mathematics

MDPI

Article

Probabilistic Automated Model Compression via Representation Mutual Information Optimization

Wenjie Nie, Shengchuan Zhang and Xiawu Zheng *

Department of Artificial lntelligence, Xiamen University, Xiamen 361101, China; 31520231154307@stu.xmu.edu.cn (W.N.); zsc_2016@xmu.edu.cn (S.Z.)
* Correspondence: zhengxiawu@xmu.edu.cn; Tel.: +86-153-9627-3732

Abstract: Deep neural networks, despite their remarkable success in computer vision tasks, often face deployment challenges due to high computational demands and memory usage. Addressing this, we introduce a probabilistic framework for automated model compression (Prob-AMC) that optimizes pruning, quantization, and knowledge distillation simultaneously using information theory. Our approach is grounded in maximizing the mutual information between the original and compressed network representations, ensuring the preservation of essential features under resource constraints. Specifically, we employ layer-wise self-representation mutual information analysis, sampling-based pruning and quantization allocation, and progressive knowledge distillation using the optimal compressed model as a teacher assistant. Through extensive experiments on CIFAR-10 and ImageNet, we demonstrate that Prob-AMC achieves a superior compression ratio of 33.41× on ResNet-18 with only a 1.01% performance degradation, outperforming state-of-the-art methods in terms of both compression efficiency and accuracy. This optimization process is highly practical, requiring merely a few GPU hours, and bridges the gap between theoretical information measures and practical model compression, offering significant insights for efficient deep learning deployment.

Keywords: probabilistic model compression; representation mutual information; neural network compression; automated compression pipeline; information theory

MSC: 68T05; 68T10; 62H30

check for updates

Academic Editors: Tao Dai and Jinbao Wang

Received: 18 November 2024
Revised: 17 December 2024
Accepted: 25 December 2024
Published: 30 December 2024

Citation: Nie, W.; Zhang, S.; Zheng, X. Probabilistic Automated Model Compression via Representation Mutual Information Optimization. *Mathematics* **2025**, *13*, 108. https://doi.org/10.3390/math13010108

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

While convolutional neural networks (CNNs) [1] have revolutionized computer vision tasks, including classification [2–4], detection [5,6], and semantic segmentation [7,8], their deployment on resource-constrained devices remains challenging due to the massive computational and memory requirements. This challenge has motivated the development of various model compression techniques, each representing different probabilistic transformations of the original network distribution. Traditional compression approaches primarily fall into three categories: parameter quantization [9–15], network pruning [16–24], and knowledge distillation (KD) [25–29].

While previous work [30] demonstrates the effectiveness of combining these strategies sequentially, finding optimal compression configurations remains a challenging probabilistic optimization problem. Recent studies [11,31–33] reveal that the performance of compressed models is highly sensitive to hyperparameter selection, leading to the emergence of AutoML-based approaches [34–36] for automated compression. Deep learning has been used in all walks of life [29,37–51].

However, the existing automated frameworks typically optimize only one or two compression aspects in isolation, overlooking the probabilistic dependencies between different compression strategies. For instance, AMC [21] and NetAdapt [16] employ reinforcement learning to search only layer-wise channel numbers, while other methods [34–36] focus on isolated aspects of compression. This fragmented approach has two major limitations: First, the independent optimization of compression strategies ignores their inherent probabilistic interactions, potentially leading to suboptimal solutions. Second, the exponential growth in hyperparameters when considering multiple compression methods simultaneously makes traditional search approaches computationally prohibitive.

To address these challenges, we present Fully-AMC (Fully Automated Model Compression), a probabilistic framework that unifies network pruning, parameter quantization, and knowledge distillation. Our key insight is that optimal compression solutions maintain high mutual information with the original network's probability distribution, regardless of the specific compression strategy. This information-theoretic perspective enables us to formulate compression as a probabilistic optimization problem that preserves essential features of the original network while satisfying resource constraints.

As illustrated in Figure 1, our framework consists of three probabilistic components: (1) layer-wise self-representation mutual information analysis to determine compression ratios, (2) sampling-based pruning and quantization allocation guided by representation mutual information, and (3) progressive knowledge distillation using the optimal compressed model as a teacher assistant. This probabilistic approach not only unifies different compression strategies but also significantly reduces the computational resources required for finding optimal solutions.

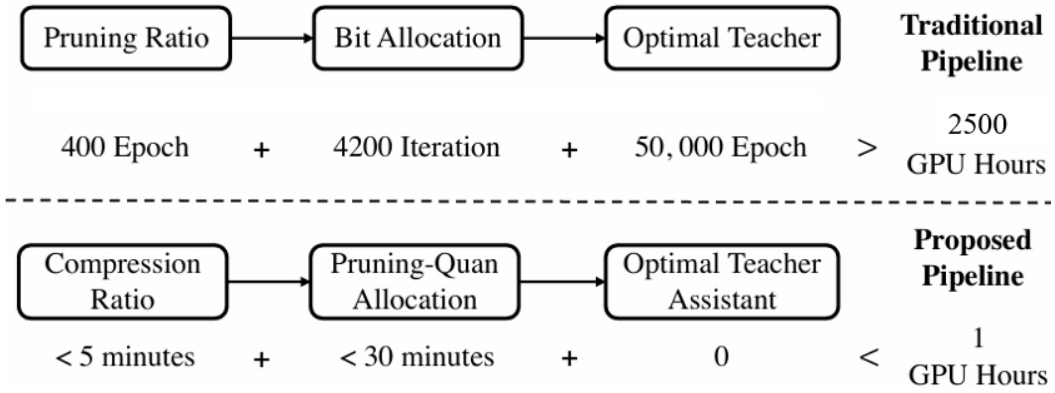


Figure 1. Comparison of the traditional and the proposed compression pipeline used to obtain the optimal compression configuration.

To summarize, our main contributions are as follows:

- We propose the first unified probabilistic framework for automated model compression that jointly optimizes network pruning, parameter quantization, and knowledge distillation.
- We establish representation mutual information as a reliable probabilistic metric for evaluating compression quality, supported by extensive statistical analysis (extensive experiments in Section 4 show that representation mutual information shows a high correlation with the performance of the compressed model using pruning and quantization).
- We develop a novel compression pipeline that efficiently navigates the vast search space using information-theoretic principles, significantly reducing computational requirements while maintaining compression quality.

Based on the above challenges and motivations, this paper aims to address the following research questions:

- **RQ1:** How can we effectively unify and jointly optimize different compression strategies (pruning, quantization, and knowledge distillation) in a single automated framework?
- **RQ2:** Can representation mutual information serve as a reliable metric for evaluating compression quality across different compression strategies?
- **RQ3:** How can we efficiently navigate the vast search space of compression configurations while maintaining compression quality?

Our experimental results demonstrate the effectiveness of this probabilistic approach across different architectures and datasets. Notably, our method achieves a 33.41× compression ratio on ResNet-18 with only 1.01 performance degradation, without requiring extensive search-time training.

2. Related Work

As we mentioned before, researchers have proposed various compression methods, which are classified as parameter quantization [9–15], network pruning [16–24], and knowledge distillation [25–29]. Therefore, in this section, we briefly review these areas as well as the methods that combine these compression strategies sequentially.

