computers

MDPI

Article

On the Robustness of Compressed Models with Class Imbalance

Baraa Saeed Ali [1,†], Nabil Sarhan [2] and Mohammed Alawad [2,*]

[1] Electrical Engineering Department, University of Anbar, Ramadi, Anbar 55431, Iraq; baraa@wayne.edu
[2] Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA; nabil.sarhan@wayne.edu
* Correspondence: alawad@wayne.edu
† Current address: ECE Department, Wayne State University, Detroit, MI 48202, USA.

Abstract: Deep learning (DL) models have been deployed in various platforms, including resource-constrained environments such as edge computing, smartphones, and personal devices. Such deployment requires models to have smaller sizes and memory footprints. To this end, many model compression techniques proposed in the literature successfully reduce model sizes and maintain comparable accuracy. However, the robustness of compressed DL models against class imbalance, a natural phenomenon in real-life datasets, is still under-explored. We present a comprehensive experimental study of the performance and robustness of compressed DL models when trained on class-imbalanced datasets. We investigate the robustness of compressed DL models using three popular compression techniques (pruning, quantization, and knowledge distillation) with class-imbalanced variants of the CIFAR-10 dataset and show that compressed DL models are not robust against class imbalance in training datasets. We also show that different compression techniques have varying degrees of impact on the robustness of compressed DL models.

Keywords: class imbalance; deep learning; model compression; robustness

Citation: Ali, B.S.; Sarhan, N.; Alawad, M. On the Robustness of Compressed Models with Class Imbalance. *Computers* **2024**, *13*, 297. https:// doi.org/10.3390/computers13110297

Academic Editor: Thuseethan Selvarajah

Received: 1 October 2024
Revised: 10 November 2024
Accepted: 12 November 2024
Published: 16 November 2024

1. Introduction

The deployment of DL models in a wide range of real-life applications has become more popular and more successful than ever [1]. The success of DL models is partly due to their huge size—often consisting of millions or even billions of parameters—which contributes to their high accuracy [2]. Due to their massive size and memory footprints, DL models require extended periods of time for training and inference, resulting in increased demands for computational and memory resources [3]. For efficient deployment of such models in resource-constrained environments, such as edge computing, smartphones, and personal devices, it is necessary to shrink the model size and memory footprints without compromising model accuracy [4,5]. Toward this goal, the DL research community has focused on compressing the size of DL models while maintaining their original accuracy. Recent works show that DL models can be compressed by up to 90% [6] with minimal or no drop in accuracy. Many model compression techniques have been proposed in the literature, such as pruning [3], quantization [7], and knowledge distillation [8], that greatly reduce model size and memory footprints and maintain comparable test accuracy.

Compressed DL models, just like their original dense counterparts, should be robust against different kinds of unusual settings such as adversarial attacks, out-of-distribution (OOD) instances, and class imbalance in training datasets. Although the adversarial robustness and OOD robustness of compressed models have been hot research topics in the literature [9–12], little effort has been made to study the class imbalance robustness of compressed DL models.

The preserved accuracy of compressed DL models comes after training those models on balanced datasets, where each class has an equal number of instances in the training

dataset. In real-life applications, however, not all classes have the same number of instances, which causes those datasets to be class-imbalanced.

Class imbalance is a well-known problem in real-life DL-based classifiers [13]. The class imbalance happens when some classes (majority classes) have more samples in the training dataset, while the other classes (minority classes) have fewer samples. Examples include, but are not limited to, disease diagnosis [14], defect detection [15], and fraud detection. Class imbalance is a long-lasting challenge for the DL research community, and it impacts the convergence of DL models in the training process and the generalization on the test set [13,16,17]. We refer interested readers to the work [13] as it covers many aspects of class imbalance.

Many data-level solutions are proposed in the literature to alleviate the class imbalance problem, such as oversampling and undersampling. Those solutions try to artificially re-balance the datasets without improving the classifier's algorithm. On the other hand, algorithm-level techniques, such as thresholding, cost-sensitive learning [13], and reinforcement learning (RL) [18], try to improve the model generalizability.

The majority of research works in the literature aim to preserve the compressed models' accuracy. The authors in [9,19,20] show that compressing DL models does not harm model general accuracy. On the other hand, the work in [21] shows that model compression can actually hurt model accuracy and adversarial robustness.

Other works consider the class imbalance problem [22,23], proposing different approaches to alleviate the effect of the class imbalance problem in the training set. For example, instance reweighting [24] and customized loss functions [17].

The work in [25] investigates the impact of class imbalance on the per-class performance and fairness in pruned models. In [10], the authors highlight the impact of pruning and knowledge distillation on the OOD robustness for pretrained language models and propose a two-stage regularization method to improve the OOD robustness of compressed models.

The DL research community has overlooked the class imbalance robustness of compressed DL models, as most works in the literature focus either on the model compression aspect or the class imbalance aspect. Our work tries to bridge the gap between model compression and class imbalance robustness by investigating the effect of class imbalance on the robustness of compressed DL models.

Using popular model compression techniques, we compress DL models with different compression ratios, train them on many variants of class-imbalanced CIFAR-10, and analyze their robustness. Our work is a necessary step toward building robust, compressed DL models.

Our contributions can be summarized as follows:

- We present an in-depth literature review of the relevant work.
- We conduct a systematic study to evaluate and analyze the robustness of DL models when compressed using three popular techniques: pruning, quantization, and knowledge distillation. These compressed models are trained on the class-imbalanced CIFAR-10 dataset with varying degrees of imbalance.
- We provide results and insights from our extensive experiments on selected DL models. The insights presented in this work are intended to advance future research efforts in addressing the issue of class imbalance robustness in compressed DL models.

The rest of this paper is organized as follows: Section 2 motivates our work. Section 3 presents works related to the problem of robustness of compressed DL models. Section 4 describes the experimental setup. The results are presented in Section 5. Section 6 provides a discussion of the main findings of this paper. Section 7 concludes the paper.

2. Motivation

DL model compression is becoming more urgent because DL models have become prevalent almost everywhere, especially in resource-constrained environments such as edge devices, microcontrollers, smartphones, etc. On the other hand, real-world data show

imbalanced distributions of labeled samples among classes [16,26,27], which can harm the performance of DL models on the less represented classes (minority classes). However, reducing model size should not be accomplished at the expense of model robustness. We are motivated by the fact that the problem of compressed DL models' robustness when trained on class-imbalanced datasets has been under-explored by the DL research community.

Model compression and class imbalance robustness go hand in hand in many real-life scenarios. For instance, deploying machine learning on Internet of Things (IoT) devices requires particularly small models, as IoT devices have extremely small memory and limited computing capacity [28]. Bridging the gap between such resource-limited IoT devices and over-parameterized DL models requires compressing those models to fit into the available resources. Edge computing is another example where model compression is needed so an edge computing node can perform inference without much need to transfer data to the cloud [29]. Therefore, to successfully deploy compressed DL models in such environments, they must be robust against common perturbations, such as class imbalance in training datasets.

Federated learning is another domain where robust, compressed DL models are needed. In federated learning, a model is trained collaboratively by being shared by remote devices where training datasets are stored. Then each device trains its copy of the model on its own locally stored training dataset, and only updates are sent to the cloud to be integrated into the original model [30]. Compressing the DL models is of vital importance so that updates can be shared at scale without incurring bandwidth shortages or latency issues. Furthermore, in federated learning, there is no guarantee that the training datasets will be class-balanced because the data are distributed at the edge and cannot be examined for class imbalance by other parties except the owner of that dataset [31]. Therefore, compressed DL models should be robust against class imbalance in datasets to ensure that the models can incorporate updates from remote devices without worrying about the class imbalance of non-shared datasets.

DL models deployed in safety-critical systems, such as autonomous cars [32,33], can be vulnerable to class imbalance in training datasets [34–37]. DL models deployed in such systems should be robust against class imbalance when compressed in order to classify/detect different objects/agents with equally high accuracy.

3. Related Work

The DL research community has recognized the importance of having robust, compressed DL models. Nonetheless, earlier works have focused mainly on adversarial robustness and out-of-distribution (OOD) robustness. Other works have approached the problem of class imbalance robustness, but only for dense DL models. We summarize below the main trends in approaching the robustness problem of DL models in the literature.

