Neural Networks 171 (2024) 104–113

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Towards performance-maximizing neural network pruning via global channel attention

Yingchun Wang [a,b], Song Guo [b], Jingcai Guo [b], Jie Zhang [b], Weizhan Zhang [a,*], Caixia Yan [a], Yuanhong Zhang [a]

[a] BDKE Lab, School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region of China

ARTICLE INFO

Keywords:
Model compression
Channel pruning
Global attention
Edge computing
Learn-to-rank

ABSTRACT

Network pruning has attracted increasing attention recently for its capability of transferring large-scale neural networks (e.g., CNNs) into resource-constrained devices. Such a transfer is typically achieved by removing redundant network parameters while retaining its generalization performance in a static or dynamic manner. Concretely, static pruning usually maintains a larger and fit-to-all (samples) compressed network by removing the same channels for all samples, which cannot maximally excavate redundancy in the given network. In contrast, dynamic pruning can adaptively remove (more) different channels for different samples and obtain state-of-the-art performance along with a higher compression ratio. However, since the system has to preserve the complete network information for sample-specific pruning, the dynamic pruning methods are usually not memory-efficient. In this paper, our interest is to explore a static alternative, dubbed GlobalPru, from a different perspective by respecting the differences among data. Specifically, a novel channel attention-based learn-to-rank framework is proposed to learn a global ranking of channels with respect to network redundancy. In this method, each sample-wise (local) channel attention is forced to reach an agreement on the global ranking among different data. Hence, all samples can empirically share the same ranking of channels and make the pruning statically in practice. Extensive experiments on ImageNet, SVHN, and CIFAR-10/100 demonstrate that the proposed GlobalPru achieves superior performance than state-of-the-art static and dynamic pruning methods by significant margins.

1. Introduction

Convolution neural networks (CNNs) have achieved great success in many visual recognition tasks including image classification (Khan, Fu, Brent, Luo, & Wu, 2023; Wei, Du, Wan, & Min, 2023), object detection (Ren, He, Girshick, & Sun, 2015), image segmentation (Kuang et al., 2023; Zhang et al., 2021), etc. The success of CNNs is inseparable from an excessive number of parameters that are well organized to perform sophisticated computations. Being computationally expensive, deep learning applications and services are usually deployed on resourceful servers and far away from end users, since mobile devices are usually computational and storage-limited. These two contradictory properties pose great challenges for deploying edge intelligence in the real world (Wang et al., 2022).

Network pruning has been proposed to reduce the deep model's resource cost without a significant drop in accuracy. The popular pruning methods could be divided into two main categories including static and dynamic pruning. Static pruning (Tang et al., 2020; Zhuang et al., 2020), as a training-time pruning paradigm, removes model redundancy uniformly and obtains a compact model across all samples. It is highly deployment-efficient (*only the pruned model*) by ignoring the instance-specific variations. In fact, natural images are of different channel attention for a given neural network, e.g., recognizing a giraffe focuses more on the edge features while a Sphynx may need more attention to its textures. We define the model's channel attention focused on an individual instance as **local channel attention**. Therefore, although deployment-efficient, statically trimming the average redundancy across the entire dataset might not yield maximal sparsity and accuracy gains, facing serious constraints in pruning ratios and model performance.

A higher sparsity ratio is possible for considering instances' variations. Recently, some work proposed dynamic pruning techniques

* Corresponding author.
E-mail addresses: 20116342r@connect.polyu.hk (Y. Wang), song.guo@polyu.edu.hk (S. Guo), jc-jingcai.guo@polyu.edu.hk (J. Guo), 18104473r@connect.polyu.hk (J. Zhang), zhangwzh@xjtu.edu.cn (W. Zhang).

https://doi.org/10.1016/j.neunet.2023.11.065
Received 27 June 2023; Received in revised form 24 November 2023; Accepted 29 November 2023
Available online 1 December 2023

Table 1
Difference between static pruning methods and dynamic pruning methods.