Pruning. Network pruning aims to remove "unimportant" connections from a pretrained network and fine-tune the compressed network to restore accuracy. Therefore, researchers have focused on how to define the importance of network connections early in the process. For instance, Li et al. [24] define importance according to the $L1$ norm. Meanwhile, He et al. [2] consider that the geometric median is an accurate indicator for importance. In these works, filters or weights that are classified as important are reserved, and vice versa. These methods require extra human effort to design the metric and determine hyperparameters like the layer-wise compression ratio in practice. In addition, previous works [31,32] have also demonstrated that filter-level importance does not help as much as selecting a better channel number for a specific architecture. To this end, AutoML [21] has been leveraged to automate the aforementioned process. However, using AutoML still requires additional computational costs. Meanwhile, searching only in the pruning hyperparameter is suboptimal in the scope of model compression with an integrated space that includes pruning, quantization, and distillation.

Quantization. Parameter quantization is another orthogonal approach for model compression and acceleration. Quantization allows the neural network to be represented by low-precision numbers, such as int numbers, instead of full-precision representation. The existing quantization methods can be roughly divided into two categories: quantization-aware training (QAT) and post-training quantization (PTQ). Quantization-aware training [11–13] reduces the performance gap between the original model and the quantized model by retraining. However, QAT is computationally expensive because of time-consuming fine-tuning. To deal with this problem, post-training quantization [15,52,53] directly quantizes neural network models without fine-tuning. However, traditional quantization approaches use the same low bit for parameter quantization, which may cause significant accuracy degradation. One way to address this is to use mixed-precision quantization. Recently, some works have used network architecture search [54,55] or reinforcement learning [56,57] methods to search different precisions on different layers. Both of these methods improve the accuracy of quantized models while requiring a lot of computational resources.

Knowledge Distillation. Knowledge distillation (KD) is a powerful tool to significantly improve the performance of neural networks. By imitating the teacher's output, the student network can acquire distilled knowledge to enhance its performance [27]. Specifically, Romero et al. [58] utilize the output of hidden activation as a target in KD.

Meanwhile, Heo et al. [59] propose employing an activation boundary of neurons in hidden layers for KD, and Zagoruyko et al. [60] introduce distilling knowledge from attention information. To sum up, different methods adapt different knowledge to supervise the student. However, it is also recognized that there is a significant capacity gap between the teacher and student. Cho et al. [61] reveal that the distillation of knowledge does not work when the capacity of the student network is too small to imitate the teacher model. In other words, it is necessary to develop a special method to find the optimal teacher assistant (TA). To this end, some works [62–64] focus on constructing progressive knowledge distillation routes. However, the structures of intermediate models require expert experience to design manually, which limits the application of the progressive distillation method.

Joint Optimization. Based on the fundamental character of pruning and quantization, these two neural network model compression methods can be jointly optimized. Early work [65] proposes a three-stage compression method that included pruning, post-training quantization, and Hoffman coding to significantly reduce the neural network's computational cost and memory footprint. Some methods [66,67] introduce an in-parallel pruning-quantization pipeline to compress models into a smaller size without too much of a drop in accuracy. Yang et al. [67] propose an automated Alternating Direction Method of Multipliers (ADMM) model compression framework to prune and quantize neural networks jointly while satisfying the target of the model size. Hu et al. [35] proposes a joint compression method that uses the statistical information of the pre-trained model's weights for pruning and quantization. As we mentioned before, the corresponding optimized components are considered separately, which is suboptimal and may possibly bias the output solution. Meanwhile, the search space is also largely expanded and thus requires a large amount of computational resources. Therefore, we propose to reorganize the search space and consider the aforementioned compression methods simultaneously.

3. Method

Notations. In this paper, we use uppercase (e.g., X, Y) for random variables and bold for vectors (e.g., $\boldsymbol{x}, \boldsymbol{y}$), matrices, or tensors (e.g., $\boldsymbol{X}, \boldsymbol{Y}$). Calligraphic font denotes spaces (e.g., $\mathcal{X}, \mathcal{Y}$). To better describe the proposed method, we further define some common notations in CNNs. The l-th convolutional layer in a specific CNN converts an input tensor $\boldsymbol{X}^l \in \mathbb{R}^{c^l \times h^l_{\text{in}} \times w^l_{\text{in}}}$ to an output tensor $\boldsymbol{X}^{l+1} \in \mathbb{R}^{n^l \times h^l_{\text{out}} \times w^l_{\text{out}}}$ by using a weight tensor $\boldsymbol{W}^l \in \mathbb{R}^{n^l \times c^l \times k^l \times k^l}$, where c^l and n^l denote the numbers of the input channels and filters (output channels), respectively, and $k^l \times k^l$ is the spatial size of the filters. Formally, we use $f(\cdot)$ to represent the convolution in the $l-$th layer $f^l(\boldsymbol{W}^l, \boldsymbol{X}^l) = \boldsymbol{W}^l \circledast \boldsymbol{X}^l$.

Problem Formulation. Given a pre-trained CNN model that contains L layers, we refer to $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_L)$ as the desired compression configuration, where $\alpha_l = [\alpha^p_l, \alpha^q_l]$, and $\alpha^p_l \in \mathbb{R}^{c^l \times 1}$ and $\alpha^q_l \in [1..32]$ are the pruning mask and quantization bit of the l-th layer, respectively. Formally, we address the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{W}, \boldsymbol{W}^{\text{TA}}, \boldsymbol{\alpha}, \boldsymbol{\alpha}^{\text{TA}}} & \mathcal{L}_{\text{KD}}\left(f\left(\boldsymbol{W}^{\text{TA}}, \boldsymbol{\alpha}^{\text{TA}}\right), f(\boldsymbol{W}, \boldsymbol{\alpha})\right) + \\ & \mathcal{L}_{\text{CE}}\left(f^{\text{softmax}}(\boldsymbol{W}, \boldsymbol{\alpha}), y\right), \\ & s.t.\ T(\boldsymbol{\alpha}) < \Omega, \end{aligned} \tag{1}$$

where $\mathcal{L}_{\text{KD}}$ and $\mathcal{L}_{\text{CE}}$ are the objective functions for knowledge distillation and cross-entropy, respectively. These loss functions are differentiable *with respect to* the network weight $\boldsymbol{W}, \boldsymbol{W}^{\text{TA}} \in \mathcal{W}$ and not differentiable *with respect to* the compression configuration $\boldsymbol{\alpha}, \boldsymbol{\alpha}^{\text{TA}} \in \mathcal{A}$. Superscript $^{\text{TA}}$ denotes the teacher assistant mentioned in [64]. $T(.)$ is a

function to obtain the specific evaluation metric for compression configurations, such as BitOps [57] and the compression rate, and Ω is the given constraint for different hardware.

We present an overview of Fully-AMC in Figure 2, which aims to automatically discover the optimal compression configuration for a pre-trained network. The detailed motivations, descriptions, and analysis are presented in the following sub-sections.

