applied system innovation

MDPI

Article

Matching the Ideal Pruning Method with Knowledge Distillation for Optimal Compression

Leila Malihi * and Gunther Heidemann

Department of Computer Vision, Institute of Cognitive Science, Osnabrück University, 49074 Osnabrück, Germany; gheidema@uos.de
* Correspondence: lemalihi@uos.de

Abstract: In recent years, model compression techniques have gained significant attention as a means to reduce the computational and memory requirements of deep neural networks. Knowledge distillation and pruning are two prominent approaches in this domain, each offering unique advantages in achieving model efficiency. This paper investigates the combined effects of knowledge distillation and two pruning strategies, weight pruning and channel pruning, on enhancing compression efficiency and model performance. The study introduces a metric called "Performance Efficiency" to evaluate the impact of these pruning strategies on model compression and performance. Our research is conducted on the popular datasets CIFAR-10 and CIFAR-100. We compared diverse model architectures, including ResNet, DenseNet, EfficientNet, and MobileNet. The results emphasize the efficacy of both weight and channel pruning in achieving model compression. However, a significant distinction emerges, with weight pruning showing superior performance across all four architecture types. We realized that the weight pruning method better adapts to knowledge distillation than channel pruning. Pruned models show a significant reduction in parameters without a significant reduction in accuracy.

Keywords: knowledge distillation; network efficiency; parameter reduction; unstructured pruning; structured pruning

Citation: Malihi, L.; Heidemann, G. Matching the Ideal Pruning Method with Knowledge Distillation for Optimal Compression. *Appl. Syst. Innov.* **2024**, *7*, 56. https://doi.org/10.3390/asi7040056

Academic Editors: Friedhelm Schwenker and Luís Oliveira

Received: 13 November 2023
Revised: 14 February 2024
Accepted: 20 June 2024
Published: 29 June 2024

1. Introduction

In deep learning, optimizing models for edge devices requires effective compression techniques like knowledge distillation and pruning. Knowledge distillation transfers insights from a large teacher model to a smaller student model, while pruning removes redundant network connections for efficiency. Despite their benefits, both methods face challenges in balancing compression and performance [1].

To overcome this limitation, our paper focuses on a crucial aspect of controlled model compression—comparing the effectiveness of weight pruning and channel pruning in combination with knowledge distillation. We proposed a unique sequence that starts with knowledge distillation, followed by pruning and fine-tuning [1]. This strategic combination of techniques enables us to achieve a more precise and controlled compression of the model while maintaining performance.

Our research builds upon our previous work [1], where we introduced a model with weight pruning and now extend it to explore channel pruning while retaining the same underlying architecture. Channel pruning represents a novel avenue for reducing model complexity, offering potential advantages over weight pruning in terms of computational efficiency and model performance. Through this exploration, we seek to uncover insights into the efficacy of channel pruning, particularly when combined with knowledge distillation, thereby providing a more holistic understanding of pruning techniques and their implications for future model compression efforts.

Moreover, investigating channel pruning serves a crucial role in advancing the field of deep learning model compression. Without such exploration, there remains a gap in

our understanding of the full spectrum of techniques available for reducing model size and computational complexity. Neglecting to explore channel pruning could result in missed opportunities for optimizing model performance, particularly in scenarios where computational resources are limited or where deployment on resource-constrained devices is necessary. By comprehensively examining both weight pruning and channel pruning in the context of knowledge distillation, we gain valuable insights into their comparative effectiveness and applicability across different architectures. This knowledge empowers researchers and practitioners to make informed decisions when selecting pruning techniques for specific use cases, ultimately facilitating the development of leaner, more efficient deep learning models that can be deployed in real-world applications. Our primary objective is fourfold:

- We begin with a comprehensive comparison of two renowned pruning methods, namely weight pruning and channel pruning, in combination with knowledge distillation (KD). Both methods utilize the L1 norm as the shared criteria, allowing for a fair and rigorous evaluation of their efficacy in compressing models while preserving performance.
- Furthermore, we delve into the concept of "Performance Efficiency" in our framework, a novel formula designed to quantify model efficiency by considering reductions in parameters alongside accuracy. This metric serves as a valuable tool for assessing the effectiveness of compression techniques in real-world deployment scenarios.
- Additionally, we employ rigorous statistical analysis, including t-tests, to evaluate the significance of differences between pruning methods in terms of their impact on model performance. By subjecting our findings to statistical scrutiny, we ensure the reliability and robustness of our conclusions, enhancing the credibility of our research outcomes.
- To demonstrate the effectiveness of our proposed pipeline, we conducted evaluations involving 10 model combinations on the CIFAR-10 and CIFAR-100 datasets. These experiments provide empirical evidence of the advantages and limitations of each approach, shedding light on their applicability in practical settings.

The rest of the paper is organized as follows: The "Related Work" section provides an overview of the state-of-the-art; the "Materials and Methods" section describes our experimental setup and approach; the "Experiments" section presents detailed empirical findings and their broader implications; and finally, the "Discussion" and "Conclusion" sections analyze and summarize our findings, offering additional insights into this task.

2. Related Works

In the field of model compression and optimization, researchers have been exploring various techniques to enhance the efficiency and effectiveness of deep neural networks. Two notable approaches that have gained significant attention are knowledge distillation and pruning strategies. These techniques play a crucial role in achieving models that are both efficient and high-performing.

2.1. Knowledge Distillation

Knowledge distillation (KD), as introduced by Hinton et al. [2], constitutes a pivotal concept in which the understanding of a larger, more complex model (teacher) is imparted to a smaller counterpart (student). The process involves utilizing the teacher's soft probabilities (logits) as "soft targets" during student training, steering its learning trajectory. This strategy showcases that despite the student's compactness and computational efficiency, it can achieve comparable or even superior performance to the teacher.

Attention transfer (AT), presented by Zagoruyko and Komodakis [3], introduces attention transfer within knowledge distillation. The integration of attention maps directs the focus toward significant regions in the input data. By orchestrating the student's learning to mimic the attention patterns exhibited by the teacher, the student's generalization and overall performance can be enhanced. Variational information distillation (VID), an innovation by Ahn et al. [4], injects the realm of variational distributions into knowledge

distillation. The student model assimilates knowledge from the teacher while accounting for the uncertainty inherent in the teacher's predictions. By considering the variance in the teacher's logits, the student captures both the central tendency and the confidence level associated with the teacher's assessments. "FitNets: Hints for Thin Deep Nets", proposed by Remero et al. [5], used hints or intermediate representations from a deeper teacher network to guide the training of a shallower student network. The FitNet architecture is designed to facilitate the distillation process, allowing the student network to learn not only from the final output of the teacher network but also from intermediate representations. This approach helps transfer knowledge more effectively, particularly in scenarios where the teacher network is significantly deeper than the student network.

Contrastive representation distillation (CRD), proposed by Tian et al. [6], brings contrastive learning into the fold of knowledge distillation. The alignment of teacher and student models employs contrastive loss, leveraging positive pairs from the same class and negative pairs from distinct classes. This fosters the student's ability to encapsulate distinguishing features while upholding intra-class coherence and inter-class divergence. Similarity-preserving knowledge distillation, as outlined by Tung et al. [7], introduces a paradigm where the student's learning is guided by preserving the pairwise similarities inherent in the teacher's features. This meticulous preservation of relative feature similarities empowers the student to capture the nuanced intricacies present in the teacher's representations. Self-residual representation learning (SRRL), brought forth by Zhang et al. [8], advances knowledge distillation with a focus on self-residual representations. The teacher's self-residuals serve as a conduit for amplifying the knowledge transfer process. The student's training entails learning to harness the teacher's self-residuals, leading to an elevation in its generalization capabilities. "Feature Normalized Knowledge Distillation for Image Classification" (FN), proposed by Xu et al. [9], presents an innovative approach to address label noise in the context of knowledge distillation (KD). The authors systematically analyze the impact of one-hot label encoding noise on the L2-norm of penultimate layer features, noting its potential distortion. They propose a novel feature normalized knowledge distillation method, introducing sample-specific correction factors to replace the conventional temperature parameter in KD.