Method	When to prune	How to prune	What to prune	Auxiliary unit	Pruning rate	Memory-efficient
Static pruning	Training-time	Permanent	Common redundancy	✓	Low	✓
Dynamic pruning	Running-time	Temporary	Sample-wise redundancy	✗	High	✗
GlobalPru	Training-time	Permanent	Majority-voted redundancy	✗	High	✓

(Tang et al., 2021; Wang, Zhang, Hu, Zhang, & Su, 2020). They focus on local channel attention and tailor inference paths to suit the characteristics of individual samples. In this way, dynamic pruning yields superior model efficiency via fine-grained and instance-wise sparsity. Unfortunately, the adaptively accelerated nature of dynamic pruning necessitates *the maintenance of the original unpruned model* throughout the whole inference. No matter how impressive dynamic methods appear, their value is negated by the challenge of application on resource-constrained devices.

To alleviate the memory requirements and achieve maximal pruning performance simultaneously, we propose injecting the data awareness of dynamic pruning into deployment-friendly static pruning methods. In this paper, we propose a new pruning paradigm, named GlobalPru, to address the inconsistency in model channel importance across different samples. Specifically, GlobalPru forces all images to agree on the same ranking of channel saliency (referred to as **global channel ranking**) through a learn-to-rank regularization. The whole pipeline can be divided into two stages. (i) We first use a majority-voting-based strategy to find the most recognized global ranking (also to stabilize the following training process). (2) Then, all the image-specific channel rankings are forced to agree on the same ranking prior via a learn-to-rank regularization during parameters' optimization. As a result, GlobalPru avoids the disadvantage of existing dynamic pruning which stores the entire model and performs more efficient pruning on globally ordered channels. We list the difference of the above-mentioned three pruning paradigms in Table 1. To our knowledge, this is the first endeavor to address the inherent conflict between static and dynamic pruning, while simultaneously integrating the strengths of both approaches.

Our contributions are summarized as follows:

- We propose GlobalPru, a static network pruning method. GlobalPru tackles the issue of image-specific channel redundancy faced by existing static methods by learning a global ranking of channels w.r.t redundancy. GlobalPru produces a pruned network such that GlobalPru is a more memory-efficient solution than existing dynamic methods.
- To the best of our knowledge, we are the first to propose a global channel attention mechanism where all the images share the same ranking of channels w.r.t. importance. Instead of repeatedly computing image-specific channel rankings under existing local attention mechanisms, our proposed global attention enriches the representation capacity of models and therefore greatly improves the pruning efficiency.
- Extensive experimental results show that GlobalPru can achieve state-of-the-art performance with almost all popular convolution neural network architectures.

2. Related work

2.1. Unstructured, structured and semi-structured pruning

Model pruning methods can be divided into three main categories based on the granularity of the pruning process: unstructured pruning, structured pruning, and the recent semi-structured pruning. Unstructured pruning (Chen, Zhu, Jiang, & Tsui, 2020; Kwon et al., 2020) focuses on individual weights and can result in higher compression ratios without causing a significant drop in accuracy. However, unstructured pruning methods are typically not hardware-friendly. It is hard to get practical speedup in modern computing platforms such as CPU and GPU due to the irregularity. In contrast, structured pruning (He, Lin, et al., 2018) is more hardware-friendly as it typically removes entire structured model branches (He, Zhang, & Sun, 2017a; Liu, Li, Shen, Huang, Yan, & Zhang, 2017a; Zhuang et al., 2020). However, although leads to better hardware utilization, structured pruning usually compromises the sparsity for maintaining accuracy.