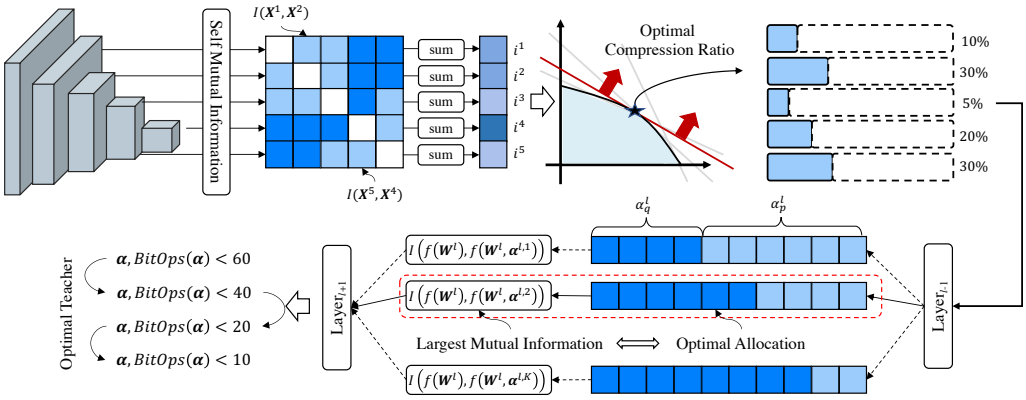


Figure 2. Overview of the proposed Fully-AMC. In particular, we first sample n samples to obtain the feature map of each layer, which is then used to obtain the layer-wise mutual information map $\boldsymbol{H}$. Then, we sum the elements of the corresponding row in $\boldsymbol{H}$ except for itself to obtain the layer-wise importance vector $\boldsymbol{i}$. In this way, we model the layer-wise importance and compression constraints as a linear programming problem, where the optimal layer-wise compression ratio $\boldsymbol{c}$ is obtained by solving the problem. Pruning–quantization allocations around the corresponding compression ratio c^l are sampled to construct the search space for layer l. Meanwhile, the allocation $\boldsymbol{\alpha}$ with the largest mutual information $I\left(f\left(\boldsymbol{W}^l\right), f\left(\boldsymbol{W}^l, \boldsymbol{\alpha}\right)\right)$ is selected as the optimal solution. Finally, models compressed with different constraints are considered as the optimal teacher assistant, where progressive KD is adopted to obtain the final result.

3.1. Search Space

Motivation. The search space defines which component of the corresponding compression strategy needs to be represented in principle. A good search space needs to solve the problem of laborious human efforts in the traditional compression pipeline. Meanwhile, it also needs to influence the final performance as much as possible. The above rules are adopted as the principle of the search space design.

Search Space Design. The search space defined in our paper is outlined as follows: (i). *Pruning ratio/layer-wise channel number.* A conventional pruning pipeline can be divided into two sequential components: pruning ratio determination and filter pruning determination. Traditional works [24,68] propose pruning filters with different metrics, where the pruning ratio is neglected and set to be uniform. However, as we mentioned in Section 2, extensive experiments in [31,32] have shown that pruning ratio determination is much more important than filter-level pruning. We thus only consider the pruning ratio as our search space and the $L1$ norm is adopted for filter pruning. (ii). *Mixed precision/layer-wise bit allocation.* Similar to the pruning, uniformly quantizing is empirically proved to lead to significant accuracy degradation. In contrast, mixed precision shows much more promising performance and has attracted more attention recently. Therefore, only mixed precision is considered in the quantization of our search space. (iii). *Teacher assistant search.* Compared with pruning and quantization, KD is much more complicated. Specifically, different KD algorithms use different knowledge types, knowledge sources, loss functions, and teachers [69]. Among these components, the architecture or configuration of the teacher is empirically proved to be the key factor to influence the final performance [33,70,71]. In addition, the selection of the architecture is also orthogonal to other components in KD and can be combined with these components to further improve the performance. Considering

that we only search the pruning ratio and mixed precision in the previous section, we thus further search the optimal teacher assistant in the aforementioned search space.

3.2. Fully Automated Model Compression with Representation Mutual Information

Search Pipeline. As illustrated in Figure 1, a conventional compression pipeline is sequentially divided into pruning, quantization, and KD, which is also adopted in the previous joint optimization works [34–36]. As we mentioned before, such a search paradigm is suboptimal and possibly biases the output solution, as there is no guarantee that the optimal pruning solution is independent of the optimal mixed-precision solution. Therefore, we summarize the search spaces in different compression strategies to conduct an integrated search space and pipeline, which is illustrated in Figure 2. Firstly, both the pruning ratio and mixed precision implicitly indicate the layer-wise compression ratio. We thus conclude that it is a key factor to determine the final performance, which needs to be searched first. Secondly, after the layer-wise compression ratio is determined, allocating the pruning and quantization ratios is still needed to simulate the traditional search space in [34–36]. We thus propose searching for the pruning and quantization allocation in the second step. Finally, after obtaining the optimal compression configuration, we still need to search for the best teacher assistant. For simplicity, we limit the search space to the compression search space mentioned in this paper. And then, we search for multiple teacher assistants in the search space to perform progressive distillation [64] to improve the final performance. To sum up, we propose to fully automate the compression model with compression ratio determination, pruning–quantization allocation, and optimal teacher assistant selection, which will be elaborated in the following contents.

Effective Performance Estimation. Even if we propose an efficient search pipeline, it still requires a large amount of computational resources. The key to search efficiency lies in how to evaluate the compression configuration. Here, we propose a new hypothesis to solve this issue, which is also empirically demonstrated in Section 4.

Hypothesis 1. *If compression configuration A has higher representation mutual information with the original network than compression configuration B on a randomly sampled training batch, then the final performance of A tends to be better than B on the validation set.*

The hypothesis illustrates a simple yet important rule in automated model compression. The comparison of different compression configurations can be finished with a batch of data. Meanwhile, training to converge on the entire training data is unnecessary and time-consuming. In fact, extensive experiments in Section 4 have demonstrated that such an indicator is robust across different compression strategies. More specifically, representation mutual information shows a high correlation with the final performance within a few optimization steps, no matter what compression strategy is adopted. In our paper, we directly optimize the mutual information on one training batch, where the result is employed as an indicator to explore the search space.

Compression Ratio. We now describe how to obtain the optimal compression ratio by using self-representation mutual information. In particular, we first sample n images to obtain the feature map of each layer $\boldsymbol{X}^1, \boldsymbol{X}^2, ..., \boldsymbol{X}^L$ in the original network. We then use these feature maps to obtain the independence between different layers, thereby constructing an independence matrix $\boldsymbol{H} \in \mathbb{R}^{L \times L}$, where $\boldsymbol{H}$ has entries $\boldsymbol{H}_{i,j} = \mathrm{I}(X^i, X^j)$. As mentioned in [72], a layer correlated to other layers is less important. Therefore, the importance of a specific layer l is formally defined as

$$\boldsymbol{i}^l = e^{-\beta \sum_{i=1, i \neq l}^{L} \mathrm{I}(X^l, X^i)}, \tag{2}$$

where β is proposed to control the relative independence between different layers. To explain, the compressed network tends to have an independence compression rate difference between different layers with a larger β, and vice versa. With the layer-wise importance $\boldsymbol{i} \in \mathbb{R}^{L\times 1}$, searching for the optimal compression ratio is transformed into a linear programming problem. Formally, the layer-wise compression ratio $\boldsymbol{\alpha}^+ \in \mathbb{R}^{L\times 1}$ is obtained through

$$\max_{\boldsymbol{\alpha}^+} \boldsymbol{i}^T \boldsymbol{\alpha}^+ \, s.t. \; T(\boldsymbol{\alpha}^+) < \Omega. \tag{3}$$

In our paper, we widely use BitOPs as the compression constraint metric, which can be decomposed into a sum over different layers. Note that solving the problem in Equation (3) is extremely effective, which only takes a few seconds on a single CPU by using the solver in [73].