Semantic conditioned knowledge distillation (SemCKD), introduced by Chen et al. [10], injects the concept of semantic conditioning into knowledge distillation. The student's learning journey is molded to emphasize specific semantic facets present in the teacher's predictions. This strategic emphasis empowers the student to internalize essential semantic cues, culminating in a heightened performance on intricate tasks.

Simple knowledge distillation (SimKD), an offering by Chen et al. [11], presents a streamlined approach by incorporating the teacher's discriminative classifier directly into the student's inference process. Further, the student's encoder is trained by harmonizing its features with those of the teacher, facilitated by a solitary $\ell 2$ loss. This elegant approach simplifies knowledge transfer while maintaining effectiveness.

Semantic conditioned knowledge distillation (SemCKD), introduced by Chen et al. [9], injects the concept of semantic conditioning into knowledge distillation. The student's learning journey is molded to emphasize specific semantic facets present in the teacher's predictions. This strategic emphasis empowers the student to internalize essential semantic cues, culminating in a heightened performance on intricate tasks.

Simple knowledge distillation (SimKD), an offering by Chen et al. [10], presents a streamlined approach by incorporating the teacher's discriminative classifier directly into the student's inference process. Further, the student's encoder is trained by harmonizing its features with those of the teacher, facilitated by a solitary $\ell 2$ loss. This elegant approach simplifies knowledge transfer while maintaining effectiveness.

2.2. Pruning

Pruning techniques have significantly transformed the field of deep neural networks by offering methods for compressing models, improving resource efficiency, and enabling

efficient inference without sacrificing performance. These techniques encompass both structured and unstructured pruning approaches, each with their own unique contributions. Structured pruning involves removing entire structures or components from the network, such as filters or channels, while unstructured pruning focuses on removing individual weights. Magnitude pruning, a popular form of unstructured pruning, identifies and prunes weights with low magnitudes, resulting in sparse networks with reduced parameters. L1 channel pruning, on the other hand, falls under the category of structured pruning. It removes entire channels from convolutional layers based on the L1 norm of the channel's weights. This approach has gained attention for its ability to reduce model size and computational requirements while maintaining performance.

Unstructured pruning involves the elimination of individual weights, yielding sparse networks with reduced parameters. A key technique within this domain is magnitude pruning, which focuses on removing low-magnitude weights. Magnitude pruning can be enhanced by incorporating L1 norm regularization, which emphasizes weight sparsity. Han and Liu [12] introduced deep compression, a pioneering framework that seamlessly integrates magnitude pruning with quantization. This approach efficiently compresses models by pruning unimportant weights. Furthermore, Han et al. [13] introduced the concept of "Learning Both Weights and Connections", extending unstructured pruning to include connection removal, demonstrating the synergy between structured and unstructured pruning. Structured pruning techniques maintain the network's architecture while reducing complexity. L1 channel pruning, a standout method, involves removing entire channels based on the L1 norm of weights, ensuring the preservation of critical information. Li et al. [14] proposed structured pruning with proximal operators, a technique combining magnitude pruning and structured sparsity. This method optimizes model performance retention during pruning. Additionally, Lin et al. [15] introduced adaptive channel widening, a novel approach that blends magnitude pruning with dynamic channel operations to adaptively adjust channel widths. Recent advancements have showcased the efficacy of combining structured and unstructured pruning techniques for enhanced model compression. Molchanov et al. [16] introduced an efficient inference engine for compressed neural networks, incorporating L1-norm-based pruning and quantization. Ding et al. [17] introduced approximated oracle filter pruning, which combines magnitude pruning with filter-level optimization for width reduction. The concept of "Deep Compression" by Han et al. [10] further demonstrates the efficacy of integrating structured and unstructured pruning, quantization, and Huffman coding in a comprehensive approach.

Pruning alone does not provide precise control over the amount of compression, making it challenging to find the right balance between compression and performance. However, combining pruning with knowledge distillation (KD) addresses this limitation and offers a more controlled compression strategy. By incorporating KD alongside pruning, our approach overcomes the challenges of finding an optimal compression level. KD allows us to leverage the distilled knowledge of a teacher model to guide the pruning process. This eliminates the need for extensive adjustments and fine-tunes only the last connected layer. As a result, we achieve a more compressed network that outperforms conventional pruning techniques.

The advantage of our method lies in its ability to intelligently combine knowledge distillation and pruning, resulting in greater compression. By leveraging the teacher model's knowledge, we achieve a more accurate and balanced compression strategy. This demonstrates the success of our approach in effectively balancing compression and performance.

Previous works, such as Aghly and Ribeiro [18], Chen et al. [10], Xie et al. [19], Cui and Li [20], Kim et al. [21], and Wang et al. [22], have explored the combination of pruning and knowledge distillation for model compression. These studies have shown that knowledge distillation can be used after pruning to improve accuracy and restore performance in pruned models.

Our current research represents a natural progression from our previous paper, where we introduced a pioneering method that prioritized knowledge distillation before prun-

ing [1]. This approach highlighted the importance of using distilled knowledge to increase model compression while maintaining or even improving performance. In our previous work, we demonstrated the efficacy of this approach and its advantages over conventional methods. By starting with knowledge distillation, we provided the student model with advanced insights from the teacher model, resulting in more efficient pruning and improved compression rates [1].

Building upon these foundational findings, the present study extends the investigation by focusing on a comparative analysis of weight and channel pruning techniques within the framework of knowledge distillation. By systematically evaluating their performance across various metrics and datasets, we aim to elucidate their respective strengths and limitations in the context of model compression. This comparative analysis allows us to deepen our understanding of optimal compression strategies and inform future developments in the field of deep learning model optimization for edge device deployment.

3. Materials and Methods

In this section, we introduce our novel approach, as detailed in [1], which is designed to achieve efficient model compression while minimizing its impact on accuracy. For more information and a comprehensive understanding of our methodology, please refer to the original paper [1].

3.1. Simple Knowledge Distillation (SimKD)

Simple knowledge distillation (SimKD) introduces an intriguing "classifier-reusing" operation where the pre-trained teacher classifier is directly employed for student inference. By discarding the necessity for label information, this approach relies solely on feature alignment loss, fostering the generation of gradients and the sharing of feature extractors across tasks. SimKD capitalizes on the teacher classifier's inferential capabilities and enforces feature alignment, effecting robust knowledge transfer [10,11].

3.2. Weight Pruning

L1 unstructured pruning (weight pruning) is a technique aimed at inducing sparsity within neural networks by selectively removing individual weights based on their magnitudes, as measured by the L1 norm. The L1 norm of a weight vector is the sum of the absolute values of its elements. In this approach, a designated percentage of weights with the smallest L1 norm values are pruned, effectively setting them to zero. This results in a model with sparse connectivity, where numerous weights become non-trainable and are excluded during model inference [17].

Post-pruning, the resulting sparse model with zero-valued weights can be fine-tuned or directly deployed for inference. L1 unstructured pruning offers notable advantages, including diminished model size, improved computational efficiency, and potential opportunities for hardware acceleration due to the sparsity it introduces [1].

3.3. Channel Pruning

Channel L1 pruning stands as a transformative technique within the realm of neural network compression, enabling the refinement of models through structured sparsity while preserving architectural integrity. This method, rooted in the concept of the L1 norm, empowers practitioners to create leaner and more resource-efficient networks. At its core, channel L1 pruning operates by selectively pruning entire channels or feature maps within the convolutional layers of a neural network. This approach relies on the L1 norm of the weights associated with each channel, allowing for the identification of less influential channels that can be safely removed. Channel L1 pruning hinges on the notion that certain channels contribute minimally to the network's overall performance. By quantifying the importance of each channel through its L1 norm, the technique sorts and prunes channels with lower L1 norm values [23]. This meticulous approach maintains a delicate balance between structural preservation and efficiency gain. Mathematically, the L1 norm of the

weights within a channel is determined by summing the absolute values of its constituent weights for a channel, as follows:

$$L1_{norm}(C) = |W_1| + |W_2| + \ldots + |W_n| \quad (1)$$

During channel L1 pruning, channels exhibiting the smallest L1 norm values are pruned. Channel L1 pruning is a process that selectively removes entire feature maps from a neural network, reducing its complexity while retaining important information. This pruning technique offers several benefits that reshape the design of neural networks:

- Architectural Integrity: By focusing on entire channels, channel L1 pruning preserves the underlying architecture of the network. This ensures that critical information flow patterns, which contribute to the model's effectiveness, are maintained.
- Resource Efficiency: Removing less influential channels results in a leaner model, leading to improved resource efficiency. This reduces the computational resources and memory required during both training and inference, making the model more efficient.
- Regularization and Generalization: Channel L1 pruning encourages the network to rely on essential features while diminishing reliance on redundant or less informative channels. This regularization process helps improve the generalization capabilities of the model and reduces overfitting, resulting in better performance on unseen data.