The first group of works focuses on improving the adversarial robustness of DL models. Adversarial robustness means how robust DL models are against adversarial examples, which are obtained by adding little perturbations onto benign examples designed purposefully to confuse the classifier [21,38]. Stability training, a form of adversarial training, suggested in [39], improves adversarial robustness by training DL models on many distorted versions of the input data. To overcome the limitations of traditional adversarial training, model transformation is proposed [40], in which the original classifier is transformed into an isomorphic regression model whose loss function is more sensitive to small perturbations in the input data. In [41], the authors point out that self-supervised learning can enhance model robustness against adversarial examples, label corruption, common input corruptions, and out-of-distribution inputs. This group of works evaluates their proposed methods on full-size models, so they neither consider model compression nor class imbalance.

Some works take model compression into account when evaluating adversarial robustness. The authors in [21] propose a framework for simultaneous adversarial training and weight pruning to improve adversarial robustness. Also in [21], the authors conclude that

model compression (pruning in their case) is necessary to maintain an adversarially robust DL model as opposed to training a small-size model from scratch. In [42], the authors highlight that DL models trained using backpropagation exhibit robustness to various weight distortions, including quantization.

The authors in [9] demonstrate that model compression, particularly when involving pruning followed by optional quantization, can improve OOD model robustness. They find that lottery-ticket-style methods are particularly useful for producing compressed and OOD robust models. The authors in [10] show that compressed natural language processing (NLP) models are less robust than their original counterparts for OOD datasets. The authors in [43] show that severely compressed large models are more robust than mildly compressed small models.

Some works consider class imbalance when evaluating model robustness. In [26], the authors find that tuning key training parameters enables DL models to achieve state-of-the-art accuracy for the minority classes. Nonetheless, these works do not cover compressed DL models.

Two recent works [25,44] investigate the performance of compressed models with lightly imbalanced datasets. In [25], the author investigates the effect of pruning on per-class performance. However, that work is limited to one compression technique (pruning). Moreover, they do not consider severe imbalance ratios and large numbers of minority classes, which is the case in real-world applications [45]. Instead, they rely on datasets that are not perfectly balanced, such as QMNIST. The authors in [44] consider evaluating compressed models against the GTSBR dataset [46], a randomly imbalanced dataset with an imbalance ratio of 10. Despite being the closest work to our work, GTSBR is not severely imbalanced, and the authors do not consider different imbalance settings and types. The authors also have not included popular compression techniques, such as structured pruning, quantization-aware training, and knowledge distillation. Additionally, the authors' use of a mildly imbalanced dataset and relatively small models made it hard to show the effects of model compression on the class imbalance robustness of the DL model. We take the problem to the next level and present more insights and deeper observations.

In summary, state-of-the-art works have not fully explored the impact of class imbalance on the robustness of compressed deep learning models. Our work aims to study this problem, analyze the performance of compressed DL models trained on imbalanced datasets, and provide deep insights to motivate future research to provide real solutions to improve the class imbalance robustness of compressed DL models. Table 1 shows the main trends in DL model robustness.

Table 1. No work addresses the problem of class imbalance robustness of compressed DL models.

	Robustness			
Work	**Adversariel**	**OOD**	**Class Imbalance**	**Model Compression**
[38–40]	✓	×	×	×
[41]	✓	✓	×	×
[9,10]	×	✓	×	✓
[21,43]	✓	×	×	✓
[26,42]	×	×	✓	×
[25,44]	×	×	mild imbalance only	Pruning only

4. Experimental Setup

In this section, we will describe the experimental setup and the methodology we follow to expose the negative effects of model compression on the class imbalance robustness of DL models. We will discuss the models we use, the training datasets, the class imbalance types and imbalance ratios, the model compression techniques used in our experiments, and the tuning of their different parameters. All the experiments are conducted on a CentOS machine with one V100 GPU, using Distiller [47], a Python package for DL model compression research and analysis.

4.1. DL Models Used

In our experiments, we utilize ResNet-20 and ResNet-56 DL models [48], with ResNet-56 serving as the larger model. The ResNet architecture is well-regarded for its depth and use of residual connections, which enable the training of deep networks without the vanishing gradient problem [49]. By employing ResNet-20 and ResNet-56, we aim to thoroughly evaluate the robustness of compressed DL models under different class imbalance scenarios. Doing so will help us gain insights into how model size and complexity influence resilience to class imbalance when applying compression techniques.

4.2. Dataset Used

We use the CIFAR-10 dataset in all our experiments. CIFAR-10 is a balanced dataset with 10 classes, each containing 5000 32×32 color images for training and 1000 32×32 color images for testing. The multiclass architecture and moderate size of CIFAR-10 make it an ideal choice for our research goals, as it allows for various settings of class imbalance. Additionally, it makes the training process more time-efficient. To generate class-imbalanced versions of CIFAR-10, we utilize an implementation provided by [17].

4.3. Imbalance Settings

To produce class-imbalanced versions of CIFAR-10, we follow the imbalance procedure suggested in [13]. We perform two types of training in terms of class imbalance: balanced CIFAR-10 and imbalanced CIFAR-10. For balanced CIFAR-10, we train DL models with the standard balanced CIFAR-10. For imbalanced CIFAR-10, we train the models for step imbalance and linear imbalance.

For step imbalanced CIFAR-10, we have five different imbalance ratios $\rho = \{2, 10, 20, 50, 100\}$. The imbalance ratio can be defined as follows:

$$\rho = \frac{Max.\, number\, of\, samples}{Min.\, number\, of\, samples}. \tag{1}$$

For each imbalance ratio, we set the fraction of minority classes (μ) to have three values: $\mu = \{2, 5, 8\}$. The number of samples per class i for step-class imbalance is calculated as follows:

$$Samples = Max.\, number\, of\, samples \times (\rho)^{-1}. \tag{2}$$

For linearly imbalanced CIFAR-10, we will use the same five different imbalance ratios as for the step imbalance (μ is not applicable in the case of linear imbalance). The number of samples per class i for linear class imbalance is calculated as follows:

$$Samples = Max.\, number\, of\, samples \times (\rho)^{\frac{class\, i\, index}{num.\, classes - 1.0}}. \tag{3}$$

In total, we will have 20 different variants of class-imbalanced CIFAR-10. Figure 1 shows examples of the imbalance distributions of CIFAR-10 used in our experiments.

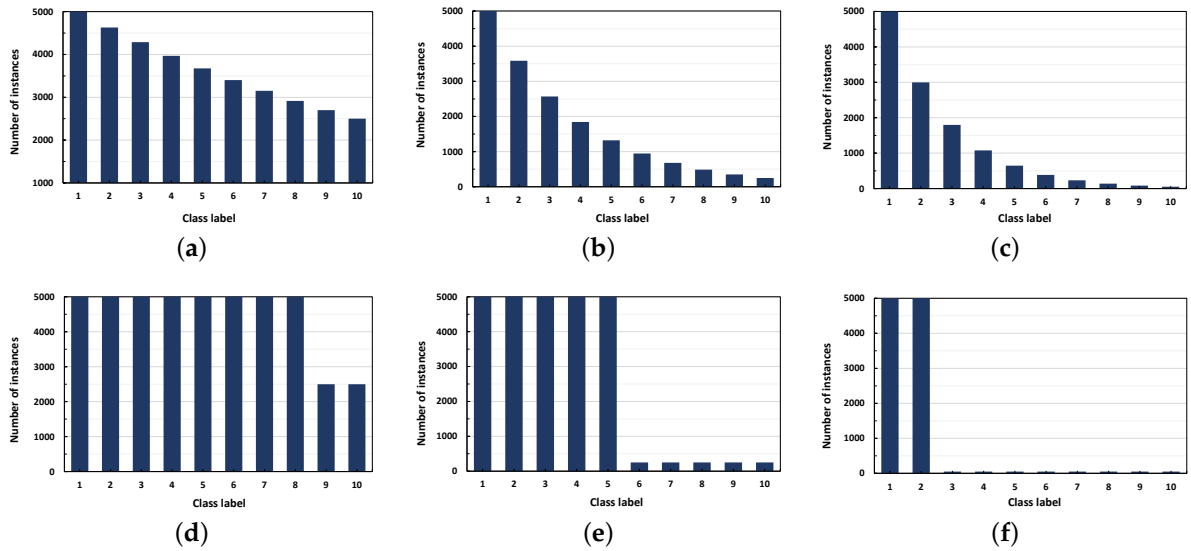

Figure 1. Examples of class imbalance settings in CIFAR-10. (**a**) Linear imbalance, $\rho = 2$; (**b**) Linear imbalance, $\rho = 20$; (**c**) Linear imbalance, $\rho = 100$; (**d**) Step imbalance, $\mu = 2$, $\rho = 2$; (**e**) Step imbalance, $\mu = 5$, $\rho = 20$; (**f**) Step imbalance, $\mu = 8$, $\rho = 100$.