Pruning–Quantization Allocation. After the layer-wise compression ratio is determined, we then search for the corresponding pruning–quantization allocation. For a specific layer l, given the founded compression ratio $\boldsymbol{\alpha}_l^+$, we first sample the corresponding search space according to $\boldsymbol{\alpha}_l^+$. In this case, we first sample the quantization bits α_l^q and then obtain the corresponding pruning mask α_l^p by using the $L1$ norm and ensuring $\alpha_l^{+,q} \times \alpha_l^{+,p} \approx \boldsymbol{\alpha}_l^+$, where $\alpha_l^{+,p}$ and $\alpha_l^{+,q}$ denote the compression ratio using the corresponding pruning mask and quantization bit. As we mentioned before, even though the search space is significantly reduced, it still requires a large amount of computational resources. Here, we use the proposed Hypothesis 1 to solve this issue. Assuming that there are K candidates for a layer l, we calculate the representation mutual information between each candidate $\boldsymbol{\alpha}^l$ and the original network to determine the optimal solution for this layer. Formally, according to the proposed hypothesis, the optimal solution is obtained by

$$\alpha^{*,l} =_{k\in[1..K]} \mathrm{I}\Big(f\Big(\boldsymbol{W}^l\Big), f\Big(\boldsymbol{W}^l, \boldsymbol{\alpha}^{l,k}\Big)\Big). \tag{4}$$

It is worth noting that after we traverse all the layers with Equation (4), the final result may not meet the compression constraint due to the approximation of the compression and the reduction in channels. We therefore further fine-tune the obtained pruning–quantization allocation, which aims to maximize the representation mutual information through different layers. In particular, similar to the compression ratio optimization step, we construct a linear programming problem with quadratic constraints, where the importance in Equation (2) is obtained between the original network and the compressed network.

An Extra Bonus: Optimal Teacher Assistant. The extensive experiments in Section 4 also demonstrate that Hypothesis 1 is satisfied with knowledge distillation. Specifically, compression configurations with high representation mutual information show better performance in KD. In other words, the optimal compression configuration that we found with the goal of optimizing representation mutual information in the previous steps can be used as a teacher assistant. More specifically, the compression configurations searched with different compression constraints can be used as teacher assistants to perform progressive KD [64] for the optimal performance. Our algorithm is also summarized in Algorithm 1.

Algorithm 1: Fully-AMC

Input: Sampled n i.i.d samples $\mathcal{D}$, pre-trained model $\mathcal{M}$, Constraints list $\mathbf{\Omega} = [\Omega_1, .., \Omega_M]$.

Output: Optimized architectures $\boldsymbol{\alpha}_1^*, \boldsymbol{\alpha}_2^*, ...\boldsymbol{\alpha}_M^*$.

for Ω_i *in* $\mathbf{\Omega}$ **do**
- Input $\mathcal{D}$ into $\mathcal{M}$ to generate layer-wise feature map $\boldsymbol{X}^1, ..., \boldsymbol{X}^L$;
- Obtain the similarity matrix $\boldsymbol{H}$;
- Calculate the layer-wise importance $\boldsymbol{i}$ by Equation (2);
- Obtain $\boldsymbol{\alpha}_i^+$ by using Equation (3);
- Obtain $\boldsymbol{\alpha}_i^*$ by using Equation (4);

end

3.3. Acceleration and Discussion

Acceleration of $\mathrm{I}(X, Y)$. Please note that the aforementioned mutual information is hard to compute in practice. On one hand, the distribution in $\mathrm{I}(X^i, X^j)$ is intractable. In addition, estimating the distribution of X^i is also time-consuming. On the other hand, hidden representation X^i has a high dimension, which suffers from the curse of dimensionality. Here, we introduce the normalized Hilbert–Schmidt Independence Criterion (HSIC) [72,74] (the normalized HSIC is also known as CKA [75], the RV coefficient [76], and Tucker's congruence coefficient [77]) to solve this problem. Specifically, let $\mathcal{D} := \{(x_1, y_1), ..., (x_n, y_n)\}$ contain n i.i.d samples drawn from the distribution P_{XY}. In this case, the normalized Hilbert–Schmidt Independence Criterion (HSIC) [72,74,75] is defined as

$$\mathrm{I}(X, Y) \approx \mathrm{nHSIC}_{\mathrm{linear}}(X, Y) = \frac{||\boldsymbol{Y}^T\boldsymbol{X}||_F^2}{||\boldsymbol{X}^T\boldsymbol{X}||_F||\boldsymbol{Y}^T\boldsymbol{Y}||_F}. \tag{5}$$

Efficiency Discussion. Compared to the previous methods [21,35,36], Fully-AMC is effective and efficient. In terms of efficiency, Fully-AMC jointly searches the optimal compression solution in an integrated search space, which may potentially find a better solution. Meanwhile, the proposed hypothesis provides an accurate and effective performance estimator, which is also generalized to the search space. In terms of effectiveness, Fully-AMC obtained the optimal solution directly and did not need any training process. In addition, the calculation of $\mathrm{I}(X, Y)$ is effective, i.e., $\mathcal{O}(n^2)$, where n is the number of samples. It only takes a few seconds on a single GPU and CPU. And the optimal solution is obtained within 1 GPU hours.

4. Experiments

We evaluate our proposed Fully-AMC framework on the ResNet-18 and ResNet-20 architectures. These models are selected as the benchmark architectures for several compelling reasons:

- **Representative Architecture**: ResNet models represent a fundamental and widely adopted CNN architecture that introduced residual connections [2]. Their structure has influenced numerous subsequent architectures, making them ideal candidates for demonstrating compression techniques that can generalize to other networks.
- **Industry Standard**: ResNet-18/20 are standard benchmarks in model compression research [21,35,57], enabling direct comparisons with state-of-the-art methods. Their moderate size allows for comprehensive analysis while maintaining practical relevance.
- **Practical Deployment**: These models offer a good balance between model capacity and computational efficiency, making them popular choices for real-world applications on

resource-constrained devices. ResNet-18's 11.7M parameters and ResNet-20's 0.27M parameters represent different scales of practical deployment scenarios.

- **Structural Characteristics**: The residual connections and layer organization in ResNet architectures present unique challenges for compression, including the following:
  - Skip connections that require careful handling during pruning;
  - Multiple layer types (conv, batch norm, and pooling) that benefit from different compression strategies;
  - Varying sensitivities to compression across different network depths.

  These characteristics make them excellent test cases for evaluating the effectiveness of our unified compression approach.

To demonstrate the generality of our method, we conduct experiments on both the CIFAR-10 and ImageNet datasets, with ResNet-20 for CIFAR-10 and ResNet-18 for ImageNet following standard practices in the field [11,36]. This choice allows us to evaluate our method's effectiveness across different model scales and dataset complexities.

Implementation Details. The proposed method is implemented by PyTorch. In our paper, we randomly obtain 64 training examples to obtain the layer-wise importance. Since pruning has a greater impact on performance than quantization, we also adopt a two-stage strategy to fine-tune the compressed networks on ImageNet. In particular, we first fine-tune the pruned networks without quantization until they roughly recover the performance of the original uncompressed models. And then, quantization is further applied at the second fine-tuning stage. Fully-AMC is extremely effective and only needs less than 1 h on a piece of Nvidia Geforce GTX 1080Ti and a single Intel(R) Xeon(R) CPU E5-2620 v4. In addition, Fully-AMC is flexible and is capable of searching for different channel and bit configs under the given compression constrains.

4.1. Comparison with SOTA Methods

Experiments on CIFAR-10. We first conduct the Fully-AMC experiments of ResNet-20 on CIFAR-10. Results are reported in Table 1. We compare our method with pruning methods and quantization methods. The pruning methods include magnitude-based (i.e., FPGM [78]), automatic (i.e., TAS [18]), and data-driven (i.e., SCOP [79]) methods. Meanwhile, the quantization methods include the low-bit quantization algorithm (LQ-Nets [80]) and mixed-precision quantization method (HAWQ [11]). In addition, we further compare Fully-AMC with PACT [81], which proposes a new activation function for QAT. In comparison with these methods, our method can obtain the Pareto optimum, which guarantees that at least one of the indicators is optimal among the compared methods listed in Table 1. With an eighty-fold compressed ratio, our method obtains a performance that, instead of being lower than the benchmark model, is 0.21 percentage points higher than the baseline model, which again demonstrates the effectiveness of our method. Note that our method only uses linear quantization to compress the network. In other words, it may have better performance when combining the quantization methods of LSQ [82] and PACT [81] with the searched compression configuration.