3.4. Efficiency Metric

In the pursuit of advancing neural network efficiency, a new and practical formula called "Efficiency" has emerged as a key tool for evaluating model performance. This formula strikes a careful balance between two crucial elements: predictive accuracy and model simplicity. It revolves around the interaction of two key components: accuracy and the number of parameters after pruning.

"Accuracy" represents how well the model can provide accurate predictions for the given task. It measures the model's effectiveness in achieving the desired outcomes.

The "Number of Parameters after Pruning" quantifies the complexity of the model after applying pruning techniques. Pruning involves selectively removing unnecessary parameters from the model, resulting in a streamlined architecture.

Mathematically, the efficiency formula can be expressed as a ratio or a function that combines these two elements. It provides a quantitative measure of how effectively the model balances accuracy and complexity. Mathematically, it can be expressed as follows:

$$\text{Performance Efficiency} = (\omega_1 \times \text{Accuracy})/(\omega_2 \times \text{Number of Parameters after pruning}) \quad (2)$$

Within this equation, the coefficients "ω_1" and "ω_2" perform a vital role in fine-tuning the emphasis on accuracy and complexity, respectively. The performance efficiency formula is a flexible tool that allows researchers and practitioners to customize it according to their specific optimization goals. It takes into account the impact of regularization techniques during pruning, ensuring that less important parameters are removed while preserving or even improving the model's predictive abilities. The formula also considers the role of hyper-parameters in the accuracy component, gracefully integrating their influence. By comprehensively assessing the interplay between regularization, hyper-parameters, accuracy, and model complexity, the formula provides a clear way to gauge and enhance the efficiency of pruned models. The efficiency metric captures how well a model performs relative to its complexity. A higher efficiency value indicates that the model achieves a higher level of accuracy per parameter, indicating a more effective use of model capacity. This metric is useful for comparing different models or techniques, providing a quantitative measure of how well each model balances accuracy and complexity. However, it is important to note that the efficiency formula is just one aspect of the broader narrative. It does not encompass factors like interpretability, feature relevance, or domain-specific considerations, which can further enrich the holistic assessment of model efficiency.

In conclusion, the performance efficiency formula represents the ongoing pursuit of optimal neural network efficiency. Its ability to balance accuracy and complexity while considering regularization and hyperparameters makes it a comprehensive tool for evaluating model performance. Although challenges remain, the efficiency formula remains an invaluable asset in the toolkit of machine learning practitioners.

4. Experiments

For our experimental framework, we used the potential of two benchmark image classification datasets: CIFAR-100 [24] and CIFAR-10 [24]. We used common data augmentation techniques and normalized all images using channel means and standard deviations. This detailed preprocessing is consistent with established practices in the field [25,26]. Employing the SGD optimizer with a Nesterov momentum of 0.9 across all datasets, we set the learning rate at 0.001. A mini-batch size of 64 was embraced, coupled with a weight decay parameter of 5×10^{-4}. Remarkably, the temperature parameter T in the KD loss remained consistent at 4.

4.1. Pruning

In our paper [1], we provide a comprehensive performance comparison of various distillation approaches using 11 network combinations where teacher and student models have either similar or completely different architectures. We compare nine different knowledge distillation methods to determine which one yields the best performance. Our results show that SimKD consistently outperforms all competitors on CIFAR-100 and CIFAR-10. For more detailed analysis and results, please refer to the paper [1].

With our distilled student models identified through the SimKD [11], we performed a series of network pruning experiments on distilled student models. The pruning process consisted of gradually decreasing the number of parameters from 0% to 90% in steps of 10%. To ensure accuracy and reliability, we followed a rigorous approach. We conducted 10 iterations for each teacher-student pair and averaged the results to increase the robustness of our analysis. This repeated averaging helped us reduce potential biases or instabilities in our findings. It considered factors such as the capacity of the initial model, the effect of random initialization, and the need for a comprehensive evaluation of the knowledge transfer process. By averaging the results over multiple iterations, we obtained a more representative understanding of the flexibility and adaptability of the model. These features are very important in evaluating the effectiveness of the knowledge transfer mechanism.

After conducting accurate pruning iterations, we focused on accurately calculating and documenting the remaining parameters in both the pruned teacher and student networks. This involved counting the non-zero weights or connections that were retained in the pruned networks. This quantification helped us understand the level of parameter reduction achieved through pruning, providing valuable insights into the streamlined architecture of the model.

However, a "breakpoint" arises where accuracy drops significantly faster relative to parameter reduction. Remarkably, in our results, many student models achieve comparable or better accuracy than distilled models at this breakpoint. This emphasizes the effectiveness of combining knowledge distillation and pruning for improved efficiency without accuracy loss. At the breakpoint, students leverage the teacher's insights, maintaining performance despite fewer parameters [1].

As shown in Figures 1 and 2, we present a comprehensive pruning analysis for weight and channel pruning on the CIFAR-10 and CIFAR-100 datasets. Through these figures, we explore the relationship between accuracy and the remaining parameters after pruning. Across all model architectures, we consistently observe that weight pruning achieves comparable or higher accuracy at a significantly lower number of remaining parameters, as demonstrated by the corresponding figures for ResNet 101 to 18 and EfficientNet B1 to B0. This trend suggests that weight pruning is adept at identifying and retaining essential parameters while effectively reducing the model's complexity. Consequently, weight

pruning emerges as a powerful technique for achieving enhanced efficiency with fewer parameters, making it particularly well-suited for scenarios with strict computational or memory constraints.

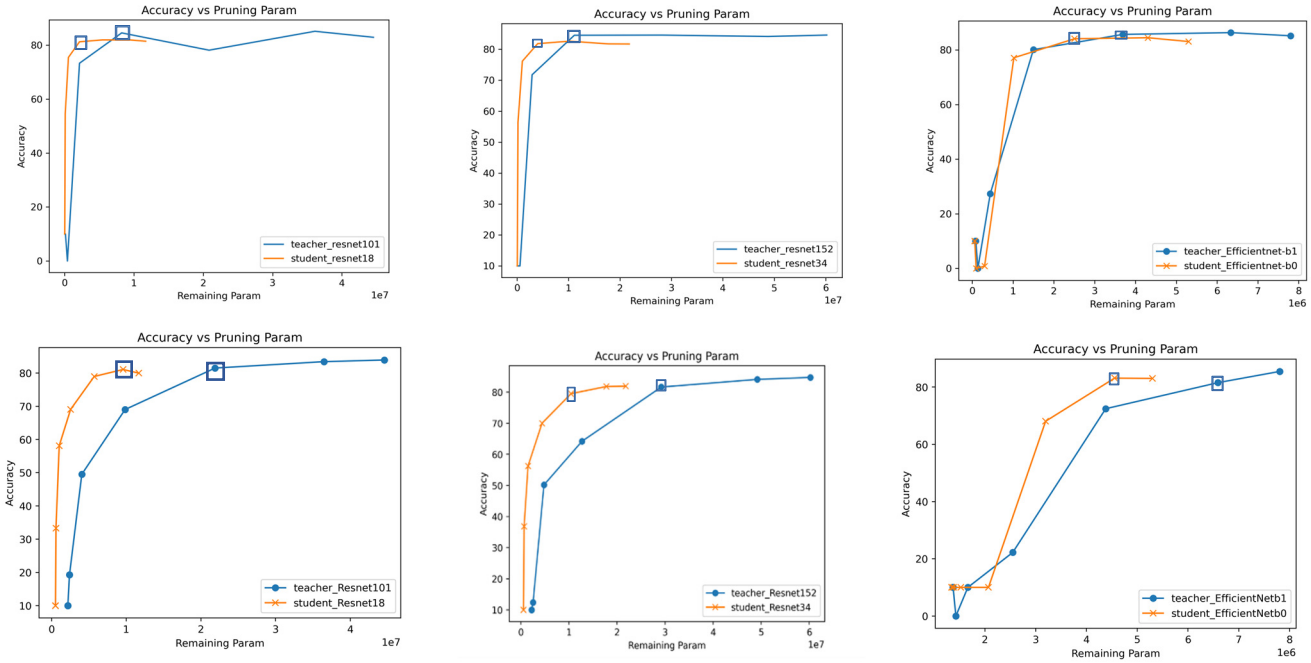

Figure 1. Pruning analysis. Comparative parameter count in student and teacher models on CIFAR-10: The first row is weight pruning, and the second row is channel pruning.

