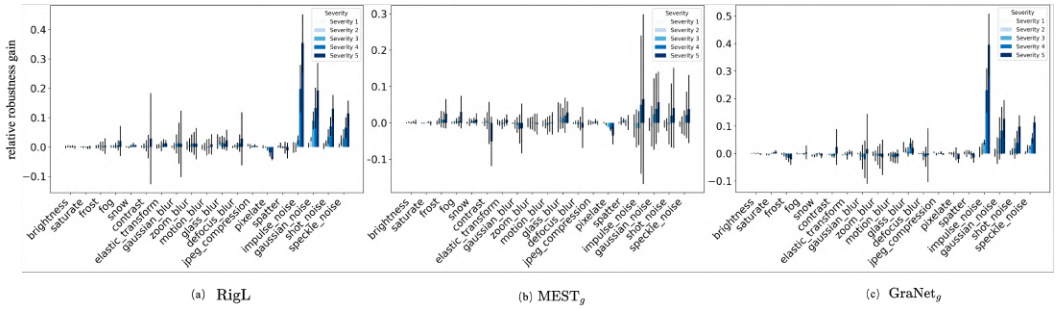


Figure 11: Compared with the the dense model, the relative robustness gain of different dynamic sparse models with sparsity 0.5 trained by (a) RigL, (b) MEST$_g$, (c) GraNet$_g$ under different severities of CIFAR10-C.

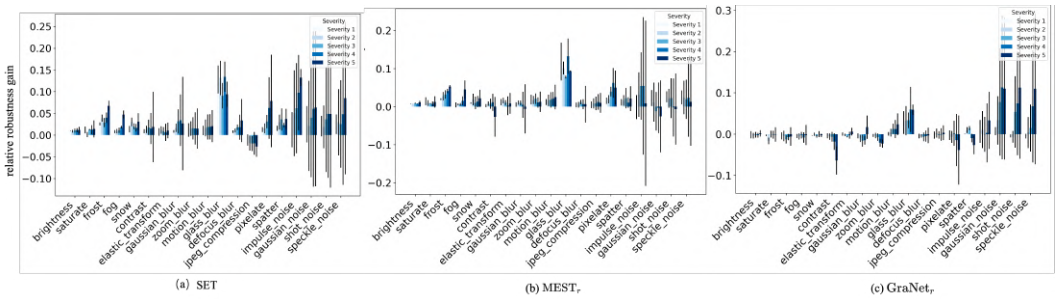


Figure 12: Compared with the the dense model, the relative robustness gain of different dynamic sparse models with sparsity 0.5 trained by (a) RigL, (b) MEST$_r$, (c) GraNet$_r$ under different severities of CIFAR100-C.

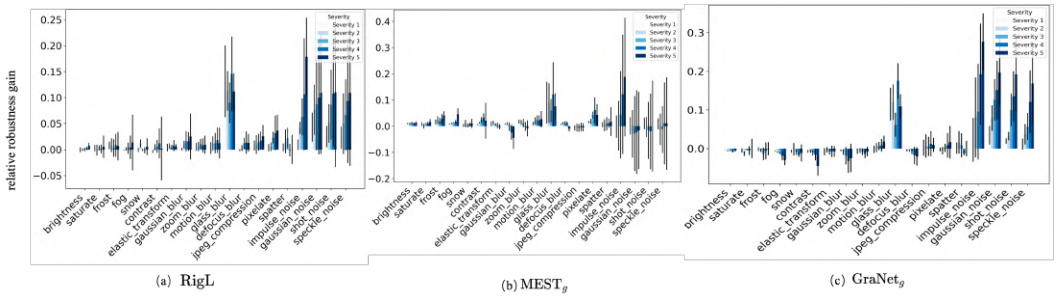
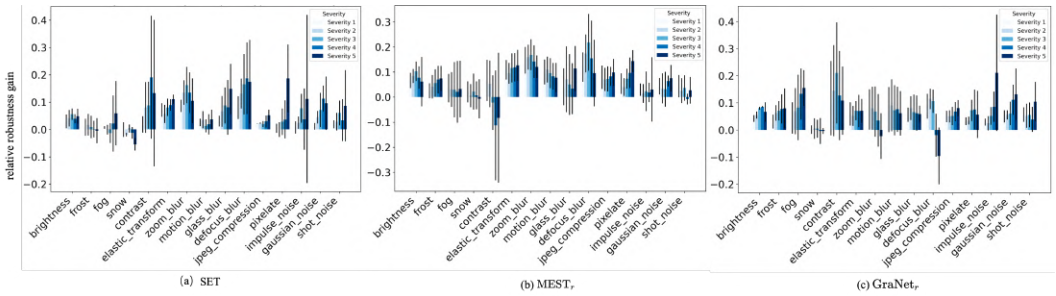
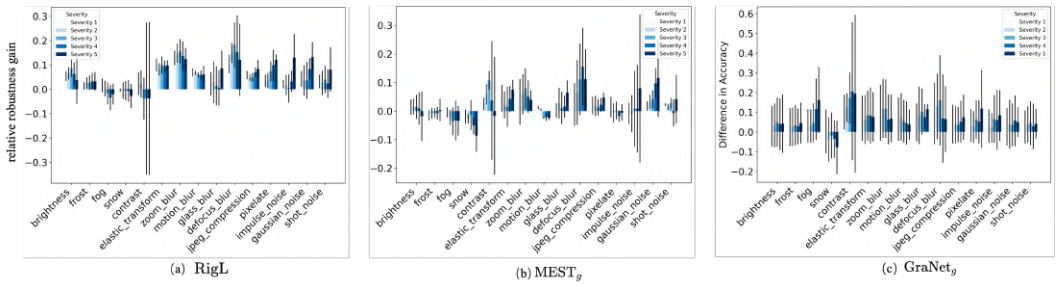