4.4. Filter Rank Pruning: Baselines and Settings

Pruning involves removing redundant weights (unstructured pruning) or filters (structured pruning) according to a specific criterion [50,51]. Pruning can significantly reduce model size, memory footprint, energy consumption, and inference time [52]. A main drawback of pruning is the reduced accuracy, especially with high degrees of pruning [10]. Pruning is typically performed in three stages: training, pruning, and retraining (fine-tuning) [3]. Retraining pruned DL models could enable them to escape a previous local minima, and that could improve performance [53].

We choose Filter Rank Pruning (FRP) [54], a structured pruning technique where entire filters along with their corresponding feature maps are pruned according to their relative importance. The relative importance of a filter is determined by calculating ℓ_1-norm, which is the sum of its absolute weights [54]. We refer interested readers to this paper [54] for more details on filter pruning.

For FRP, our baseline model is unpruned ResNet-20, trained on balanced CIFAR-10, and the accuracy obtained here is the reference accuracy for the FRP part of the experiments. For ResNet-20 and ResNet-56, we prune away filters in the 1st convolutional layer of each residual block (a total of 9 convolutional layers pruned; the first and last layers are excluded). To achieve different levels of sparsity, we use three pruning ratios: mild, moderate, and severe. Mild pruning, moderate pruning, and severe pruning yield compressed models with 15%, 71%, and 97% sparsity, respectively.

The baseline model and the pruned models are trained on CIFAR-10 and on the imbalanced versions of CIFAR-10 for 180 epochs with the learning rate set to 0.1.

4.5. Quantization-Aware Training: Baselines and Quantization Parameters

In quantization, weights and activations are represented by lower bit-widths (integer precision) such as 8, 4, or 2 bits instead of 32 bit-widths (floating point precision, or FP32), and this can speed up training and inference and effectively reduce model size while retaining the FP32 model's original structure [55,56]. One of the main drawbacks of quantization is dropped performance when using ultra-low bit-widths, which may require more advanced techniques [57].

There are two approaches to applying quantization to DL models: post-training quantization (PTQ) and quantization-aware training (QAT). PTQ uses calibration data (a small portion of the training dataset) to learn the clipping ranges and the scale factors,

then quantizes a pre-trained model based on the calibration outcome. In QAT, a pre-trained model is quantized and then re-trained (fine-tuned) on the entire training dataset [7,58]. Quantization is done by mapping the continuous values of weights and activations (float32) to discrete values $[0, 2^b - 1]$ for activations and $[-2^{b-1}, 2^{b-1} - 1]$ for weights, where b is the desired lower bit-width. The quantizer function $Q_b(.)$ that quantizes the continuous float32 values of weights and activations v to quantized values of b bit-width v^q can be expressed as [59,60]:

$$v^q = Q_b(v; s) = round(clip(fracvs, min_b, max_b)) \times s, \quad (4)$$

QAT is a preferred way to implement quantization because it aims to avoid accuracy loss originating from lower precision, such as INT4, by replacing weights and activations with the quantized ones during the training process [60,61].

The baseline model for QAT (Baseline FP32) is FP32 ResNet-20 and ResNet-56 trained on CIFAR-10. To obtain the quantized versions of DL models, we use DoReFa quantizer [62]. We quantize the models with activations bit-widths of 8 and weights bit-widths of 4 (Quantized (A8, W4)). We create another quantized DL model with activations bit-widths of 3 and weights bit-widths of 3 (Quantized (A3, W3)). Our choice to use these bit-widths follows [63]. The first and last layer are not quantized as they are sensitive to weight quantization [64]. We use a weight decay value of 0.0002. For the rest of this paper, we will refer to the first quantized model as QAT (A8, W4) and to the second quantized model as QAT (A3, W3).

The QAT baseline model and the quantized models are trained on CIFAR-10 and the imbalanced versions of CIFAR-10 for 200 epochs with the learning rate set to 0.1.

4.6. Knowledge Distillation: Baselines and Distillation Parameters

Knowledge distillation (KD) aims to transfer knowledge from a larger model (a.k.a. the teacher model) to a smaller model (a.k.a. the student model) such that the student model can mimic the performance of the teacher model [65–67]. The knowledge transfer process involves dividing the logits by a parameter called *temperature* before feeding them to the softmax layer. A higher temperature value boosts the activations of the incorrect classes, promoting more information to flow to the student model during backpropagation [8]. KD offers the flexibility of choosing a student model from a different family/size than the teacher model [8]. However, KD has some shortcomings, such as how to choose the right teacher model for the given task and how to decide the shallowness of the student model [68].

There are two approaches for performing KD to produce student models. The first approach is to choose a student model with a smaller architecture than the teacher model's architecture. For example, the teacher model could be a pre-trained ResNet-56, and the student model could be a ResNet-18. In the second approach, the student model is a compressed version, hence smaller, of the teacher model. For example, an FP32 ResNet-56 is the teacher model, while a quantized ResNet-56 is the student model. We opt to follow the latter approach since it aligns better with our research goals of evaluating compressed model robustness [66].

For our KD experiments, we use FP32 ResNet-20 or ResNet-56 as the teacher model (Teacher FP32). For the student models, we use quantized versions of the teacher models. Specifically, the first student model is obtained by quantizing the teacher model with activations and weights with bitwidths of 8 and 4, respectively. The second student model is obtained by quantizing the teacher model with activations and weights bitwidths of 3 and 3, respectively. We use QAT with DoReFa quantizer for obtaining the student models. For the rest of this paper, we refer to the first student model as Student (A8, W4), and the second student model as Student (A3, W3).

We follow the approach outlined in [47] to set the distillation parameters: the weight for the distillation loss is set to 0.7, the weight for the student versus label loss is set to 0.3, and the softmax temperature is set to 1. Both the teacher model and the student models are

trained on CIFAR-10 and its imbalanced versions for 200 epochs, with the learning rate set to 0.1.

4.7. Evaluation Metrics

We use a similar metric as in [10] for model robustness evaluation. We calculate the performance gap as the percent ratio between the macro F-1 score of the model trained on balanced CIFAR-10 and the macro F-1 score of the model trained on class-imbalanced CIFAR-10 as

$$performance\ gap = \frac{F\text{-}1\ score_{bal} - F\text{-}1\ score_{imbal}}{F\text{-}1\ score_{bal}}. \quad (5)$$

For the uncompressed model, the performance gap is denoted as $\Delta_{uncompressed}$, and for the compressed model, the performance gap is denoted as $\Delta_{compressed}$. If $\frac{\Delta_{compressed}}{\Delta_{uncompressed}}$ is < 1, then the model is not robust. Otherwise, it is robust.

F1 scores can be calculated as follows:

$$F1\ score = \frac{Recall + Precision}{Recall * Precision}, \quad (6)$$

where

$$Recall = \frac{Truepositive}{True\ positive + False\ negative} \quad (7)$$

and

$$Precision = \frac{True\ positive}{True\ positive + False\ positive}. \quad (8)$$

F1 scores are more suitable for imbalanced datasets than accuracy because F1 scores show per-class performance. Aggregating per-class F1 scores by taking the unweighted mean produces macro F1 score. Macro F1 score gives the same importance to all classes (majority and minority classes), which makes it a more realistic metric for imbalanced datasets as it gives under-represented (i.e., minority) classes the same importance given to majority classes.

5. Results

We conduct various experiments to evaluate the robustness of compressed DL models against class imbalance. To ensure more stable results, each model is trained three times [69], and we compute the average of the macro F1 scores obtained across these three runs.

To explore whether our findings with ResNet-20 apply to larger DL models, we selected ResNet-56 for further experimentation. Using the same experimental setup, we conducted a series of tests to determine whether the patterns observed with ResNet-20 would hold for a more complex model.

5.1. Filter Rank Pruning Results

ResNet-20. Figure 2 shows the performance gaps (in percentage) between unpruned ResNet-20 and pruned ResNet-20 across different class imbalance settings.

The results show that the severely pruned ResNet-20 is not robust against class imbalance across all the class imbalance settings. The performance gap widens as the imbalance ratio (ρ) increases. For step imbalance with high ρ, the performance gaps range from 85% to 98%, indicating that the pruned model is extremely vulnerable to high degrees of class imbalance. A similar observation, though to a lesser extent, applies to linear imbalance, with a performance gap of 56% for ρ = 100.