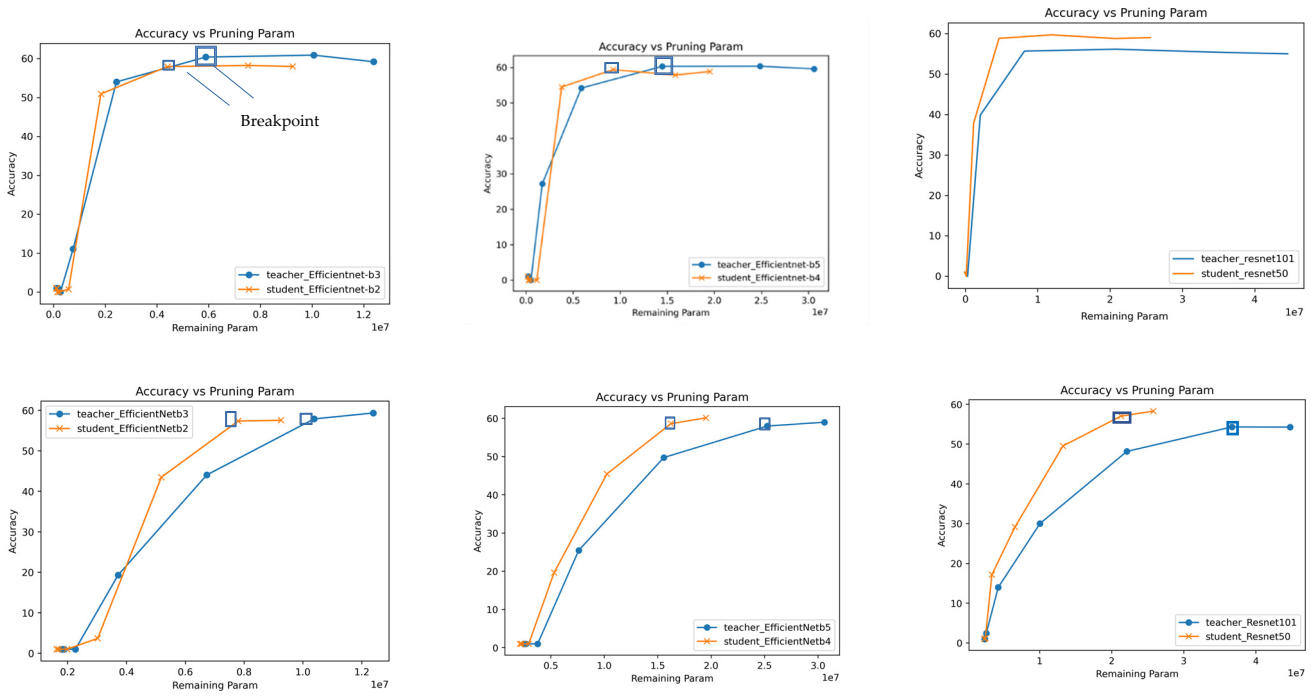

Figure 2. Pruning analysis. Comparative parameter count in student and teacher models on CIFAR-100: The first row is weight pruning, and the second row is channel pruning.

In contrast, channel pruning exhibits a different pattern, as highlighted in Figures 1 and 2. Although the breakpoint is generally reached with comparable accuracy, the corresponding number of parameters is notably higher in channel pruning. This implies that while channel pruning maintains similar accuracy levels, it retains a larger proportion of parameters compared to weight pruning. Consequently, channel pruning may be considered a more conservative approach in terms of parameter reduction, potentially sacrificing some efficiency gains to maintain greater model capacity. A comparison between weight pruning and channel pruning highlights the inherent trade-offs between model complexity, accuracy, and efficiency. Weight pruning shows its power in significantly reducing the number of parameters while maintaining or even increasing accuracy.

A remarkable trend emerges where weight pruning consistently achieves levels of accuracy that are comparable across different combinations of teacher and student models, or even surpass the accuracy of the teacher models at the breaking point. For instance, consider the transition from ResNet 101 to ResNet 50. The student model attains an impressive 58.92% accuracy after weight pruning, outperforming both teacher and channel pruning accuracies, which stand at 54.34% and 56.78%, respectively. The findings of our analysis of channel pruning yield a notable observation. Across the range of student models analyzed, a consistent trend is observed: with the exception of the ResNet-50 architecture, pruned student models are less accurate than their corresponding teacher models. This model emphasizes the complexity of channel pruning and the importance of a delicate approach to balance reducing model complexity while maintaining predictive accuracy. Further exploration of the factors contributing to this divergence can provide valuable insights to refine channel pruning strategies and optimize the performance of different architectures.

The distinct characteristics exhibited by weight pruning and channel pruning highlight the importance of careful and thoughtful evaluation when choosing between these two pruning methods. In the case of the ResNet 101 to ResNet 50 transition, after weight pruning, the student model's parameter count is only 0.5×10^7, while after channel pruning, it amounts to 2.1×10^7.

Weight pruning shines as a method capable of optimizing both accuracy and efficiency. Its potential to surpass teacher accuracy while achieving substantial parameter reduction positions it as an attractive choice for applications where resource efficiency is paramount. On the other hand, channel pruning excels at producing lightweight models by drastically reducing parameter counts. While accuracy is sometimes compromised, this method is well-suited for environments with strict computational constraints. The choice between weight pruning and channel pruning should be guided by a thorough understanding of the desired balance between model accuracy, parameter reduction, and available resources. To illustrate the comparative effectiveness of these pruning techniques, we provide detailed results in Tables 1 and 2, comparing accuracy and parameter count at different pruning breakpoints for CIFAR-10 and CIFAR-100 across various models.

Table 1. Accuracy and parameter count at breakpoint for CIFAR-100.

Teacher, Student	Res101, Res50	Res152, Res 34	Res101, Res 18	Dens169, Mobile	Dens161, Dens 169	Dens161, Dens 201	Effici b1, Effici b0	Effici b3, Effici b2	Effici b3, Mobile [3]	Effici b5, Effici b4
Weight Pruning [1]										
ACC(t)	54.07	52.2	54.07	57.2	58.2	58.2	59.2	58.1 ± 0.5	58.1	59.5
ACC(s) distillation	59.12	51.3	55.6	52.1	56.45	54.78	57.1	55.87	52.07	59.4
P [1](t)$_{\text{before pruning}} \times 10^7$	4.45	6.02	4.45	1.42	2.68	2.68	0.78	1.22	1.22	3.04
P [2](s)$_{\text{before pruning}} \times 10^7$	2.56	2.13	1.17	0.42	1.42	2	0.56	0.91	0.42	1.93
ACC(t)$_{\text{breakpoint}}$	54.34	47.48	53.15	57.95	57.71	58.94	60.17	60	60	60.06
ACC(s)$_{\text{breakpoint}}$	58.92	53.47	55.25	55.1	57.61	55.23	58.63	55.91	54.8	59.8
P(t)$_{\text{breakpoint}} \times 10^7$	0.91	1.1	0.91	0.35	0.5	0.52	0.39	0.6	0.55	1.42
P(s)$_{\text{breakpoint}} \times 10^7$	0.5	0.37	0.32	0.18	0.29	0.41	0.27	0.42	0.15	0.8