Figure 13: Compared with the the dense model, the relative robustness gain of different dynamic sparse models with sparsity 0.5 trained by (a) RigL, (b) MEST$_g$, (c) GraNet$_g$ under different severities of CIFAR100-C.



(a) SET (b) MEST$_r$ (c) GraNet$_r$

Figure 14: Compared with the the dense model, the relative robustness gain of different dynamic sparse models with sparsity 0.5 trained by (a) RigL, (b) MEST$_r$, (c) GraNet$_r$ under different severities of TinyImageNet-C.



(a) RigL (b) MEST$_g$ (c) GraNet$_g$

Figure 15: Compared with the the dense model, the relative robustness gain of different dynamic sparse models with sparsity 0.5 trained by (a) RigL, (b) MEST$_g$, (c) GraNet$_g$ under different severities of TinyImageNet-C.

A.5 Analysis Through the Lens of Filter

Figure 16 and 17 showcases the non-zero weight counts and the sum of weight magnitudes (i.e., absolute values) for each kernel in a specific layer of dynamic sparse VGG16 and ResNet34 trained on CIFAR10 and CIFAR100, respectively.

Additionally, we investigated the accuracy of test data subject to high-frequency or low-frequency information attenuation during dense or dynamic sparse training . In Figure 18, the results indicate that at the beginning of training, the superiority of dynamic sparse models in handling high-frequency information removal starts to appear, becoming more evident as training progresses. Considering the phenomenon of sparse weight concentration on the filter after training, it becomes evident that dynamic sparse training dynamically allocates computational resources to prioritize relevant low-frequency features during the learning process.

Table 9: Robustness accuracy (%) on CIFAR100 dataset with and without Mixup.

w/ mixup	Dense training	SET
×	51.68	**52.60**
✓	54.75	**54.93**

A.6 How Data Augmentation Interacts with DST?

We extend our analysis to explore how data augmentation (e.g., Mixup Zhang et al. (2018)) interacts with dynamic sparse training in comparison to dense training. From Table 9, we observe: In general, data augmentation helps improve model generalization by providing increased input variability. For DST, combining data augmentation with dynamic sparse training can lead to even greater robustness compared to dense training, suggesting that dynamic sparse training (e.g., SET) interacts with data augmentation in a complementary way.

Table 10: Average classification accuracy (%) on ImageNet-R dataset. The bold highlights result that are better than the dense model.