For moderate pruning, the model also exhibits a lack of robustness. The drop in robustness becomes more pronounced starting at ρ = 10 and above. However, for ρ = 2 and small numbers of minority classes such as μ = 2 and 5, it appears that the moderately pruned model can handle slight class imbalance.

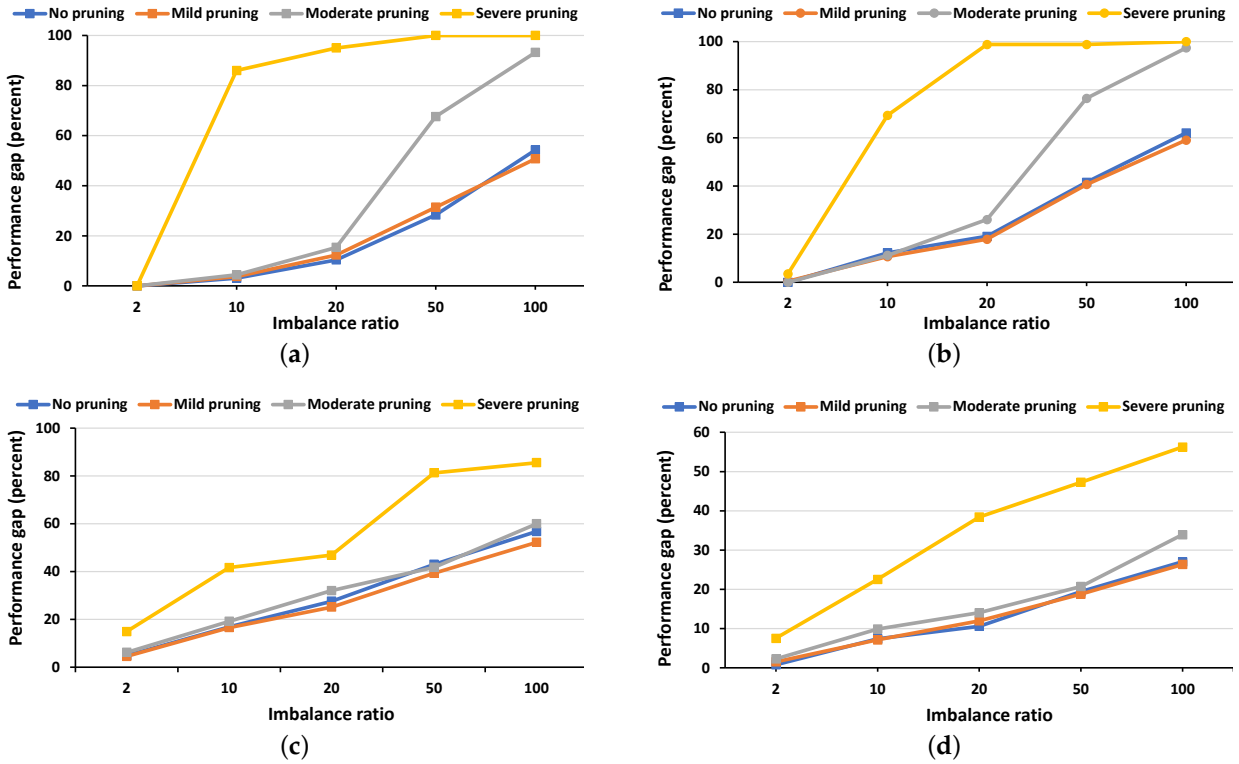

Figure 2. Performance gaps of Resnet-20 and its pruned versions trained on the imbalanced CIFAR-10. (**a**) Step imbalance $\mu = 2$; (**b**) Step imbalance $\mu = 5$; (**c**) Step imbalance $\mu = 8$; (**d**) Linear imbalance.

For mild pruning, ResNet-20 experiences fluctuations as $\frac{\Delta_{compressed}}{\Delta_{uncompressed}}$ is not always < 1. We believe the reason for the model showing robustness is that mild pruning can help the model generalize better. This phenomenon has been observed by other researchers [9,52,70]. Tables A1 and A2 in Appendix A show the results in numbers for step imbalance and class imbalance, respectively.

ResNet-56. Table 2 summarizes the model pruning results, showing the performance gaps between the pruned ResNet-56 and its unpruned counterparts with different imbalance settings.

Table 2. Performance gaps in F1 scores of the pruned ResNet-56 compared with its unpruned counterparts trained on imbalanced datasets. For each value of ρ, we show three F1 drop percentages corresponding to the three μ values (2, 5, and 8, respectively).

ρ	Unpruned	Mildly Pruned	Moderately Pruned	Severely Pruned
2	3.31%, 3.99%, and 6.19%	2.74%, 2.74%, and 5.33%	1.5%, 3.16%, and 4.97%	5.3%, 9.18%, and 14.52%
10	4.01%, 11.94%, and 31.62%	3.23%, 11.14%, and 27.53%	3.89%, 11.80%, and 29.12%	8.58%, 52.33%, and 43.01%
20	5.91%, 21.76%, ans 55.13%	5.52%, 20.71%, and 41.03%	5.01%, 19.36%, and 44.04%	21.15%, 52.33%, and 67.90%
50	13.69%, 47.99%, and 83.56%	11.31%, 43.36%, and 82.29%	12.82%, 44.23%, and 81.22%	29.13%, 65.75%, and 85.19%
100	16.90%, 49.24%, and 83.56%	14.89%, 45.17%, and 79.48%	15.82%, 48.38%, and 84.74%	30.46%, 64.08%, and 85.85%

For severe pruning (97% sparsity), the pruned ResNet-56 exhibits significant performance drops compared with the unpruned ResNet-56, indicating that the severely pruned ResNet-56 is not robust against class imbalance.

As for mild and moderate pruning, our experiments show no drop in robustness for ResNet-56. When pruned at these pruning levels, ResNet-56 appears to be robust against class imbalance. This finding aligns with studies showing that pruned DL models, when applied to over-parameterized DL models, can maintain or even surpass the accuracy of their unpruned counterparts [52,70]. The authors in [9] also found that pruning can sometimes improve OOD robustness, indicating improved model generalization, including handling class imbalance, after pruning. Table A7 in Appendix B shows a comparison of the model size for unpruned and pruned models.

5.2. Quantization-Aware Training Results

ResNet-20. Figure 3 compares between FP32 ResNet-20 and its QAT versions across different class imbalance settings.