Table 1. *Cont.*

Teacher, Student	Res101, Res50	Res152, Res 34	Res101, Res 18	Dens169, Mobile	Dens161, Dens 169	Dens161, Dens 201	Effici b1, Effici b0	Effici b3, Effici b2	Effici b3, Mobile [3]	Effici b5, Effici b4
Channel Pruning										
ACC(t)	54.07	52.2	54.7	57.2,	58.2	58.2	59.2	58.1 ± 0.5	58.1	59.5
ACC(s)$_{\text{distillation}}$	59.12	51.3	55.6	52.1	56.45	54.78	57.1	55.87	52.07	59.4
P [1](t)$_{\text{before pruning}}$ $\times 10^7$	4.45	6.02	4.45	1.42	2.68	2.68	0.78	1.22	1.22	3.04
P [2](s)$_{\text{before pruning}}$ $\times 10^7$	2.56	2.13	1.17	0.42	1.42	2	0.56	0.91	0.42	1.93
ACC(t)$_{\text{breakpoint}}$	54.07	52.2	54.1	57.2	57.87	57.87	54.1	56.2 ± 0.5	58.1	56.2
ACC(s)$_{\text{breakpoint}}$	56.78	50.7	53.78	52.89	56.89	53.67	55	55.87	51.98	56.98
P(t)$_{\text{breakpoint}}$ $\times 10^7$	3.5	2.7	3.5	1.19	1.5	1.5	0.65	1	1	2.5
P(s)$_{\text{breakpoint}}$ $\times 10^7$	2.1	1.8	0.8	0.35	1.1	1.7	0.45	0.75	0.27	1.6

[1]. P: number of parameters P(t): parameter count in the teacher; P(s): parameter count in the student. [2]. P(s)$_{\text{before}}$/P(s)$_{\text{break}}$: number of parameters before pruning/number of parameters after pruning at the break point for the student. [3]. Mobile: MobileNet.

Table 2. Accuracy and parameter count at breakpoint for CIFAR-10.

Teacher, Student	Res101, Res50	Res152, Res 34	Res101, Res 18	Dens169, Mobile	Dens161, Dens 169	Dens161, Dens 201	Effici b1, Effici b0	Effici b3, Effici b2	Effici b3, Mobile	Effici b5, Effici b4
Weight Pruning [1]										
ACC(t)	81.1	83.3	81.7	83.15	83.07	83.07	81.1	80.5	80.5	83.4
ACC(s)$_{\text{distillation}}$	81.7	80.1	80.56 ± 0.75	78.2	82.76	82.33	80.2	81.14	80.2	83.3
P(t)$_{\text{before pruning}}$ $\times 10^7$	4.45	6.02	4.45	1.42	2.68	2.68	0.78	1.22	1.22	3.04
P(s)$_{\text{before pruning}}$ $\times 10^7$	2.56	2.13	1.17	0.42	1.42	2	0.56	0.91	0.42	1.93
ACC(t)$_{\text{breakpoint}}$	82.7	83.15	83.32	83.53	82.86	83.53	81.62	80.52	80.12	83.1
ACC(s)$_{\text{breakpoint}}$	83.51	80.71	80.78	82.93	83.02	83.02	80.74	81.15	80.13	83.78
P(t)$_{\text{breakpoint}}$ $\times 10^7$	0.91	1.1	0.91	0.25	0.38	0.58	0.37	0.47	0.22	1.4
P(s)$_{\text{breakpoint}}$ $\times 10^7$	0.6	0.51	0.34	0.16	0.2	0.41	0.27	0.27	0.18	0.8
Channel Pruning										
ACC(t)	81.1	83.3	81.7	83.15	83.07	83.07	81.1	80.5	80.5	83.4
ACC(s)$_{\text{distillation}}$	81.7	80.1	80.65 ± 0.75	78.2	82.76	82.33	80.2	81.14	80.2	83.3
P(t)$_{\text{before pruning}}$ $\times 10^7$	4.45	6.02	4.45	1.42	2.68	2.68	0.78	1.22	1.22	3.04
P(s)$_{\text{before pruning}}$ $\times 10^7$	2.56	2.13	1.17	0.42	1.42	2	0.56	0.91	0.42	1.93
ACC(t)$_{\text{breakpoint}}$	80.2	80.1	80.2	79.12	82.45	82.45	80.1	80.65	80.5	83.4
ACC(s)$_{\text{breakpoint}}$	81.7	80	79.56	80	81.3	82.33	80.3	79.1	80.2	83.3
P(t)$_{\text{breakpoint}}$ $\times 10^7$	2.1	3	2.1	0.73	1.5	1.5	0.67	1	1	2.5
P(s)$_{\text{breakpoint}}$ $\times 10^7$	2	1	0.6	0.33	0.75	1.58	0.43	0.75	0.27	1.6

Weight pruning often results in a notably reduced parameter count compared to channel pruning, with differences sometimes reaching as high as a factor of four.

4.2. Impact of Pruning Methods on Efficiency

In this section, we explore what we have learned by comparing different ways to make models more efficient. We tested various pruning methods on different student models using the CIFAR-10 and CIFAR-100 datasets. The objective of our study was to assess the impact of pruning techniques on the efficiency of neural network models, enabling us to understand the trade-offs between model complexity and performance.

In Figures 3 and 4, we depict the outcomes of our efficiency comparison analysis. These results are based on the application of the efficiency formula, where we have chosen the balanced weights of w1 = w2 = 0.5. The comparison encompasses diverse student models and employs distinct pruning methodologies. The CIFAR-10 and CIFAR-100 datasets serve as the testing grounds for our investigation. These bar charts offer a visual representation of the impact of pruning techniques on the efficiency of the student models, shedding light on the trade-offs between model complexity and performance. They illustrate the efficiency comparison of student models across two pruning methods on the CIFAR-10 and CIFAR-100 datasets. Each bar in the chart represents the normalized efficiency score of a student model under different pruning strategies. Notably, the distinct patterns that

emerge from the chart provide valuable insights into the effectiveness of each pruning method. Our investigation sheds light on how each method affects the efficiency of the models and offers valuable insights into their relative advantages and limitations. From the generated charts for both the CIFAR-10 and CIFAR-100 datasets, it is evident that the impact of pruning methods on efficiency varies across different student models. Weight pruning consistently results in improved efficiency compared to the other methods, with reductions in the number of parameters leading to better performance. This aligns with the expectation that weight pruning targets redundant parameters, leading to more efficient representations while maintaining accuracy. In the realm of channel pruning, it is evident that the efficiency of the pruned models surpasses that of the distilled student models, albeit by a relatively modest degree. However, a more pronounced distinction emerges when comparing channel pruning to weight pruning. In this context, the efficiency achieved through weight pruning consistently outperforms that of channel pruning. This discernible contrast emphasizes the divergent impact of these pruning techniques on model efficiency and raises intriguing considerations when selecting an optimal approach to enhance neural network performance.