Dense	RigL	MEST$_g$	GraNet$_g$
37.84	**40.04**	**38.33**	**38.46**

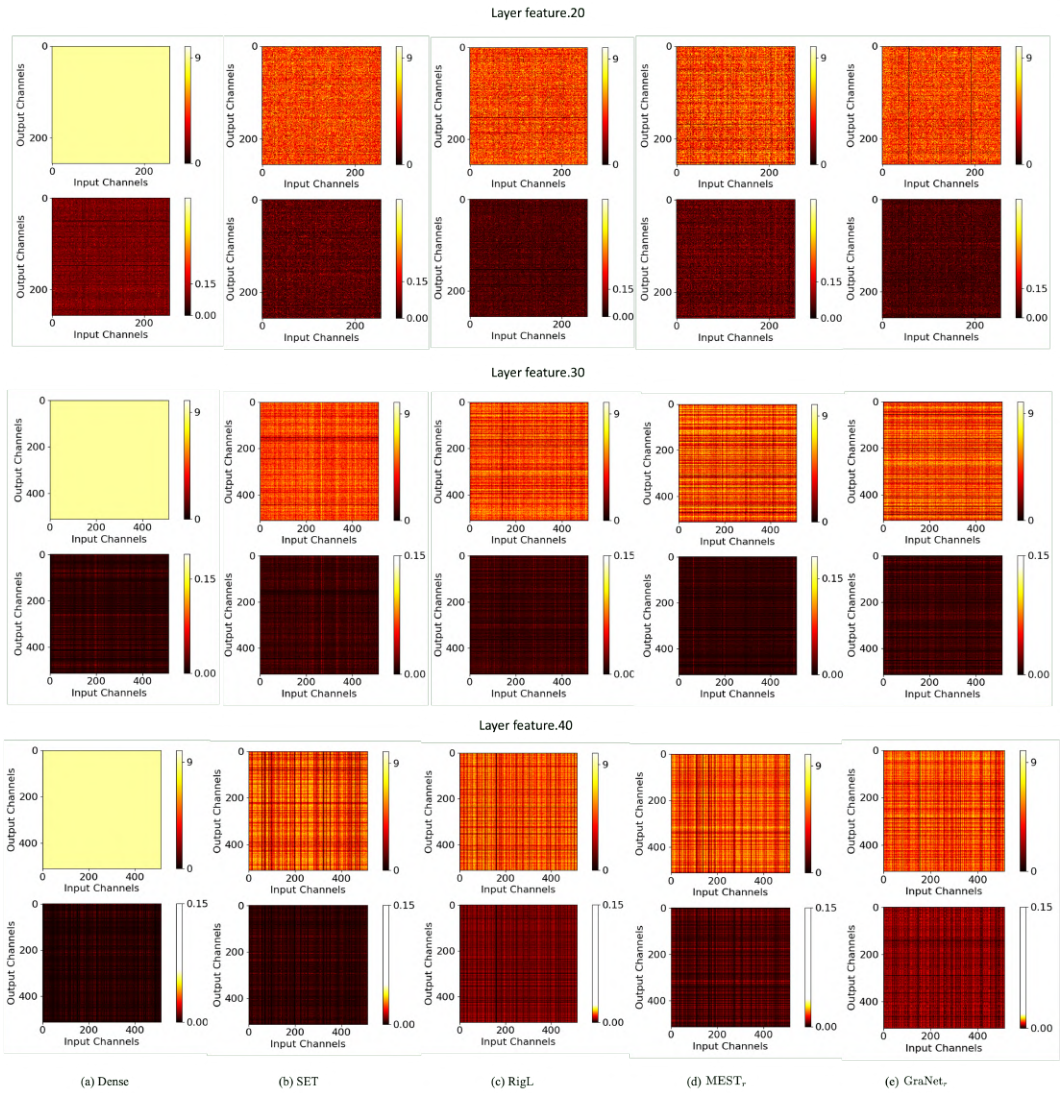


Figure 16: Visualizing non-zero weight count (1st row) and the sum of weight magnitudes (2nd row) in filters of VGG16 (#layer feature.20, #layer feature.30 and #layer feature.40 with kernel size 3×3) after training on CIFAR100 using different sparsification algorithms: (b) SET, (c) RigL, (d) MEST$_r$ and (e) GraNet$_r$, compared with (a) dense counterpart. Each point represents the corresponding value of a filter.

A.7 THE PERFORMANCE OF DST ON OUT-OF-DOMAIN (OOD) TEST DATA

ImageNet-R Hendrycks et al. with 30,000 images of art, cartoons, and other 14 renditions from 200 ImageNet classes, presents a notable domain shift from the original dataset. We report the average test accuracy on ImageNet-R for dense and DST models trained on the original ImageNet.