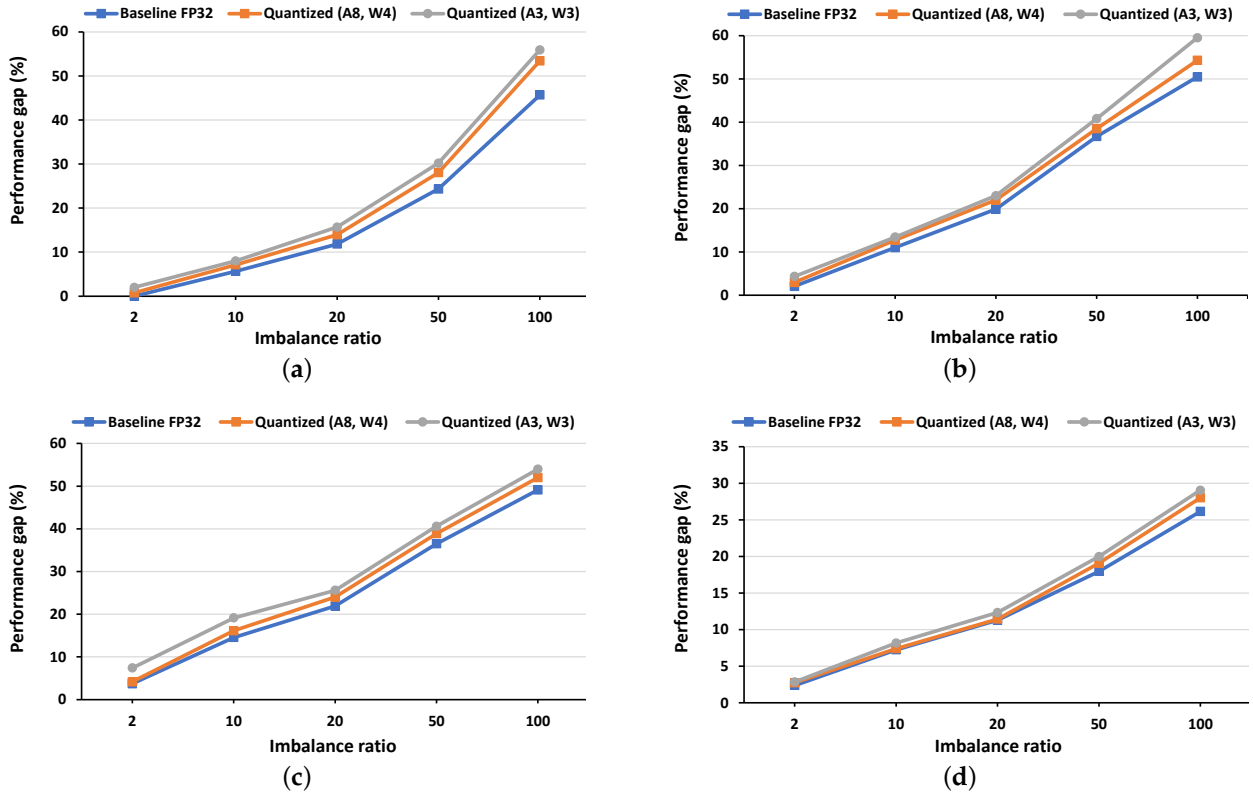

Figure 3. Performance gaps between Resnet-20 and its QAT versions trained on imbalanced CIFAR-10. (**a**) Step imbalance $\mu = 2$; (**b**) Step imbalance $\mu = 5$; (**c**) Step imbalance $\mu = 8$; (**d**) Linear imbalance.

For step imbalance with $\mu = 2$, as illustrated in Figure 3a, the performance gap between ResNet-20 and the quantized ResNet-20 models increases as the imbalance ratio (ρ) increases, with the largest gap observed at $\rho = 100$.

Figure 3b shows the performance gap between ResNet-20 and the quantized ResNet-20 models for step imbalance $\mu = 5$. Although the performance gap is not significant until $\rho > 20$, it indicates that the robustness worsens as ρ increases.

For $\mu = 8$, Figure 3c shows that the performance gap between ResNet-20 and the quantized ResNet-20 models worsens (becomes larger) as early as $\rho = 2$, and it continues to worsen as (ρ) increases. This is likely due to the fact that 8 of the classes are minority classes, which the quantized ResNet-20 struggles to handle effectively.

Figure 3d shows the performance gap between ResNet-20 and the quantized ResNet-20 models when trained on linear class imbalance. The performance gap grows as ρ increases. Because linear class imbalance is different from step class imbalance, as it involves all the classes (see Equation (3)), the performance gaps for linear imbalance are slightly smaller compared with those for step imbalance, especially for $\rho = 2$. However, they clearly indicate that quantized ResNet-20 models are not robust against linear class imbalance.

The performance gaps indicate that the more severe quantization (i.e., QAT (A3, W3)) has a greater negative impact on robustness than conservative quantization (i.e., QAT (A8, W4)). In conclusion, our results show that the quantized ResNet-20 models, whether QAT (A8, W4) or QAT (A3, W3), are not robust against class imbalance. Tables A3 and A4 in Appendix A show the results in numbers for step imbalance and class imbalance, respectively.

ResNet-56. Table 3 shows the performance gaps between FP32 ResNet-56 and its QAT (A8, A4) counterpart when trained on class-imbalanced CIFAR-10. The results show that QAT ResNet-56 is not robust against class imbalance, as it experiences larger performance gaps than the FP32 ResNet-56 when both are trained on the same class-imbalanced dataset. Specifically, as the class imbalance ratio (ρ) increases, the performance of the QAT ResNet-56 degrades significantly compared with the FP32 ResNet-56. This trend is consistent across various imbalance settings, indicating that the quantization process adversely impacts the model's ability to handle class imbalance. For instance, under severe imbalance conditions, the performance gap widens considerably, demonstrating the vulnerability of QAT models to class imbalance. Table A4 in Appendix A shows a comparison of the model size for FP32 and QAT models.

Table 3. Performance gap percentages of the quantized model compared with the FP32 model trained on step-imbalanced datasets. For each value of ρ, we show three accuracy drop percentages, corresponding to the three μ values (2, 5, and 8, respectively).

ρ	FP32 ResNet-56	Quantized (A8, W4) ResNet-56
2	1.14%, 1.68%, 3.1	1.24%, 1.87%, 3.56%
10	6.85%, 10.54%, 14.57%	6.98%, 11.37%, 16.97%
20	12.98%, 18.67%, 22.29%	14.19%, 20.29%, 26.05%
50	22.01%, 36.07%, 36.44%	25.46%, 37.4%, 41.51%
100	37.69%, 53.37%, 44.4%	40.6%, 55.47%, 54.05%

5.3. Knowledge Distillation Results

ResNet-20. Figure 4 illustrates the knowledge distillation experiment results. It shows the performance gaps between the teacher and the student models across different imbalance settings.

Figure 4a shows the performance gaps when $\mu = 2$. The performance gap between the teacher and the student models becomes larger as ρ increases. The performance gaps are more noticeable when $\rho > 20$ than for lower ρ values because we have only 2 minority classes.

For $\mu = 5$, Figure 4b shows that the performance gaps between the teacher and the student models become more significant when $\rho \geq 10$. This behavior is attributed to the fact that half of the classes are now minority classes, making it harder for the student models to overcome this imbalance.

For $\mu = 8$, the performance gaps between the teacher and the student models shown in Figure 4c demonstrate that even for $\rho = 2$, the gaps are significant. This behavior is attributed to the fact that 8 of the classes are minority classes.

For the linear class imbalance, the performance gaps are apparent even for values of ρ just above 2. The performance gap grows as ρ increases. Notably, the performance gaps here are more significant than those for step class imbalance at the same values of ρ.

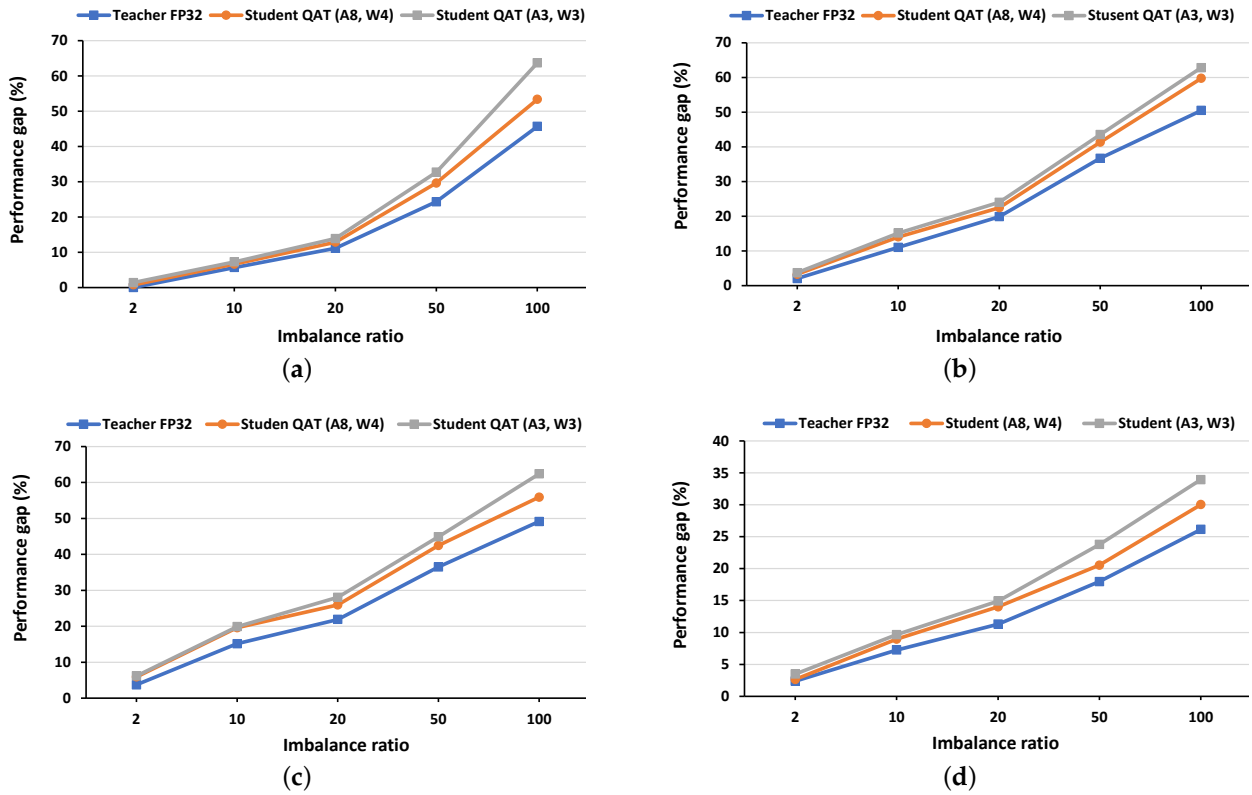

Figure 4. Performance gaps between the teacher model and the student models, trained on imbalanced variants of CIFAR-10. (**a**) Step imbalance $\mu = 2$; (**b**) Step imbalance $\mu = 5$; (**c**) Step imbalance $\mu = 8$; (**d**) Linear imbalance.