Table 1. Classification accuracy and compression ratio for the proposed method, pruning-based methods (the first block), and quantization methods (the second and third blocks).

Method	Top1	Acc↑↓	W/A bit	Size	Rate
Baseline	92.37	-	W32/A32	0.71	1.00×
FPGM [68]	91.09	−1.48	W32/A32	0.43	1.67×
Taylor [83]	91.51	−1.06	W32/A32	0.35	2.11×
DHP [84]	91.54	−1.00	W32/A32	0.35	2.08×
APS [17]	91.86	−0.71	W32/A32	0.34	1.94×
TAS [18]	91.99	−0.58	W32/A32	0.33	2.04×
ManiDP [85]	92.05	−0.17	W32/A32	0.32	2.18×
Ours	**92.58**	**0.21**	**mixed/A5**	**0.06**	**80.23×**
LQ-Nets [80]	92.00	−0.37	W2/A32	0.10	10.61×
LQ-Nets [80]	91.80	−0.57	W2/A32	0.10	15.91×
HAWQ [11]	92.22	−0.15	mixed/A4	0.10	67.56×
Ours	**92.58**	**0.21**	**mixed/A5**	**0.07**	**80.23×**
Dorefa [10]	89.90	−2.47	W3/A3	0.08	113.26×
PACT [81]	91.10	−1.27	W3/A3	0.08	113.26×
Ours	**91.32**	**−1.05**	**mixed/A5**	**0.10**	**116.70×**

Experiments on ImageNet. In order to fully validate the effectiveness of our algorithm on large-scale datasets, we further conducted experiments on the ILSVRC-2012 dataset. For ResNet-18, we compare Fully-AMC with pruning only (i.e., FPGM [68], DSA [86], and ManiDP [85]) and quantization only (i.e., RVQuant [87] and HAWQ-V3 [88]) methods. As shown in Table 2, Fully-AMC shows the best trade-off between accuracy and the compression ratio on ResNet-18. For example, Fully-AMC achieves higher accuracy with a smaller model size and fewer BOPs (72.08%, 6.1 MB, and 56G BOPs v.s. 70.22%, 6.7 MB, and 72G BOPs) compared with HAWQ-V3 [88]. As shown in Table 3, for ResNet-50, we further compare Fully-AMC with the prune-quant method (i.e., Deep Compression [30], CLIP-Q [34], and OPQ [35]). Results are reported in Table 1. The result shows that our algorithm achieves another Pareto dominance, reaching higher performance with a lower computation cost (76.45% top-1 and 95G BOPs), which is attributed to our joint optimization compression approach. Note that OPQ and CLIP-Q only involve sparsity, making it hard to perform acceleration, while Fully-AMC uses channel pruning with a simple $L1$ norm, which shows the superiority of our method. We then perform progressive distillation on the compressed network. Note that the capacity gap between the original network and compressed network is relatively large, so we select a 300G BOP-compressed network as a teacher assistant network to further improve the accuracy.

Table 2. Comparison results on ImageNet-1k with ResNet-18. "Uniform" represents uniform quantization.

Method	Type	Top1	Size (MB)	BOPs (G)	Uniform	W bit	A bit	Rate
Baseline	-	73.09	44.6	1858	-	32	32	1.00×
FPGM [68]	Prune	68.41	27.2	1088	-	32	32	1.71×
DSA [86]	Prune	68.61	28.0	1122	-	32	32	1.66×
ManiDP [85]	Prune	68.88	22.9	916	-	32	32	2.04×
RVQuant [87]	Quant	70.01	11.1	116	✗	8	8	16.13×
HAWQ-V3 [88]	Quant	71.56	11.1	116	✓	8	8	16.13×
Ours	**Joint**	**72.30**	**10.6**	**91**	✓	**mixed**	**8**	**20.56×**
PACT [81]	Quant	69.8	7.2	50	✓	5	5	37.42×
LQ-Nets [80]	Quant	70.00	5.8	225	✗	4	32	8.31×
HAWQ-V3 [88]	Quant	70.22	6.7	72	✓	mixed	mixed	25.98×
Ours	**Joint**	**72.08**	**6.1**	**56**	✓	**mixed**	**8**	**33.41×**

Table 3. Comparison results on ImageNet-1k with ResNet-50. "Ours+Distillation" represents further performing progressive distillation.

Method	Type	Top1	BOPs (G)	W bit	A bit	Rate
Baseline	-	77.72	3951	32	32	1×
ThiNet [23]	Prune	71.01	1908	32	32	2.07×
HAQ [57]	Quant	76.14	123	4	8	32×
ACIQ [89]	Quant	75.30	123	4	8	32×
CLIP-Q [34]	Joint	73.70	124	3.28	8	31.81×
Deep Compression [30]	Joint	76.15	123	4	8	32×
OPQ [35]	Joint	76.41	103	3.25	8	38.03×
Ours	Joint	**76.45**	**95**	mixed	**5**	**41.58**×
Ours+Distillation	Joint	**76.53**	**95**	mixed	**5**	**41.58**×

4.2. Ablation Study

In this section, we analyze the architecture found by Fully-AMC and validate the effectiveness of the proposed compression pipeline. To sum up, the architecture found by Fully-AMC shows surprisingly similar characteristics compared with the optimal one. Meanwhile, each step of the proposed pipeline shows a clear performance improvement. The detailed experiments are described as follows. We also conduct ablation studies on the hyperparameters, which are reported in the Supplementary Materials.

Effectiveness of the Representation Mutual Information. We conduct experiments to show the mutual information between the compressed network and original network during the training process. As shown in Figure 3, the compressed network with higher mutual information achieves higher accuracy after training. In other words, the mutual information can be considered as a good indicator that predicts the quality of the compressed network.

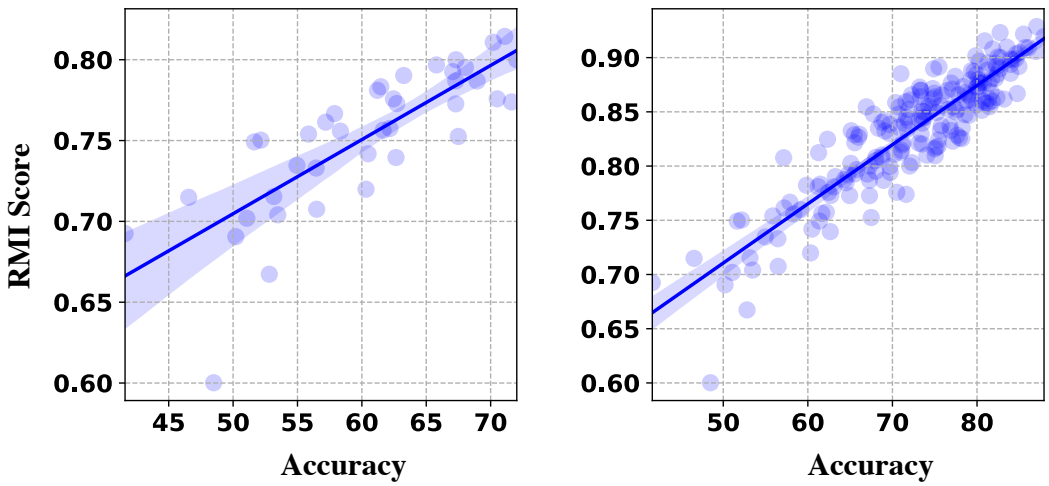


Figure 3. Ablation on mutual information between networks. (**Left**): Representation mutual information between original ResNet-20 and ResNet-20, which is randomly compressed to 2%. (**Right**): Representation mutual information between original ResNet-20 and ResNet-20 that is randomly compressed to a different compression rate. The horizontal axis is the accuracy of the compressed ResNet-20 after training.