ImageNet-v2 Recht et al. is a valuable dataset for evaluating the generalization of models trained on original ImageNet dataset. ImageNet-v2 contains three test subsets: Top-images, Threshold-0.7, and Matched-frequency. Each subset comprises ten images per class, selected from a pool of candidates according to various selection frequencies.

As shown in Table 10 and 11, the results from ImageNet-R and ImageNet-v2 correspond to the evaluation on domain adaptation and generalization, respectively. The results show that DST models reliably surpass dense models on ImageNet-R and exhibit superior performance in the majority of scenarios on ImageNet-v2. This suggests that models trained with DST not only handle common data corruptions effectively but also excel in the face of significant domain shifts.

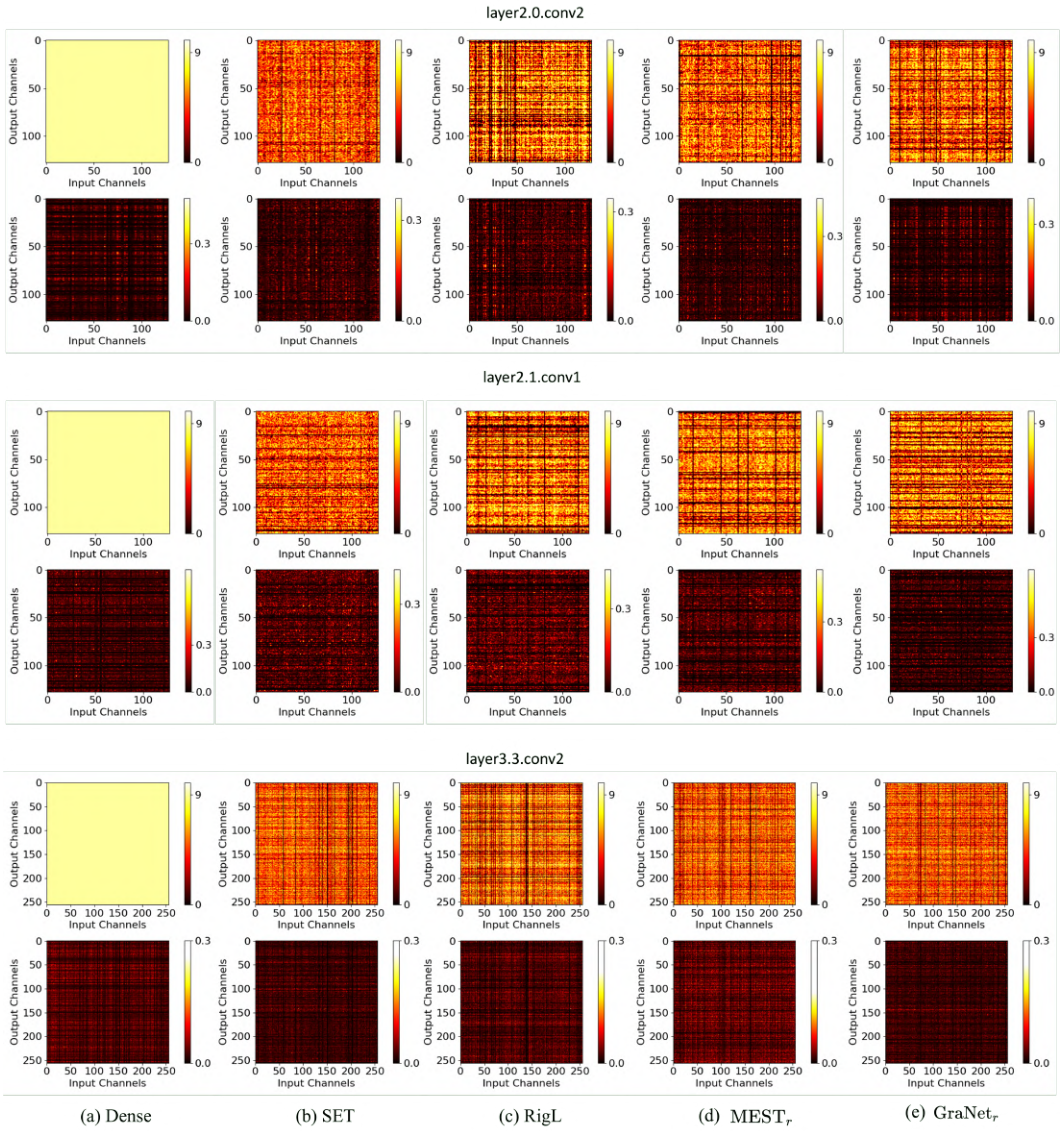


Figure 17: Visualizing non-zero weight count (1st row) and the sum of weight magnitudes (2nd row) in filters of ResNet34 (#layer2.0.conv2, #layer2.1.conv1 and #layerlayer3.3.conv2 with kernel size 3×3) after training on CIFAR100 using different sparsification algorithms: (b) SET, (c) RigL, (d) MEST$_r$ and (e) GraNet$_r$, compared with (a) dense counterpart. Each point represents the corresponding value of a filter.

A.8 GRAD-CAM

Grad-CAM (Gradient-weighted Class Activation Mapping) Selvaraju et al. (2017) generates a heatmap highlighting important regions by weighting the feature maps based on the gradient values. It provides practitioners with a visual indication of the areas on which the network focuses when making predictions. We visualize and compare the Grad-CAM outputs of both dense models and dynamic sparse models, as shown in Figure 19. We present different cases, including clean background ((a) and (b)), and complex background (c). Interestingly, we observe that the highlighting region of dynamic sparse models is smaller than that of the dense model. In