From the performance gap comparisons presented above, it is clear that the student models are not robust against class imbalance. Furthermore, our results show that the smaller the student model, the less robust it becomes against class imbalance. Tables A5 and A6 in Appendix A show the results in numbers for step imbalance and class imbalance, respectively.

ResNet-56. Table 4 presents the performance gaps between the teacher model (FP32 ResNet-56) and the student model (QAT ResNet-56 (A8, W4)) with different imbalance settings. The results reveal that the student model struggles with class imbalance, as evidenced by its larger performance gaps compared with the teacher model when both are trained on the imbalanced CIFAR-10 dataset. As the class imbalance ratio (ρ) increases, the student model's performance declines more than that of the teacher model. This trend is consistent across various imbalance settings, suggesting that the knowledge distillation process diminishes the student model's robustness to class imbalance. In scenarios with high imbalance, the student model's performance gap becomes particularly significant, underscoring its greater vulnerability to class imbalance compared with the teacher model.

Table 4. Performance gap percentages of the student model compared with the teacher model trained on step-imbalanced datasets. For each value of ρ, we show three accuracy drop percentages, corresponding to the three μ values (2, 5, and 8, respectively).

ρ	Teacher Model	Student QAT (A8, W4) Model
2	0.2%, 1.53%, 3.02	3.25%, 1.77%, 3.67%
10	2.25%, 8.85%, 14.23%	2.61%, 10.39%, 15.93%
20	4.51%, 16.07%, 23.70%	4.85%, 16.15%, 24.4%
50	7.51%, 25.47%, 35.75%	7.76%, 27.22%, 39.18%
100	11.07%, 33.59%, 44.76%	12.79%, 34.74%, 49.54%

6. Discussion

This paper investigates the robustness of compressed DL models in the presence of class imbalances in training datasets. We informally define the empirical class imbalance robustness as the compressed model's ability to preserve the original accuracy when the training dataset is class-imbalanced. Most works in the literature have overlooked the issue of class imbalance robustness in compressed DL models. Our study aims to fill this gap, shedding light on this critical and often neglected aspect of model performance.

Our results indicate that compressed DL models are not robust against class-imbalanced datasets, whether the compression method involves pruning, quantization, or knowledge distillation. This finding has significant implications for the deployment of compressed models in real-world applications, where class imbalance is common. It suggests that practitioners should be cautious when applying compression techniques in such scenarios.

Furthermore, our experiments reveal that large-scale DL models, when subjected to severe pruning, exhibit reduced robustness against class imbalance. Additionally, quantized models struggle to handle class imbalance as effectively as their full-precision counterparts. Moreover, student models resulting from knowledge distillation do not achieve the same level of robustness as their teacher models. This finding suggests that the popularity of compressed DL models might be limited by their inability to maintain robustness in the face of class imbalance. Tables 2–4 show that for a heavy-weight model such as ResNet-56 in our case, different compression techniques yield different robustness levels. For example, mild and moderate pruning, Table 2, do not incur a robustness drop. QAT, Table 3, seems to cause less robustness drop than KD; see Table 4. This hints that one can carefully decide which compression technique to use to trade-off between model size and model robustness.

We observe that various compression techniques result in differing levels of class imbalance robustness. Notably, KD produces nearly the same level of robustness as QAT. These conclusions are based on the specific experimental settings of our study. Future work will explore whether these findings hold across different model compression configurations and class imbalance scenarios. Additionally, we plan to study existing techniques and develop new methods to enhance the robustness of compressed DL models against class imbalance.

We note that the number of minority classes (μ) significantly affects a DL model's performance on the given dataset. In our experiments, when $\mu = 8$, the performance gaps are larger compared with when $\mu = 2$. This trend is also observed for linear imbalance, where the class distribution is long-tailed. Additionally, increasing imbalance ratios further deteriorate robustness. High levels of class imbalance and numerous minority classes result in critically low robustness levels.

Data rarity [71], an inherent attribute in class-imbalanced datasets, especially in severe class imbalances, incurs a performance drop even in uncompressed DL models. However, model compression also worsens the performance drop due to the removal of model overparametrization. As highlighted in *the law of robustness* [72], large DL models are more robust than small ones. In the context of model compression, pruning (trimming connections and removing entire filters), quantization (using lower precision to represent

weights and activations), and knowledge distillation (using a smaller model), the DL models transform from large (i.e., uncompressed) models to small (i.e., compressed) models, hence making them less robust. These reasons together may explain the drop in the class imbalance robustness of compressed DL models.

In our work, we use three methods for model compression: structured filter rank pruning, quantization-aware training, and knowledge distillation. We believe these popular methods are sufficient for the goals of our study and have chosen not to include other compression techniques such as unstructured pruning, post-training quantization, and regularization. We use CIFAR-10 as our dataset. CIFAR-10 is a midsize multiclass dataset, making it a suitable choice for time-efficient training and inference. Additionally, it allows for more configurations of class imbalance. For future work, we will consider other DL models and datasets that cover other domains in DL, such as NLP, where class imbalance is also a significant concern.

Comparative analysis with existing studies shows that our findings align with some reported phenomena, such as pruned DL models maintaining or even outperforming unpruned models in terms of accuracy. However, the impact of class imbalance on robustness remains under-explored and presents a valuable area for further research. Our work sets the stage for more comprehensive studies that could lead to the development of more resilient model compression techniques.

7. Conclusions

Model compression techniques should reduce model size without compromising robustness; however, class imbalance, a well-known issue in real-life training datasets, complicates this goal. We have investigated the challenges that class imbalance poses to the robustness of compressed DL models. Our empirical results demonstrate that model compression techniques leave compressed models vulnerable to the detrimental effects of class imbalance. We conclude that when compressed DL models are trained on class-imbalanced datasets, their robustness suffers significantly, regardless of the compression method used. Therefore, effective solutions are needed to address this problem. This work aims to inspire future research to develop better solutions for enhancing the class imbalance robustness of compressed DL models.

Author Contributions: B.S.A., N.S., and M.A. developed the analytical methods presented in the paper; B.S.A. wrote the paper; Illustrations were generated by B.S.A.; experimental results were recorded by B.S.A.; Analysis of experiments was performed by B.S.A., N.S., and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Results Tables

We provide below the results we presented in Section 5 as numbers for readers interested in reproducing our results. The results show that compressed DL models are not robust to class imbalance in training datasets, especially severe step class imbalance and long-tailed class imbalance.

Appendix A.1. Filter Rank Pruning

Tables A1 and A2 show the drop in robustness for pruned ResNet-20 trained on imbalanced CIFAR-10. Again, we notice that the robustness drop increases as the pruning ratio increases for a given imbalance setting. This is more noticeable for moderate and severe pruning with bigger imbalance ratio (ρ).

Table A1. Performance gaps in F1 scores of the pruned ResNet-20 compared with its unpruned counterparts trained on imbalanced datasets. For each value of ρ, we show three F1 drop percentages corresponding to the three μ values (2, 5, and 8, respectively).

ρ	Unpruned	Mildly Pruned	Moderately Pruned	Severely Pruned
2	0%, 0%, and 4.85%	0%, 0.43%, and 4.5%	0%, 0%, and 6.19%	0%, 3.53%, and 14.87%
10	3.08%, 12.23% and 16.86%	3.73%, 10.64%, and 16.57%	4.4%, 11.19%, and 19.16%	86%, 69.36%, and 41.67%
20	10.36%, 19.07%, and 27.56%	12.29%, 17.89%, and 25.13%	15.35%, 26.07%, and 32.02%	94.99%, 98.82%, and 46.9%
50	28.33%, 41.56%, and 42.99%	31.39%, 40.61%, and 39.29%	67.61%, 76.42%, and 41.78%	100%, 98.82%, and 81.29%
100	54.24%, 62.07%, and 56.78%	50.71%, 59.05%, and 52.25%	93.21%, 97.38%, and 60%	100%, 100%, and 85.56%

Table A2. Performance gaps in F1 scores of the pruned ResNet-20 compared with its unpruned counterparts trained on linear imbalanced datasets.