Ablation on the Search Pipeline. There are two main steps in Fully-AMC. The first step is to obtain the compression rate of each layer by calculating the layer-wise importance. The second step is to obtain an allocation configuration of bits and channels. Therefore, we conduct an experiment to validate the contributions of these two steps by replacing each with a random search. The results are reported in the table. As we can see, our

method could bring about a 2∼3% performance improvement compared to the random compression method.

Effectiveness of the Optimal Teacher Assistant. As we mentioned in Section 3.2, the searched optimal compression configuration is the optimal teacher assistant. Here, we compare progressive distillation using different strategies in Figure 4. Specifically, the "original model" indicates that using an uncompressed model as an original teacher and "w/o distillation" trains the compressed architectures without distillation. From this figure, we can see that using an uncompressed model as an original teacher will decrease the accuracy of the student model, making it even inferior to the network without distillation training. So, we draw a conclusion that Fully-AMC indeed finds the optimal teacher assistant, which improves the performance with a clear margin.

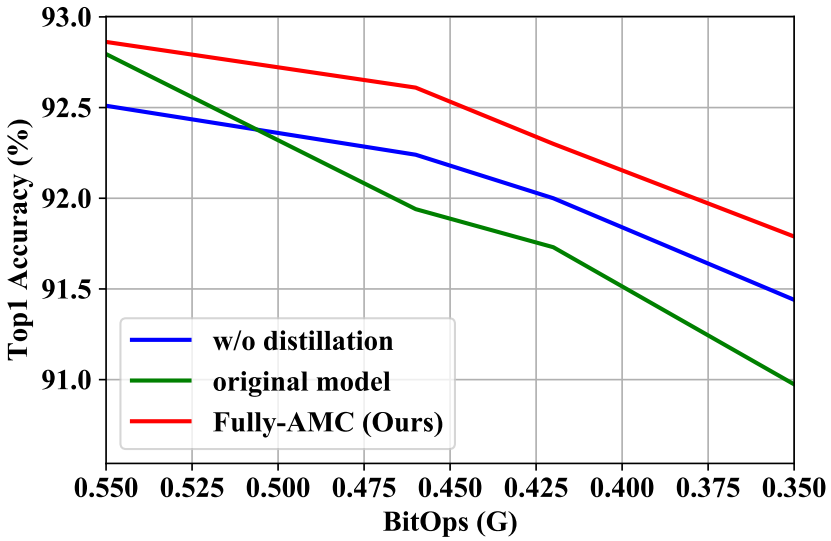


Figure 4. Top1 accuracy on CIFAR-10 between Fully-AMC for progressive distillation, original model, and without any distillation.

4.3. Addressing the Research Questions

The experimental results and analyses demonstrate that our proposed Fully-AMC framework effectively addresses the research questions posed in this study:

RQ1: Joint Optimization of Compression Strategies. Our framework successfully unifies pruning, quantization, and knowledge distillation through a novel compression pipeline, as evidenced by the superior performance achieved on both the CIFAR-10 and ImageNet datasets. The ablation studies in Table 4 demonstrate that our joint optimization approach outperforms random compression methods by 2-3% in accuracy.

RQ2: Reliability of Representation Mutual Information. The experiments validate our hypothesis about representation mutual information as a reliable compression quality metric. As shown in Figure 3, networks with higher mutual information consistently achieve better accuracy after training, confirming the effectiveness of this metric across different compression strategies.

RQ3: Efficient Search Space Navigation. Our approach efficiently navigates the vast search space through the following:

- Layer-wise self-representation mutual information analysis to determine compression ratios;
- Sampling-based pruning and quantization allocation;
- Progressive knowledge distillation using optimal compressed models.

This is demonstrated by the computational efficiency of our method, requiring less than 1 h on standard hardware while achieving state-of-the-art compression results.

Table 4. ResNet-20 accuracy of different step combinations on CIFAR-10 test set. LCR: layer-wise compression rate. PQA: pruning–quantization allocation. We sample 10 different network architectures for each configuration. - denotes using a random compression rate or random bit and channel allocation.

LCR	PQA	Top-1 ACC (%)
-	✓	89.21
✓	-	89.28
✓	✓	92.13

5. Discussion

Our experimental results and analyses reveal several key insights about automated model compression:

Performance Trade-offs. The relationship between the compression ratio and model accuracy follows a non-linear pattern:

- Moderate compression (up to 20×) maintains accuracy within 0.5;
- High compression (20×–30×) shows a gradual accuracy decline of 0.5–1.0;
- Extreme compression (>30×) leads to more significant performance degradation.

Algorithmic Insights. The effectiveness of our approach stems from three key mechanisms:

- Layer-wise mutual information analysis enables accurate importance estimation without expensive training;
- Joint optimization of pruning and quantization allows for the exploration of previously unexplored compression configurations;
- Progressive knowledge distillation with optimal teacher assistants helps mitigate accuracy loss.

Limitations and Future Work. While our framework demonstrates significant advantages, several limitations warrant further research:

1. *Architecture Generalization*:

- The current evaluation focuses primarily on ResNet architectures;
- Additional validation is needed for modern architectures (e.g., Transformers);
- There are potential challenges with complex skip connections.

2. *Theoretical Foundations*:

- A lack of formal guarantees on compression optimality;
- A limited understanding of mutual information bounds across architectures;
- A need for a theoretical framework for compression ratio optimization.

Future research directions include the following:

- Extending support to emerging architecture families;
- Developing more efficient mutual information approximation methods;
- Establishing theoretical bounds for compression–accuracy trade-offs;
- Investigating dynamic adaptation mechanisms for varying hardware conditions;
- Applying the framework to other deep learning tasks.

6. Conclusions

This paper presents Fully-AMC, a novel framework for automated neural network compression that unifies pruning, quantization, and knowledge distillation through representation mutual information optimization. Our approach achieves state-of-the-art compression ratios while maintaining model accuracy, demonstrated by achieving a 33.41× compression ratio on ResNet-18 with only a 1.01% accuracy loss. The key innovation

lies in using representation mutual information as a reliable proxy for compression quality, enabling an efficient search of the vast compression configuration space without expensive training iterations. Through extensive experiments on CIFAR-10 and ImageNet, we validate both the effectiveness and efficiency of our approach, requiring only a few GPU hours for optimization. This work establishes a promising direction for practical neural network deployment on resource-constrained devices while maintaining high performance. An associated Github repository is available at https://github.com/AreckOVO/Probabilistic-Automated-Model-Compression-via-Representation-Mutual-Information-Optimization (accessed on 26 December 2024).

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math13010108/s1.

Author Contributions: Conceptualization, W.N., S.Z. and X.Z.; Methodology, W.N., S.Z. and X.Z.; Software, W.N.; Validation, W.N.; Writing—original draft, W.N.; Writing—review & editing, X.Z.; Supervision, S.Z. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology Major Project: No. 2022ZD0118202; The National Science Fund for Distinguished Young Scholars: No. 62025603; The National Natural Science Foundation of China: No. U21B2037, No. U22B2051, No. U23A20383, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401; The Natural Science Foundation of Fujian Province of China: No. 2021J06003 and No. 2022J06001.

Data Availability Statement: Data supporting reported results can be found: https://github.com/AreckOVO/Probabilistic-Automated-Model-Compression-via-Representation-Mutual-Information-Optimization (accessed on 26 December 2024).