Table 11: Average robustness accuracy (%) on ImageNet-v2 dataset. The bold highlights result that are better than the dense model.

	Dense	RigL	MEST$_g$	GraNet$_g$
Top-images	77.80	**77.88**	**78.33**	**78.21**
Threshold-0.7	72.95	72.81	**73.23**	**73.58**
Matched-frequency	64.62	64.23	64.14	64.60
Avg. accuracy	71.79	71.64	**71.90**	**72.13**

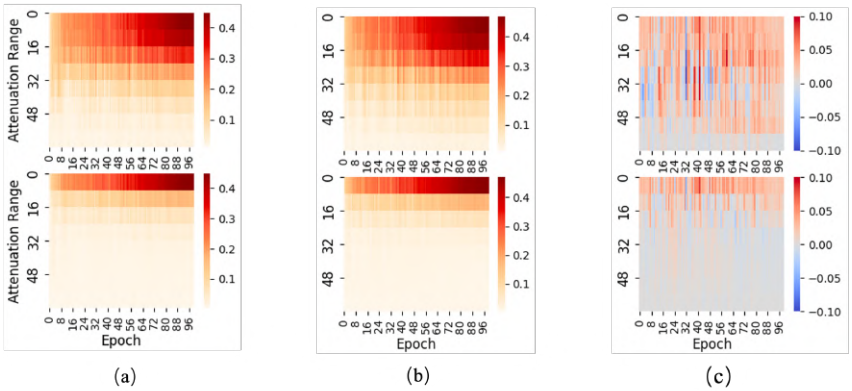


Figure 18: The impact of high-frequency (top) and low-frequency attenuation (bottom) on TinyImageNet during training. Test accuracy for (a) dense model, (b) dynamic sparse model from SET. (c) The accuracy difference between sparse and dense models, where a positive value indicates that the sparse models perform better.

Figure 19 (a), the dynamic sparse models focus on the face of the cat for prediction, while the dense models pay attention to both the face and body of the cat. In Figure 19 (b), the dynamic sparse models specifically emphasize the wings of the eagle. In contrast, the dense models primarily concentrate on a single wing while also capturing irrelevant contexts, leading to larger attention regions. In the case of a more complex background (In Figure 19 (c)), the dynamic sparse models tend to allocate some attention to the surrounding elements. However, the attention regions are still relatively smaller and primarily focused on the face of the panda, especially in the sparse model sparsified by SET.

These visualizations provide further support for our hypothesis that dynamic sparse models effectively allocate limited resources to the most crucial features. As a result, dynamic sparse models can be robust against image common corruptions, making them a promising approach for addressing real-world challenges.