ρ	Unpruned	Mildly Pruned	Moderately Pruned	Severely Pruned
2	0.77%	1.53%	2.26%	7.51%
10	7.38%	7.13%	9.88%	22.53%
20	10.58%	11.96%	14.04%	38.43%
50	19.4%	18.77%	20.71%	47.27%
100	27.01%	26.34%	33.92%	56.25%

Appendix A.2. Quantization-Aware Training

Tables A3 and A4 show the robustness drop for the quantized ResNet-20. One can see that for all imbalance settings, quantized ResNet-20 has suffered robustness drop, and the robustness drop becomes larger as the quantization becomes more aggressive and the imbalance ratios ρ get higher.

Table A3. Performance gaps in F1 scores of the quantized ResNet-20 compared with its FP32 counterparts trained on step imbalanced datasets. For each value of ρ, we show three F1 drop percentages corresponding to the three μ values (2, 5, and 8, respectively).

ρ	Baseline FP32	Quantized (A8, W4)	Quantized (A3, W3)
2	3.7%, 2.03%, and 3.7%	0.74%, 2.93%, and 4.17%	2%, 4.32%, and 7.39%
10	14.54%, 11%, and 14.54%	7.12%, 12.7%, and 16.13%	7.99%, 13.41%, and 19.13%
20	21.89%, 19.85%, and 21.89%	13.95%, 21.99%, and 24%	15.68%, 22.99%, and 25.58%
50	36.49%, 36.67%, and 36.49%	28.04%, 38.48%, and 38.89%	30.19%, 40.78%, and 40.59%
100	49.12%, 50.46%, and 49.12%	53.43%, 54.27%, and 52%	55.92%, 59.49%, and 53.99%

Table A4. Performance gaps in F1 scores of the quantized ResNet-20 compared with its FP32 counterparts trained on linear imbalanced datasets.

ρ	Baseline FP32	Quantized (A8, W4)	Quantized (A3, W3)
2	2.38%	2.74%	2.85%
10	7.25%	7.41%	8.17%
20	11.29%	11.45%	12.34%
50	17.96%	19.08%	19.99%
100	26.14%	28.02%	29.04%

Appendix A.3. Knowledge Distillation

Tables A5 and A6 show the robustness drop in the student model (QAT ResNet-20). We observe that the student model is not robust against class imbalance. This observation is in line with the observations for filter rank pruning and quantization-aware training.

Table A5. Performance gaps in F1 scores of the student ResNet-20 compared with its teacher model trained on step imbalanced datasets. For each value of ρ, we show three F1 drop percentages corresponding to the three μ values (2, 5, and 8, respectively).

ρ	Teacher FP32	Student QAT (A8, W4)	Student QAT (A3, W3)
2	0%, 2.03%, and 3.7%	0.73%, 3.26%, and 5.9%	1.37%, 3.69%, and 6.17%
10	5.65%, 11%, and 15.14%	6.63%, 14.01%, and 19.59%	7.27%, 15.13%, and 19.88%
20	11.14%, 19.85%, and 21.89%	12.85%, 22.44%, and 25.92%	13.88%, 23.98%, and 28.05%
50	24.33%, 36.67%, and 36.49%	29.61%, 41.3%, and 42.46%	32.7%, 43.5%, and 44.9%
100	45.67%, 50.46%, and 49.12%	53.36%, 59.74%, and 55.89%	63.71%, 62.79%, and 62.37%

Table A6. Performance gaps in F1 scores of the student ResNet-20 compared with its teacher counterparts trained on linear imbalanced datasets.

ρ	Teacher FP32	Student QAT (A8, W4)	Student QAT (A3, W3)
2	2.38%	2.63%	3.48%
10	7.25%	8.95%	9.65%
20	11.29%	14.01%	14.92%
50	17.96%	20.54%	23.78%
100	26.14%	30.03%	33.95%

Appendix B. Model Size

The primary goal of model compression is to reduce model size. We present in Table A7 the model size of the pruned Resnet-20 and Resnet-56. Table A8 shows the model size of quantized Resnet-56 (QAT A8, W4) and Resnet-20. For pruning, we report the number of non-zero (NNZ) parameters of the pruned model as well as the number of non-zero parameters of the original model for comparison. NNZ parameters indicate the effect of pruning in removing redundant weights and activations. For quantization, we

calculate the model size by multiplying the NNZ parameters by the bit-width, which gives the model size in Mbs.

Table A7. Model size of pruned ResNet-20. Numbers represent non-zero (NNZ) parameters.

Model Size	Unpruned	Mildly Pruned	Moderately Pruned	Severely Pruned
ResNet-20	268,336	227,296	77,248	6688
ResNet-56	851,504	722,336	245,696	21,344

Table A8. Model size of QAT ResNet-20. Numbers represent non-zero parameters (NNZ). The same table can be used for KD since the teacher model is a FP32 ResNet and the student model is a QAT ResNet.

Model Size (Mb)	FP32	QAT (A8, W4)	QAT (A3, W3)
ResNet-20	8.18	1.53	0.76
ResNet-56	25.98	4.87	N/A

References

1. Guo, Q.; Chen, S.; Xie, X.; Ma, L.; Hu, Q.; Liu, H.; Liu, Y.; Zhao, J.; Li, X. An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms. In Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, 11–15 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 810–822.
2. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
3. Liu, Z.; Sun, M.; Zhou, T.; Huang, G.; Darrell, T. Rethinking the value of network pruning. *arXiv* **2018**, arXiv:1810.05270.
4. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. A survey of model compression and acceleration for deep neural networks. *arXiv* **2017**, arXiv:1710.09282.
5. Alawad, M.; Lin, M. Scalable FPGA Accelerator for Deep Convolutional Neural Networks with Stochastic Streaming. *IEEE Trans. Multi-Scale Comput. Syst.* **2018**, *4*, 888–899. [CrossRef]
6. Frankle, J.; Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv* **2018**, arXiv:1803.03635.
7. Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M.W.; Keutzer, K. A survey of quantization methods for efficient neural network inference. *arXiv* **2021**, arXiv:2103.13630.
8. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, *2*, arXiv:1503.02531.
9. Diffenderfer, J.; Bartoldson, B.; Chaganti, S.; Zhang, J.; Kailkhura, B. A winning hand: Compressing deep networks can improve out-of-distribution robustness. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 664–676.
10. Du, M.; Mukherjee, S.; Cheng, Y.; Shokouhi, M.; Hu, X.; Hassan, A. Robustness Challenges in Model Distillation and Pruning for Natural Language Understanding. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 2–6 May 2023; pp. 1758–1770.
11. Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; Madry, A.; Kurakin, A. On evaluating adversarial robustness. *arXiv* **2019**, arXiv:1902.06705.
12. Kim, W.J.; Cho, Y.; Jung, J.; Yoon, S.E. Feature Separation and Recalibration for Adversarial Robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 8183–8192.
13. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2018**, *106*, 249–259. [CrossRef]
14. Yang, J.; El-Bouri, R.; O'Donoghue, O.; Lachapelle, A.S.; Soltan, A.A.; Eyre, D.W.; Lu, L.; Clifton, D.A. Deep reinforcement learning for multi-class imbalanced training: Applications in healthcare. *Mach. Learn.* **2024**, *113*, 2655–2674. [CrossRef] [PubMed]
15. Wang, Y.; Chung, S.H.; Khan, W.A.; Wang, T.; Xu, D.J. ALADA: A lite automatic data augmentation framework for industrial defect detection. *Adv. Eng. Inform.* **2023**, *58*, 102205. [CrossRef]
16. Yang, Y.; Xu, Z. Rethinking the value of labels for improving class-imbalanced learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19290–19301.
17. Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8183–8192.
18. Lin, E.; Chen, Q.; Qi, X. Deep reinforcement learning for imbalanced classification. *Appl. Intell.* **2020**, *50*, 2488–2502. [CrossRef]
19. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* **2015**, arXiv:1510.00149.
20. Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Learning structured sparsity in deep neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2074–2082.