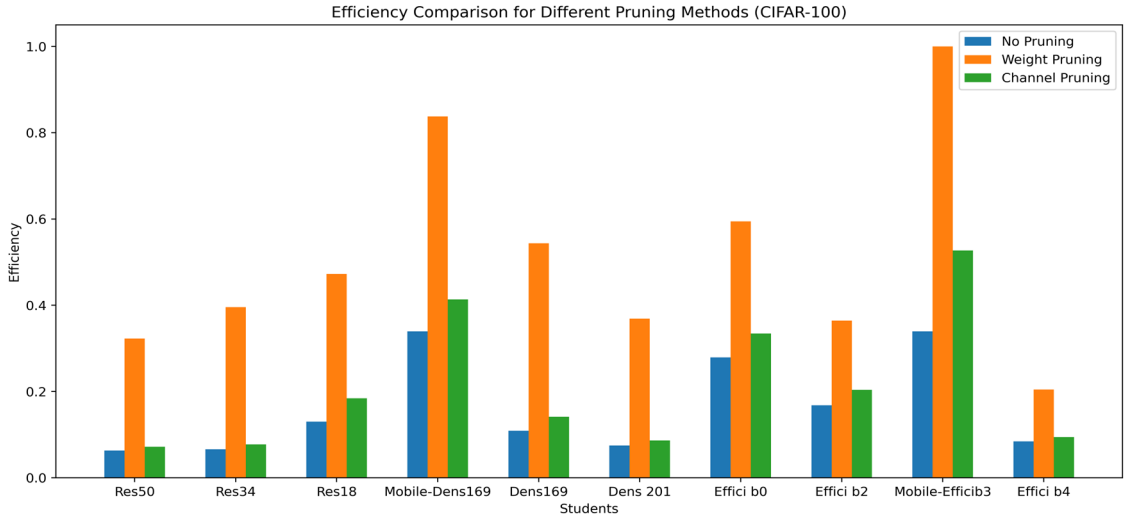

Figure 3. Efficiency comparison for different pruning methods on CIFAR-100.

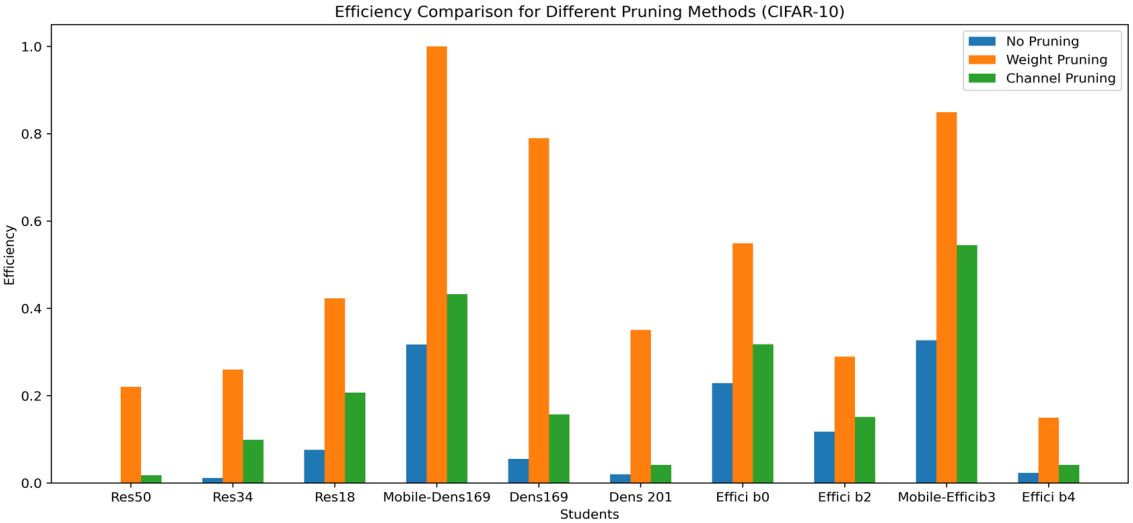

Figure 4. Efficiency comparison for different pruning methods on CIFAR-10.

4.3. How to Select the Student?

In our quest to make neural networks more efficient using knowledge distillation, we carefully chose pairs of teacher and student models by drawing from well-established ideas in the influential literature. Taking inspiration from the work of Hinton et al. [2], we follow the idea of distilling knowledge from complex teacher models (like ResNet101) to simpler student architectures (like ResNet50). Considering the attention mechanisms emphasized by Zagoruyko and Komodakis [3], our choice of teacher-student pairs involves attention transfer. An example is our selection of DenseNet169 as the teacher and MobileNet as the student, where attention transfer guides the distillation process. Following insights from Romero et al. [5], on model compression, we aim to balance model complexity and efficiency. For instance, we pair DenseNet161 as the teacher with DenseNet169 as the student, distilling knowledge from a more complex model into a slightly smaller but still intricate counterpart. Furthermore, our approach integrates insights derived from successful ensemble learning, as outlined by Furlanello et al. [27]. This is demonstrated in pairing EfficientNet b5 as the teacher with EfficientNet b4 as the student, showcasing the potential benefits of combining diverse student architectures.

In summary, we chose teacher and student models carefully, considering principles like knowledge distillation, making models smaller, specific task needs, attention transfer, and the benefits of combining different models.

4.4. Comparing the Results

Table 3 illustrates a comprehensive comparison of state-of-the-art compression methods applied to a diverse set of teacher models, including ResNet-101, ResNet-152, DenseNet-169, DenseNet-161, EfficientNet-b1, EfficientNet-b3, and EfficientNet-b5. Our method, labeled "channel pruning" and "weight pruning" [1] exhibits compelling results, outperforming seven benchmark techniques encompassing two pruning methods ([2,10]), two knowledge distillation approaches ([11,13]), and two hybrid compression methods ([17,21]). We performed evaluations on our own proprietary models and datasets due to the lack of previous results for these specific configurations. This approach ensures that performance evaluation is tailored to our distinct context and emphasizes the compatibility and effectiveness of each compression method. The superiority of our method across a range of teacher models underlines its efficiency in achieving significant compression results.

Table 3. Comparison number of parameters on CIFAR-100.

Teacher	Res101	Res152	Dens169	Dens161	Effici b1	Effici b3	Effici b5
(Channel pruning) $\times$ 10^7	0.8	1.8	0.35	1.1	0.45	0.27	**1.6**
(Weight pruning) [1] $\times$ 10^7	**0.32**	**0.37**	**0.18**	**0.29**	0.27	**0.15**	**0.8**
[2] $\times$ 10^7	1.87	2.42	0.41	1.95	0.52	0.43	**1.92**
[10] $\times$ 10^7	1.12	2.1	0.36	1.76	0.51	0.38	**1.81**
[11] $\times$ 10^7	2.23	1.2	0.68	1.52	0.58	1	**2.4**
[13] $\times$ 10^7	3.5	2.8	0.71	1.43	0.65	1.1	**2.51**
[17] $\times$ 10^7	0.81	1.15	0.37	1.09	**0.25**	0.3	**1.54**
[21] $\times$ 10^7	**0.74**	**0.83**	**0.32**	**0.85**	**0.43**	**0.21**	**1.67**

Quantitative evaluation, measured in terms of parameter reduction ($\times 10^7$), consistently demonstrated the superiority of our method over most teacher models. Notably, our approach achieved the best results in the ResNet and DenseNet architectures and demonstrated its effectiveness in compressing these models. In the case of EfficientNet-b1, our method closely competed with the performance of the [17] approach, indicating comparable efficiency. These results highlight the robustness and efficiency of our proposed compression method and emphasize its potential for model reduction while maintaining

or even improving performance, especially in our specific models and datasets where no previous results were available. The pivotal factor in this success was the strategic use of knowledge distillation (KD) before pruning. This approach not only provided more flexibility in compression options but also demonstrated that, when it comes to weight versus channel pruning, weight pruning in conjunction with KD yielded superior compression.

In summary, our experiments showcase a novel and highly effective compression strategy—distilling models to lower architectures, followed by weight pruning. This approach not only outperforms prior methods but also highlights how using knowledge distillation (KD) before pruning is a powerful way to achieve the best compression results.

5. Discussion

The paper delves into the realm of model compression techniques, specifically exploring the synergies between knowledge distillation and two pruning strategies—weight pruning and channel pruning. These techniques aim to reduce the computational and memory requirements of deep neural networks, enhancing overall model efficiency. The study introduces a novel metric termed "Performance Efficiency" to evaluate how these pruning strategies impact both model compression and performance. In Figures 3 and 4, we present the Efficiency comparison for different pruning methods on CIFAR-100 and CIFAR-10.

Importantly, the paper reveals that weight pruning is particularly well-suited to knowledge distillation, outperforming channel pruning in this context.