B Impact Statement

In an era dominated by over-parameterized models and the pervasive presence of image corruption, the design of resource-aware and robust AI models is of increasing importance. Gaining insight into the robust behavior of dynamic sparse models against image corruption paves the way for embracing these environmentally friendly models in challenging environments. This holds significant implications across various domains, including medical diagnosis, robotics, autonomous vehicles, and other AI applications. Furthermore, our insights into the inner workings of sparse models help us understand the reasoning behind their robust decisions. Overall, we advance our fundamental understanding of dynamic sparse training and provide future perspectives for scalable, efficient, and trustworthy AI. We do not anticipate any negative societal impacts resulting from this research.

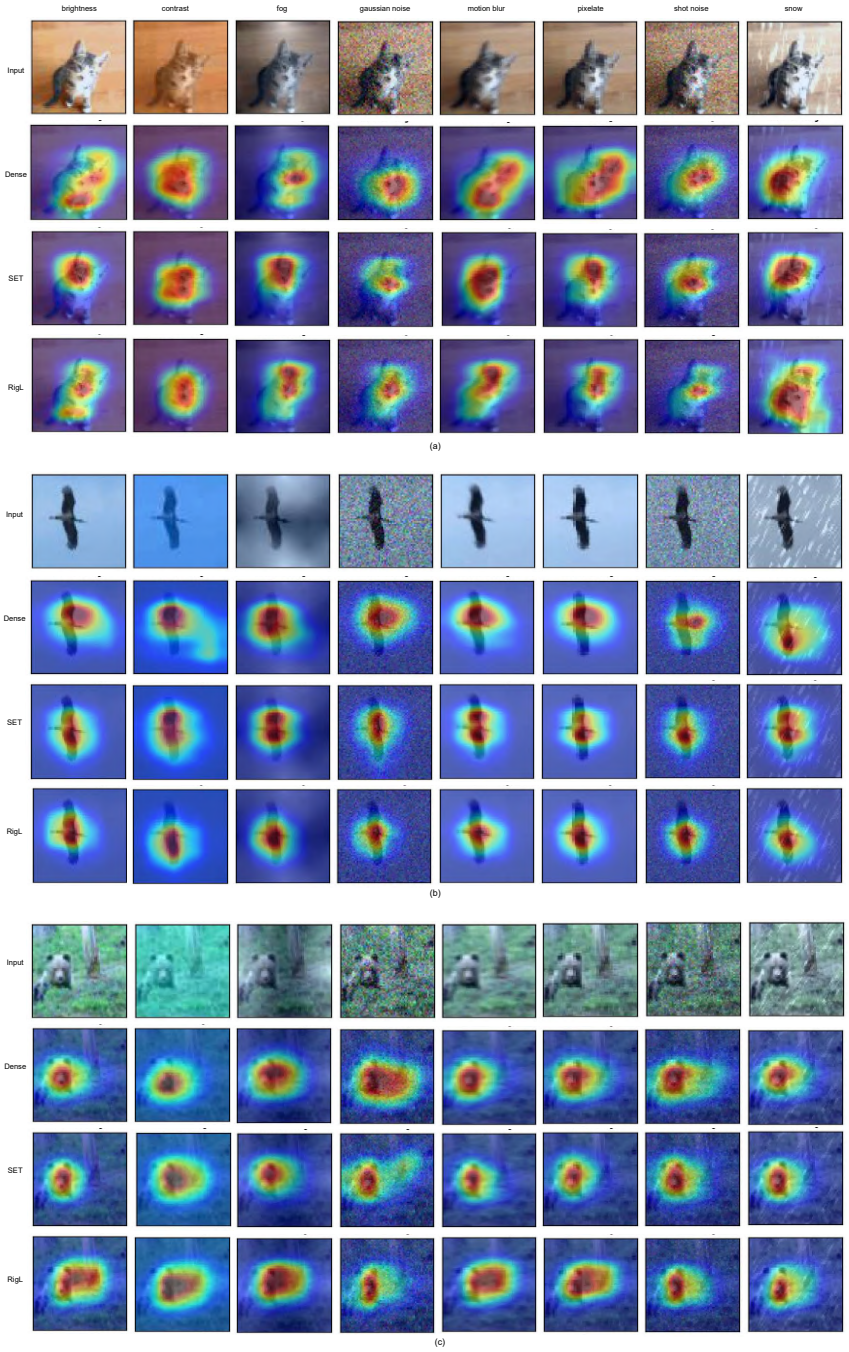


Figure 19: Grad-CAM for TinyImageNet-C. The images are corrupted by different corruptions with clean background (a) and (b), and complex background (c).