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

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012.
4. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2017**, *39*, 1137–1149. [CrossRef]
6. Girshick, R. Fast R-CNN. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
7. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2018**, *40*, 834–848. [CrossRef] [PubMed]
8. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
9. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June2018.
10. 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.
11. Dong, Z.; Yao, Z.; Gholami, A.; Mahoney, M.W.; Keutzer, K. Hawq: Hessian aware quantization of neural networks with mixed-precision. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 293–302.
12. Zhou, A.; Yao, A.; Guo, Y.; Xu, L.; Chen, Y. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv* **2017**, arXiv:1702.03044.

13. Chen, S.; Wang, W.; Pan, S.J. Metaquant: Learning to quantize by learning to penetrate non-differentiable quantization. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 3916–3926.
14. Cai, Z.; He, X.; Sun, J.; Vasconcelos, N. Deep learning with low precision by half-wave gaussian quantization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5918–5926.
15. Li, Y.; Gong, R.; Tan, X.; Yang, Y.; Hu, P.; Zhang, Q.; Yu, F.; Wang, W.; Gu, S. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv* **2021**, arXiv:2102.05426.
16. Yang, T.J.; Howard, A.; Chen, B.; Zhang, X.; Go, A.; Sandler, M.; Sze, V.; Adam, H. Netadapt: Platform-aware neural network adaptation for mobile applications. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 285–300.
17. Wang, J.; Bai, H.; Wu, J.; Shi, X.; Huang, J.; King, I.; Lyu, M.; Cheng, J. Revisiting Parameter Sharing for Automatic Neural Channel Number Search. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5991–6002.
18. Dong, X.; Yang, Y. Network pruning via transformable architecture search. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
19. Liu, Z.; Mu, H.; Zhang, X.; Guo, Z.; Yang, X.; Cheng, K.T.; Sun, J. Metapruning: Meta learning for automatic neural network channel pruning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3296–3305.
20. Yu, J.; Huang, T. Autoslim: Towards one-shot architecture search for channel numbers. *arXiv* **2019**, arXiv:1903.11728.
21. He, Y.; Lin, J.; Liu, Z.; Wang, H.; Li, L.J.; Han, S. Amc: Automl for model compression and acceleration on mobile devices. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–800.
22. Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Learning structured sparsity in deep neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Barcelona, Spain, 5–10 December 2016.
23. Luo, J.H.; Wu, J.; Lin, W. Thinet: A filter level pruning method for deep neural network compression. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
24. Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning filters for efficient convnets. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
25. Guo, J.; Han, K.; Wang, Y.; Wu, H.; Chen, X.; Xu, C.; Xu, C. Distilling Object Detectors via Decoupled Features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2154–2164.
26. Zhang, Z.; Zhang, H.; Arik, S.O.; Lee, H.; Pfister, T. Distilling effective supervision from severe label noise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9294–9303.
27. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *Stat* **2015**, *1050*, 9.
28. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 9163–9171.
29. Pan, W.; Gao, T.; Zhang, Y.; Zheng, X.; Shen, Y.; Li, K.; Hu, R.; Liu, Y.; Dai, P. Semi-Supervised Blind Image Quality Assessment through Knowledge Distillation and Incremental Learning. *Proc. Aaai Conf. Artif. Intell.* **2024**, *38*, 4388–4396. [CrossRef]
30. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016; Conference Track Proceedings.
31. Blalock, D.; Ortiz, J.J.G.; Frankle, J.; Guttag, J. What is the state of neural network pruning? *arXiv* **2020**, arXiv:2003.03033.
32. Liu, Z.; Sun, M.; Zhou, T.; Huang, G.; Darrell, T. Rethinking the Value of Network Pruning. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
33. Liu, Y.; Jia, X.; Tan, M.; Vemulapalli, R.; Zhu, Y.; Green, B.; Wang, X. Search to distill: Pearls are everywhere but not the eyes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7539–7548.
34. Tung, F.; Mori, G. Clip-q: Deep network compression learning by in-parallel pruning-quantization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7873–7882.
35. Hu, P.; Peng, X.; Zhu, H.; Aly, M.M.S.; Lin, J. OPQ: Compressing Deep Neural Networks with One-shot Pruning-Quantization. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21), Vancouver, VN, Canada, 2–9 February 2021; pp. 2–9.
36. Wang, T.; Wang, K.; Cai, H.; Lin, J.; Liu, Z.; Wang, H.; Lin, Y.; Han, S. Apq: Joint search for network architecture, pruning and quantization policy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 2078–2087.
37. Li, X.; Gao, T.; Zheng, X.; Hu, R.; Zheng, J.; Shen, Y.; Li, K.; Liu, Y.; Dai, P.; Zhang, Y.; et al. Adaptive Feature Selection for No-Reference Image Quality Assessment using Contrastive Mitigating Semantic Noise Sensitivity. *arXiv* **2023**, arXiv:2312.06158.

38. Li, X.; Hu, R.; Zheng, J.; Zhang, Y.; Zhang, S.; Zheng, X.; Li, K.; Shen, Y.; Liu, Y.; Dai, P.; et al. Integrating Global Context Contrast and Local Sensitivity for Blind Image Quality Assessment. 2024. Available online: https://openreview.net/forum?id=MRYS3Zb4iV (accessed on 17 November 2024).
39. Gao, T.; Chen, P.; Zhang, M.; Fu, C.; Shen, Y.; Zhang, Y.; Zhang, S.; Zheng, X.; Sun, X.; Cao, L.; et al. Cantor: Inspiring Multimodal Chain-of-Thought of MLLM. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, Australia, 28 October–1 November 2024.
40. Qin, G.; Hu, R.; Liu, Y.; Zheng, X.; Liu, H.; Li, X.; Zhang, Y. Data-Efficient Image Quality Assessment with Attention-Panel Decoder. *Proc. Aaai Conf. Artif. Intell.* **2023**, *37*, 2091–2100. [CrossRef]
41. Zhang, S.; Jia, F.; Wang, C.; Wu, Q. Targeted hyperparameter optimization with lexicographic preferences over multiple objectives. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
42. Zhang, S.; Wu, Y.; Zheng, Z.; Wu, Q.; Wang, C. Hypertime: Hyperparameter optimization for combating temporal distribution shifts. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, Australia, 28 October–1 November 2024; pp. 4610–4619.
43. Xia, X.; Liu, J.; Zhang, S.; Wu, Q.; Wei, H.; Liu, T. Refined Coreset Selection: Towards Minimal Coreset Size under Model Performance Constraints. In Proceedings of the Forty-First International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024.
44. Zhang, S.; Zhang, J.; Ding, D.; Garcia, M.H.; Mallick, A.; Madrigal, D.; Xia, M.; Rühle, V.; Wu, Q.; Wang, C. EcoAct: Economic Agent Determines When to Register What Action. *arXiv* **2024**, arXiv:2411.01643.
45. Zhang, S.; Zhang, J.; Liu, J.; Song, L.; Wang, C.; Krishna, R.; Wu, Q. Offline Training of Language Model Agents with Functions as Learnable Weights. In Proceedings of the Forty-First International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024.
46. Zhang, S.; Xia, X.; Wang, Z.; Chen, L.H.; Liu, J.; Wu, Q.; Liu, T. Ideal: Influence-driven selective annotations empower in-context learners in large language models. *arXiv* **2023**, arXiv:2310.10873.
47. Ding, Y.; Robinson, N.; Zhang, S.; Zeng, Q.; Guan, C. TSception: Capturing Temporal Dynamics and Spatial Asymmetry From EEG for Emotion Recognition. *IEEE Trans. Affect. Comput.* **2023**, *14*, 2238–2250. [CrossRef]
48. Ding, Y.; Robinson, N.; Tong, C.; Zeng, Q.; Guan, C. LGGNet: Learning From Local-Global-Graph Representations for Brain–Computer Interface. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 9773–9786. [CrossRef] [PubMed]
49. Ding, Y.; Zhang, S.; Tang, C.; Guan, C. MASA-TCN: Multi-Anchor Space-Aware Temporal Convolutional Neural Networks for Continuous and Discrete EEG Emotion Recognition. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 3953–3964. [CrossRef] [PubMed]
50. Ding, Y.; Guan, C. GIGN: Learning Graph-in-graph Representations of EEG Signals for Continuous Emotion Recognition. In Proceedings of the 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Sydney, Australia, 24–27 July 2023; pp. 1–5. [CrossRef]
51. Ding, Y.; Li, Y.; Sun, H.; Liu, R.; Tong, C.; Liu, C.; Zhou, X.; Guan, C. EEG-Deformer: A Dense Convolutional Transformer for Brain-computer Interfaces. *arXiv* **2024**, arXiv:2405.00719.. [CrossRef]
52. Cai, Y.; Yao, Z.; Dong, Z.; Gholami, A.; Mahoney, M.W.; Keutzer, K. Zeroq: A novel zero shot quantization framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13169–13178.
53. Nagel, M.; Baalen, M.v.; Blankevoort, T.; Welling, M. Data-free quantization through weight equalization and bias correction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1325–1334.
54. Wu, B.; Wang, Y.; Zhang, P.; Tian, Y.; Vajda, P.; Keutzer, K. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv* **2018**, arXiv:1812.00090.
55. Yu, H.; Han, Q.; Li, J.; Shi, J.; Cheng, G.; Fan, B. Search what you want: Barrier panelty NAS for mixed precision quantization. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–16.
56. Elthakeb, A.T.; Pilligundla, P.; Mireshghallah, F.; Yazdanbakhsh, A.; Esmaeilzadeh, H. Releq: A reinforcement learning approach for deep quantization of neural networks. *arXiv* **2018**, arXiv:1811.01704. [CrossRef] [PubMed]
57. Wang, K.; Liu, Z.; Lin, Y.; Lin, J.; Han, S. Haq: Hardware-aware automated quantization with mixed precision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8612–8620.
58. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv* **2014**, arXiv:1412.6550.
59. Heo, B.; Lee, M.; Yun, S.; Choi, J.Y. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3779–3787.

60. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv* **2016**, arXiv:1612.03928.
61. Cho, J.H.; Hariharan, B. On the efficacy of knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4794–4802.
62. Furlanello, T.; Lipton, Z.; Tschannen, M.; Itti, L.; Anandkumar, A. Born again neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 1607–1616.
63. Jin, X.; Peng, B.; Wu, Y.; Liu, Y.; Liu, J.; Liang, D.; Yan, J.; Hu, X. Knowledge distillation via route constrained optimization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1345–1354.
64. Mirzadeh, S.I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5191–5198.
65. 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.
66. Tung, F.; Mori, G. Deep neural network compression by in-parallel pruning-quantization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 568–579. [CrossRef] [PubMed]
67. Yang, H.; Gui, S.; Zhu, Y.; Liu, J. Automatic neural network compression by sparsity-quantization joint learning: A constrained optimization-based approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2178–2188.
68. He, Y.; Liu, P.; Wang, Z.; Hu, Z.; Yang, Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4340–4349.
69. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
70. Trofimov, I.; Klyuchnikov, N.; Salnikov, M.; Filippov, A.; Burnaev, E. Multi-fidelity neural architecture search with knowledge distillation. *arXiv* **2020**, arXiv:2006.08341. [CrossRef]
71. Wu, M.; Ma, W.; Li, Y.; Zhao, X. Automatic Optimization of super Parameters Based on Model Pruning and Knowledge Distillation. In Proceedings of the 2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC), Chongqing, China, 6–8 November 2020; IEEE: Piscataway Township, NJ, USA, 2020; pp. 111–116.
72. Zheng, X.; Ma, Y.; Xi, T.; Zhang, G.; Ding, E.; Li, Y.; Chen, J.; Tian, Y.; Ji, R. An Information Theory-inspired Strategy for Automatic Network Pruning. *arXiv* **2021**, arXiv:2108.08532.
73. Kraft, D. A Software Package for Sequential Quadratic Programming. 1988. Available online: https://degenerateconic.com/uploads/2018/03/DFVLR_FB_88_28.pdf (accessed on 17 November 2024).
74. Gretton, A.; Bousquet, O.; Smola, A.; Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt norms. In Proceedings of the International Conference on Algorithmic Learning Theory, Singapore, 8–11 October 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 63–77.
75. Kornblith, S.; Norouzi, M.; Lee, H.; Hinton, G. Similarity of neural network representations revisited. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 3519–3529.
76. Robert, P.; Escoufier, Y. A unifying tool for linear multivariate statistical methods: The RV-coefficient. *J. R. Stat. Soc. Ser. C Appl. Stat.* **1976**, *25*, 257–265. [CrossRef]
77. Lorenzo-Seva, U.; Ten Berge, J.M. Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology* **2006**, *2*, 57–64. [CrossRef]
78. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. HRank: Filter Pruning using High-Rank Feature Map. *IEEE Conf. Comput. Vis. Pattern Recognit.* **2020**, *33*, 10936–10947.
79. Tang, Y.; Wang, Y.; Xu, Y.; Tao, D.; Xu, C.; Xu, C.; Xu, C. Scop: Scientific control for reliable neural network pruning. *arXiv* **2020**, arXiv:2010.10732.
80. Zhang, D.; Yang, J.; Ye, D.; Hua, G. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 365–382.
81. Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P.I.J.; Srinivasan, V.; Gopalakrishnan, K. Pact: Parameterized clipping activation for quantized neural networks. *arXiv* **2018**, arXiv:1805.06085.
82. Esser, S.K.; McKinstry, J.L.; Bablani, D.; Appuswamy, R.; Modha, D.S. Learned step size quantization. *arXiv* **2019**, arXiv:1902.08153.
83. Molchanov, P.; Mallya, A.; Tyree, S.; Frosio, I.; Kautz, J. Importance Estimation for Neural Network Pruning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
84. Li, Y.; Gu, S.; Zhang, K.; Van Gool, L.; Timofte, R. Dhp: Differentiable meta pruning via hypernetworks. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part VIII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 608–624.

85. Tang, Y.; Wang, Y.; Xu, Y.; Deng, Y.; Xu, C.; Tao, D.; Xu, C. Manifold regularized dynamic network pruning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5018–5028.
86. Ning, X.; Zhao, T.; Li, W.; Lei, P.; Wang, Y.; Yang, H. Dsa: More efficient budgeted pruning via differentiable sparsity allocation. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part III 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 592–607.
87. Park, E.; Yoo, S.; Vajda, P. Value-aware quantization for training and inference of neural networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 580–595.
88. Yao, Z.; Dong, Z.; Zheng, Z.; Gholami, A.; Yu, J.; Tan, E.; Wang, L.; Huang, Q.; Wang, Y.; Mahoney, M.; et al. Hawq-v3: Dyadic neural network quantization. In Proceedings of the International Conference on Machine Learning, PMLR, Online, 18–24 July 2021; pp. 11875–11886.
89. Banner, R.; Nahshan, Y.; Hoffer, E.; Soudry, D. Aciq: Analytical Clipping for Integer Quantization of Neural Networks. 2018. Available online: https://openreview.net/forum?id=B1x33sC9KQ (accessed on 17 November 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.