The discussion highlights the pivotal role of knowledge distillation (KD) in the combined approach. While both weight and channel pruning contribute significantly to model compression, the paper accentuates that weight pruning, when combined with KD, showcases enhanced efficiency. The adoption of weight pruning proves more advantageous in the knowledge distillation framework, ensuring a substantial reduction in parameters without a commensurate decrease in accuracy. Weight pruning targets finer granularity within the network's architecture by removing individual parameters in neural network layers. This allows for more comprehensive compression without significantly affecting the network's representational capacity.

Weight pruning also aligns well with the process of knowledge distillation, effectively leveraging the distilled knowledge from the teacher model to emphasize the most relevant and informative connections in the student model. On the other hand, channel pruning eliminates entire channels of feature maps within convolutional layers while leaving fully connected layers untouched. However, channel pruning can still contribute to efficiency gains in certain scenarios. The paper emphasizes that the choice between weight pruning and channel pruning depends on the specific problem or data characteristics. Weight pruning is recommended when a more precise and comprehensive compression is desired and when knowledge distillation is an important aspect of the model deployment. Channel pruning, on the other hand, may be suitable when a coarser level of compression is acceptable and when preserving all features within the network is not crucial. It is important to note that weight pruning and channel pruning have distinct effects on different layers and components of the network.

In Figure 5, we present a comprehensive sparsity comparison between weight pruning and channel pruning techniques applied to three distinct architectures: EfficientNet-b0, EfficientNet-b2, and MobileNet. These architectures were distilled with knowledge from EfficientNet-b1, EfficientNet-b3, and EfficientNet-b3, respectively. Our analysis shows significant differences in sparsity levels between weight pruning and channel pruning, specifically in the final layer of the selected architectures, which corresponds to the last fully connected layer. An important distinction between weight pruning and channel pruning is in the treatment of fully connected layers. Weight pruning applies parameter reduction across all layers, including fully connected layers, ensuring more comprehensive compression. Conversely, channel pruning eliminates entire channels of feature maps within convolutional layers while leaving the fully connected layers untouched.

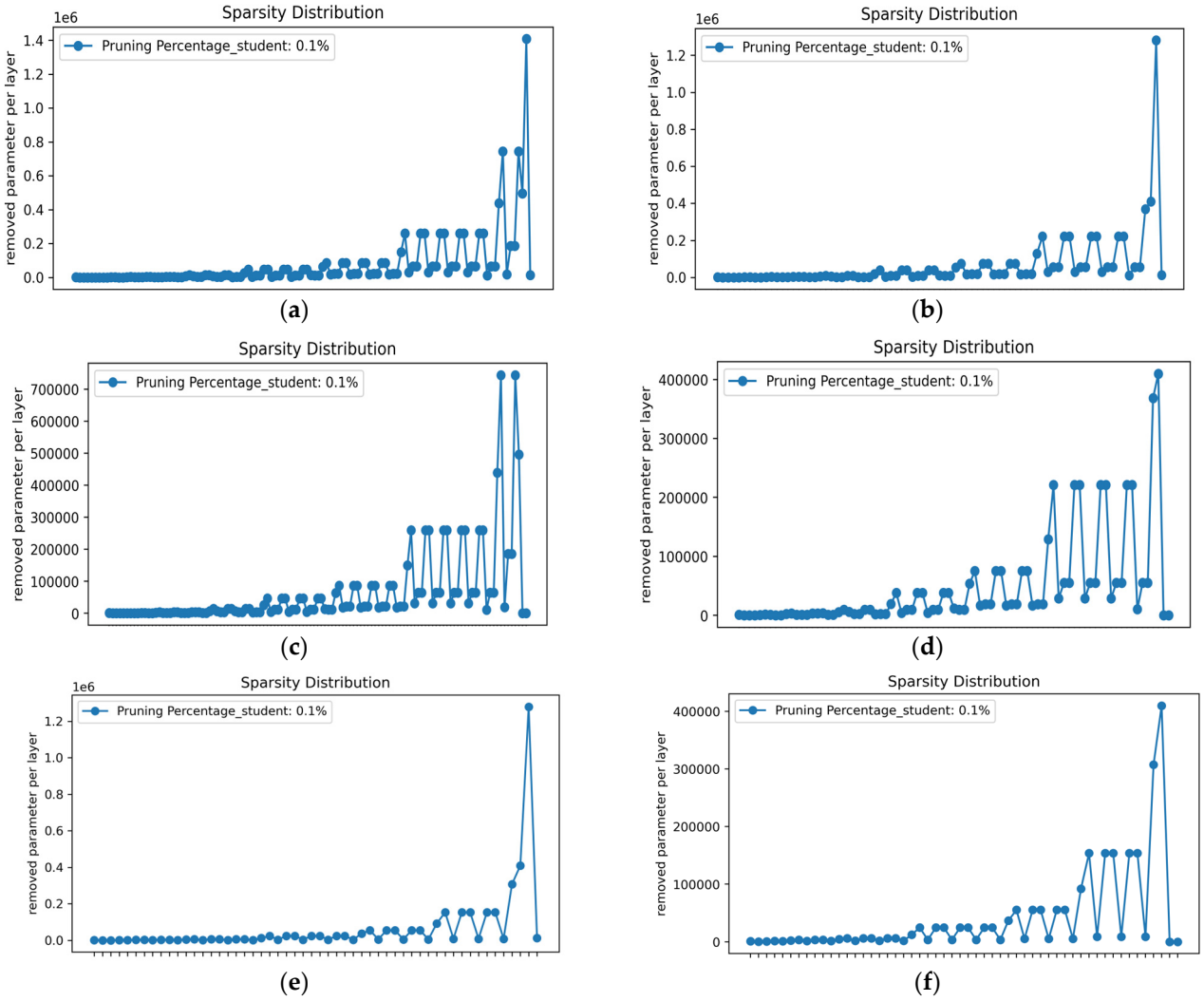

Figure 5. Sparsity distribution. Removed parameter in each layer: (**a**) Weight pruning on EfficientNet-b2; (**b**) Weight pruning on EfficientNet-b0; (**c**) Weight pruning on MobileNet; (**d**) Channel pruning on EfficientNet-b2; (**e**) Channel pruning on EfficientNet-b0; (**f**) Channel pruning on MobileNet.

Therefore, researchers should carefully consider the trade-offs between accuracy, compression ratio, and the specific requirements of their problem domain when choosing the appropriate pruning method. For a more in-depth analysis, we conducted a t-test to assess whether weight pruning is a more suitable choice for combining with knowledge distillation (KD). This statistical test helps us understand if weight pruning consistently outperforms channel pruning when integrated with KD. The results provide valuable insights into the effectiveness of combining specific pruning methods with KD for optimal model efficiency. Figure 6 illustrates the outcomes of our comprehensive t-test analysis. The t-test results, a fundamental tool in statistical hypothesis testing, offer valuable insights into the relative performance of these pruning methods. We delve into the implications of the depicted figures and what they reveal about the efficiency comparison.

The *t*-test results reveal significant differences in efficiency between weight pruning and channel pruning for both the CIFAR-10 and CIFAR-100 datasets. For CIFAR-10, weight pruning outperforms channel pruning with a t-statistic of -5.60 and a p-value of 3.3×10^{-4}, signifying high statistical significance. Similarly, for CIFAR-100, weight pruning demonstrates superior efficiency with a t-statistic of -7.06 and an even lower p-value of 5.90×10^{-5}, indicating an extremely high level of statistical significance. These results consistently show that weight pruning is a more efficient choice, suggesting that when combined with knowledge distillation, it is likely to produce a model with a better

balance between accuracy and reduced complexity. These findings provide valuable guidance for researchers seeking to optimize model performance and efficiency in different problem domains.