21. Ye, S.; Xu, K.; Liu, S.; Cheng, H.; Lambrechts, J.H.; Zhang, H.; Zhou, A.; Ma, K.; Wang, Y.; Lin, X. Adversarial robustness vs. model compression, or both? In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 111–120.
22. Lian, J.; Freeman, L.; Hong, Y.; Deng, X. Robustness with respect to class imbalance in artificial intelligence classification algorithms. *J. Qual. Technol.* **2021**, *53*, 505–525. [CrossRef]
23. Wang, W. Obtaining Robust Models from Imbalanced Data. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Tempe, AZ, USA, 21–25 February 2022; pp. 1555–1556.
24. Ren, M.; Zeng, W.; Yang, B.; Urtasun, R. Learning to reweight examples for robust deep learning. In Proceedings of the International Conference on Machine Learning (PMLR), Stockholm, Sweden, 10–15 July 2018; pp. 4334–4343.
25. Paganini, M. Prune responsibly. *arXiv* **2020**, arXiv:2009.09936.
26. Shwartz-Ziv, R.; Goldblum, M.; Li, Y.; Bruss, C.B.; Wilson, A.G. Simplifying Neural Network Training Under Class Imbalance. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 35218–35245.
27. Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; Yu, S.X. Large-scale long-tailed recognition in an open world. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2537–2546.
28. Lin, J.; Chen, W.M.; Lin, Y.; Gan, C.; Han, S. Mcunet: Tiny deep learning on iot devices. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 11711–11722.
29. Chen, J.; Ran, X. Deep learning with edge computing: A review. *Proc. IEEE* **2019**, *107*, 1655–1674. [CrossRef]
30. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
31. Wang, L.; Xu, S.; Wang, X.; Zhu, Q. Addressing class imbalance in federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 10165–10173.
32. Lee, D.; Kim, J. Resolving class imbalance for lidar-based object detector by dynamic weight average and contextual ground truth sampling. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 682–691.
33. Carranza-García, M.; Lara-Benítez, P.; García-Gutiérrez, J.; Riquelme, J.C. Enhancing object detection for autonomous driving by optimizing anchor generation and addressing class imbalance. *Neurocomputing* **2021**, *449*, 229–244. [CrossRef]
34. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11621–11631.
35. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 2446–2454.
36. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
37. Qiu, S.; Cheng, X.; Lu, H.; Zhang, H.; Wan, R.; Xue, X.; Pu, J. Subclassified loss: Rethinking data imbalance from subclass perspective for semantic segmentation. *IEEE Trans. Intell. Veh.* **2023**, *9*, 1547–1558. [CrossRef]
38. Hendrycks, D.; Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv* **2019**, arXiv:1903.12261.
39. Zheng, S.; Song, Y.; Leung, T.; Goodfellow, I. Improving the robustness of deep neural networks via stability training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4480–4488.
40. Zhang, Y.; Wang, Z.; Jiang, J.; You, H.; Chen, J. Toward Improving the Robustness of Deep Learning Models via Model Transformation. In Proce–14 October 2022.
41. Hendrycks, D.; Mazeika, M.; Kadavath, S.; Song, D. Using self-supervised learning can improve model robustness and uncertainty. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 15663–15674.
42. Merolla, P.; Appuswamy, R.; Arthur, J.; Esser, S.K.; Modha, D. Deep neural networks are robust to weight binarization and other non-linear distortions. *arXiv* **2016**, arXiv:1606.01981.
43. Li, Z.; Wallace, E.; Shen, S.; Lin, K.; Keutzer, K.; Klein, D.; Gonzalez, J. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In Proceedings of the International Conference on Machine Learning (PMLR), Online 13–18 July 2020; pp. 5958–5968.
44. Schwaiger, A.; Schwienbacher, K.; Roscher, K. Beyond Test Accuracy: The Effects of Model Compression on CNNs. In Proceedings of the SafeAI@ AAAI, Vancouver, BC, Canada, 22 February–1 March 2022.
45. Bengar, J.Z.; van de Weijer, J.; Fuentes, L.L.; Raducanu, B. Class-balanced active learning for image classification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 1536–1545.
46. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **2012**, *32*, 323–332. [CrossRef]
47. Zmora, N.; Jacob, G.; Zlotnik, L.; Elharar, B.; Novik, G. Neural network distiller: A python package for dnn compression research. *arXiv* **2019**, arXiv:1910.12232.

48. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 21–26 July 2016; pp. 770–778.
49. Noorizadegan, A.; Young, D.; Hon, Y.; Chen, C. Power-enhanced residual network for function approximation and physics-informed inverse problems. *Appl. Math. Comput.* **2024**, *480*, 128910. [CrossRef]
50. Liebenwein, L.; Baykal, C.; Carter, B.; Gifford, D.; Rus, D. Lost in pruning: The effects of pruning neural networks beyond test accuracy. *Proc. Mach. Learn. Syst.* **2021**, *3*, 93–138.
51. Kuzmin, A.; Nagel, M.; Van Baalen, M.; Behboodi, A.; Blankevoort, T. Pruning vs quantization: Which is better? *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 62414–62427.
52. Zhu, M.; Gupta, S. To prune, or not to prune: Exploring the efficacy of pruning for model compression. *arXiv* **2017**, arXiv:1710.01878.
53. Liang, T.; Glossner, J.; Wang, L.; Shi, S.; Zhang, X. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing* **2021**, *461*, 370–403. [CrossRef]
54. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning filters for efficient convnets. *arXiv* **2016**, arXiv:1608.08710.
55. So, J.; Lee, J.; Ahn, D.; Kim, H.; Park, E. Temporal dynamic quantization for diffusion models. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 48686–48698.
56. He, Y.; Liu, L.; Liu, J.; Wu, W.; Zhou, H.; Zhuang, B. PTQD: Accurate post-training quantization for diffusion models. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 13237–13249.
57. Tang, C.; Meng, Y.; Jiang, J.; Xie, S.; Lu, R.; Ma, X.; Wang, Z.; Zhu, W. Retraining-free model quantization via one-shot weight-coupling learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–24 June 2024; pp. 15855–15865.
58. Shang, Y.; Liu, G.; Kompella, R.R.; Yan, Y. Enhancing Post-training Quantization Calibration through Contrastive Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–24 June 2024; pp. 15921–15930.
59. Tang, C.; Ouyang, K.; Wang, Z.; Zhu, Y.; Ji, W.; Wang, Y.; Zhu, W. Mixed-precision neural network quantization via learned layer-wise importance. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 259–275.
60. Nagel, M.; Fournarakis, M.; Amjad, R.A.; Bondarenko, Y.; van Baalen, M.; Blankevoort, T. A white paper on neural network quantization. *arXiv* **2021**, arXiv:2106.08295.
61. Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv* **2018**, arXiv:1806.08342.
62. Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; Zou, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv* **2016**, arXiv:1606.06160.
63. Chmiel, B.; Banner, R.; Shomron, G.; Nahshan, Y.; Bronstein, A.; Weiser, U. Robust quantization: One model to rule them all. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5308–5317.
64. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both weights and connections for efficient neural network. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1135–1143.
65. Sun, S.; Ren, W.; Li, J.; Wang, R.; Cao, X. Logit standardization in knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–24 June 2024; pp. 15731–15740.
66. Mishra, A.; Marr, D. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv* **2017**, arXiv:1711.05852.
67. Hao, Z.; Guo, J.; Han, K.; Tang, Y.; Hu, H.; Wang, Y.; Xu, C. One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
68. Urban, G.; Geras, K.J.; Kahou, S.E.; Aslan, O.; Wang, S.; Caruana, R.; Mohamed, A.; Philipose, M.; Richardson, M. Do deep convolutional nets really need to be deep and convolutional? *arXiv* **2016**, arXiv:1603.05691.
69. Ashok, A.; Rhinehart, N.; Beainy, F.; Kitani, K.M. N2n learning: Network to network compression via policy gradient reinforcement learning. *arXiv* **2017**, arXiv:1709.06030.
70. Liang, C.; Zuo, S.; Chen, M.; Jiang, H.; Liu, X.; He, P.; Zhao, T.; Chen, W. Super tickets in pre-trained language models: From model compression to improving generalization. *arXiv* **2021**, arXiv:2105.12002.
71. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F.; Fernández, A.; García, S.; Galar, M.; Prati, R.C.; et al. Cost-sensitive learning. In *Learning from Imbalanced Data Sets*; Springer: Cham, Switzerland, 2018.
72. Bubeck, S.; Sellke, M. A universal law of robustness via isoperimetry. *J. ACM* **2023**, *70*, 1–18. [CrossRef]