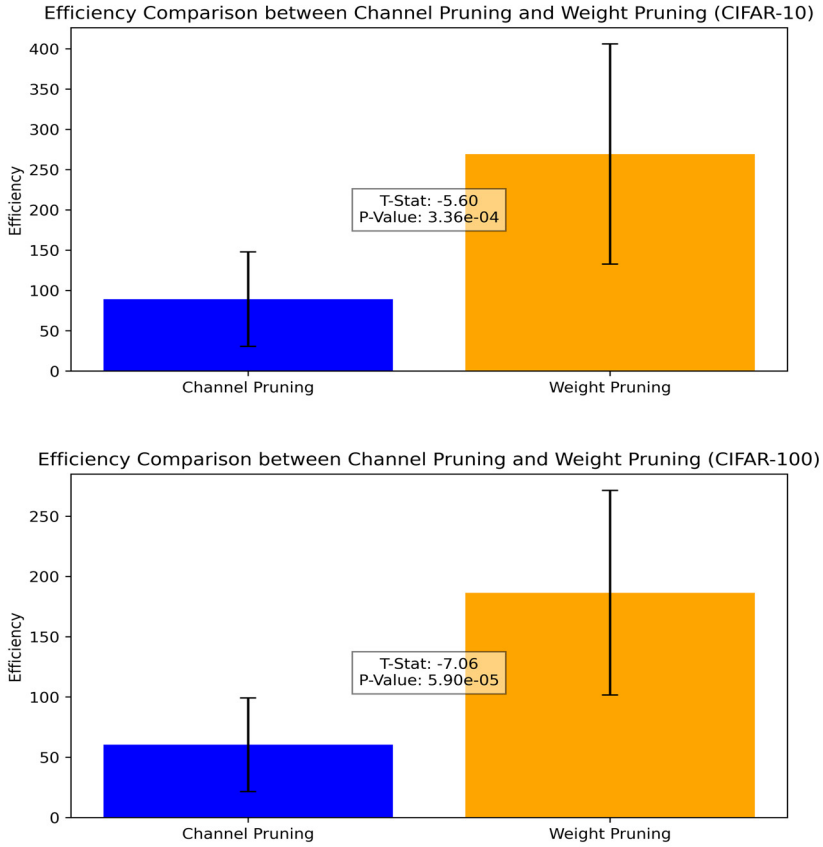

Figure 6. Efficiency comparison between channel and weight pruning using the *t*-test.

6. Conclusions

In this study, we examined the impact of weight pruning and channel pruning methods after KD. To evaluate the efficiency of these methods, we introduced a formula called "efficiency" that takes into account the number of parameters and the performance of the pruned models. Through our experiments, we observed that weight pruning consistently outperformed channel pruning when combined with knowledge distillation. This finding was consistent across various model architectures, including ResNet, DenseNet, and EfficientNet, over a total of 10 cases. Weight pruning demonstrated superior efficiency by significantly reducing the number of parameters while maintaining high model performance. Channel pruning, on the other hand, showed comparatively lower efficiency in terms of achieving both compression and performance enhancement. These results highlight the effectiveness of weight pruning as a pruning method when incorporating knowledge distillation. The combination of knowledge distillation and weight pruning offers a powerful approach for achieving efficient model compression without sacrificing performance.

Author Contributions: Conceptualization, L.M.; methodology, L.M.; software, L.M.; validation, L.M.; formal analysis, L.M.; investigation, L.M.; resources, L.M. and G.H.; data curation, L.M.; writing—original draft preparation, L.M.; writing—review and editing, L.M. and G.H.; visualization, L.M.; supervision, G.H.; funding acquisition, L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was funded by the Osnabrück University "Open Access Publizieren" of the "Deutsche Forschungsgemeinschaft" (DFG) with fund number DFG-4321.

Data Availability Statement: For researchers seeking access to the data utilized in this study, collaboration opportunities, and data access approval, we kindly request direct contact with the corresponding author. Additionally, any potential collaboration involving data access is subject to the approval of the institutional review board. This collaborative approach ensures transparency and adherence to ethical considerations. It is important to note that the authors declare the absence of conflicts of interest associated with the study. The funders did not contribute to the design, data collection, analysis, interpretation, manuscript composition, or publication decision of this research. The study's support was solely provided by the respective institutions of the authors. This declaration underscores the commitment to maintaining research integrity, independence, and impartiality throughout the entirety of the research process.

Conflicts of Interest: The authors state that there are no conflict of interest related to the study. The funders played no role in the study's design, data collection, analysis, interpretation, manuscript writing, or decision to publish. The study was solely supported by the authors' institutions. This declaration ensures transparency and independence in the research process.

References

1. Malihi, L.; Heidemann, G. Efficient and Controllable Model Compression through Sequential Knowledge Distillation and Pruning. *Big Data Cogn. Comput.* **2023**, *7*, 154. [CrossRef]
2. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
3. Zagoruyko, S.; Komodakis, N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. *arXiv* **2017**, arXiv:1612.03928.
4. Ahn, S.; Hu, S.X.; Damianou, A.; Lawrence, N.D.; Dai, Z. Variational Information Distillation for Knowledge Transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
5. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for Thin Deep Nets. *arXiv* **2015**, arXiv:1412.6550.
6. Tian, Y.; Krishnan, D.; Isola, P. Contrastive Representation Distillation. *arXiv* **2022**, arXiv:1910.10699.
7. Tung, F.; Mori, G. Similarity-Preserving Knowledge Distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
8. Pham, T.X.; Niu, A.; Kang, Z.; Madjid, S.R.; Hong, J.W.; Kim, D.; Tee, J.T.J.; Yoo, C.D. Self-Supervised Visual Representation Learning via Residual Momentum. *arXiv* **2022**, arXiv:2211.09861. [CrossRef]
9. Xu, K.; Lai, R.; Li, Y.; Gu, L. Feature Normalized Knowledge Distillation for Image Classification. In *Computer Vision—ECCV 2020 ECCV 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; pp. 664–680. ISBN 978-3-030-58594-5.
10. Chen, D.; Mei, J.-P.; Zhang, Y.; Wang, C.; Wang, Z.; Feng, Y.; Chen, C. Cross-Layer Distillation with Semantic Calibration. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 7028–7036. [CrossRef]
11. Chen, D.; Mei, J.-P.; Zhang, H.; Wang, C.; Feng, Y.; Chen, C. Knowledge Distillation with the Reused Teacher Classifier. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022.
12. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv* **2016**, arXiv:1510.00149.
13. Han, S.; Pool, J.; Tran, J.; Dally, W.J. Learning Both Weights and Connections for Efficient Neural Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
14. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. *arXiv* **2017**, arXiv:1608.08710.
15. Lin, S.; Ji, R.; Yan, C.; Zhang, B.; Cao, L.; Ye, Q.; Huang, F.; Doermann, D. Towards Optimal Structured CNN Pruning via Generative Adversarial Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
16. Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; Kautz, J. Pruning Convolutional Neural Networks for Resource Efficient Inference. *arXiv* **2017**, arXiv:1611.06440.
17. Ding, X.; Ding, G.; Guo, Y.; Han, J.; Yan, C. Approximated Oracle Filter Pruning for Destructive CNN Width Optimization. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
18. Aghli, N.; Ribeiro, E. Combining Weight Pruning and Knowledge Distillation for CNN Compression. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; pp. 3185–3192.
19. Xie, H.; Jiang, W.; Luo, H.; Yu, H. Model Compression via Pruning and Knowledge Distillation for Person Re-Identification. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 2149–2161. [CrossRef]
20. Cui, B.; Li, Y.; Zhang, Z. Joint Structured Pruning and Dense Knowledge Distillation for Efficient Transformer Model Compression. *Neurocomputing* **2021**, *458*, 56–69. [CrossRef]
21. Kim, J.; Chang, S.; Kwak, N. PQK: Model Compression via Pruning, Quantization, and Knowledge Distillation. *arXiv* **2021**, arXiv:2106.14681.

22. Wang, R.; Wan, S.; Zhang, W.; Zhang, C.; Li, Y.; Xu, S.; Zhang, L.; Jin, X.; Jiang, Z.; Rao, Y. Progressive Multi-Level Distillation Learning for Pruning Network. *Complex Intell. Syst.* **2023**, *9*, 5779–5791. [CrossRef]
23. Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Learning Structured Sparsity in Deep Neural Networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
24. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
25. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2018**, arXiv:1608.06993.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
27. Furlanello, T.; Lipton, Z.C.; Tschannen, M.; Itti, L.; Anandkumar, A. Born Again Neural Networks. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.