More recently, semi-structured pruning has emerged as a prominent research focus. Positioned between fine-grained weight pruning and coarse-grained filter pruning, it targets the reduction of weights while adhering to specific distribution rules. A significant instance is NVIDIA's N:M sparsity scheme, where N out of M contiguous weights are set to zero (Zhang et al., 2022, 2023; Zhou et al., 2021). Currently, the pattern only achieves acceleration at a sparsity ratio of 2:4, let alone the fact that it is exclusively supported by the sparse matrix multiplication-accumulate instruction specially designed for NVIDIA A100. Another approach is the block-structured sparsity originated from Google's work (Elsen, Dukhan, Gale, & Simonyan, 2020). This approach usually clusters the irregularly distributed weights with small values into structured groups or entails a contiguous group of output channels sharing an identical sparsity pattern. Their practical speedup is mostly hardware-oriented and lacks broad adaptability.

In summary, although semi-structured pruning strikes an optimal balance between accuracy reduction and model compression, however, semi-structured sparsity often requires specific operators and hardware support. Given these considerations, in this work, we focus on the most popular channel pruning, as it provides a good balance between preserving the model structure and allowing for fine-grained pruning.

2.2. Static and dynamic pruning

Static pruning is the most traditional and classic model pruning method, which is based on the idea of sharing a compact model among all different samples (Liebenwein, Baykal, Lang, Feldman, & Rus, 2019; Liu et al., 2017a; Molchanov, Mallya, Tyree, Frosio, & Kautz, 2019; Tang et al., 2020; Wen, Wu, Wang, Chen, & Li, 2016). This method selects the pruning results through trade-offs on different samples, which can lead to a final compact model that has limited representation capacity and thus results in an obvious accuracy drop with large pruning rates.

Recently, researchers have shifted their focus to the pursuit of the ultimate pruning rate and have started excavating sample-wise model redundancy, leading to the development of dynamic pruning (Dong, Huang, Yang, & Yan, 2017; Gao, Zhao, Dudziak, Mullins, & Xu, 2018; Hua, Zhou, De Sa, Zhang, & Suh, 2019; Rao, Lu, Lin, & Zhou, 2018; Tang et al., 2021). Dynamic pruning is a novel approach that generates different compact models for different samples. Unlike static pruning, dynamic pruning uses a path-decision module to find the optimal model path for each input during inference, which allows for a higher compression rate and improved accuracy compared to static pruning.

Despite the advantages of dynamic pruning, most dynamic pruning methods are not memory-efficient. This is because most of these methods require deploying the full model even in the inference stage, which can result in increased latency and decreased efficiency. To address these limitations, state-of-the-art works in the field aim to improve dynamic pruning efficiency by incorporating sample-wise information into the pruning process. For instance, one work (Liu, Wang, Han, Xu, & Xu, 2019) employs a feature decay regularization to identify

informative features for different samples, while another (Tang et al., 2021) embeds the manifold information of all samples into the space of pruned networks.

In conclusion, dynamic pruning is a promising development in the field of model pruning, offering an exciting opportunity for improving the efficiency of deep neural networks. However, to fully realize its potential, more research is needed to address the memory limitations and further improve the accuracy and compression rate of these models.

2.3. Channel attention

Channel attention is a technique used to enhance the importance of certain features and diminish the significance of others in deep neural networks. This is achieved by adding scale coefficients on feature channels, which can be thought of as an extension of the inter-channel relationship for input feature maps. One common way to implement channel attention is through extra channel attention modules, which are designed to recalibrate channel-wise feature responses in each convolutional layer by explicitly modeling inter-dependencies between channels.

A well-known example of a channel attention mechanism is SENet (Hu, Shen, & Sun, 2017), which uses the "Squeeze-and-Excitation" (SE) block. This block can be stacked together to adaptively adjust the channel-wise feature responses and is implemented by explicitly modeling the inter-dependencies between channels. Another recent development in this field is the Selective Kernel (SK) unit (Li, Wang, Hu, & Yang, 2019), which allows each neuron in a Convolutional Neural Network (CNN) to adaptively adjust its receptive field size based on multiple scales of input information. By dynamically calculating channel attention for different kernels, the SK unit realizes parameter sharing and significantly improves the model's efficiency.

It is worth noting that despite the numerous advances in channel attention, nearly all existing approaches compute the local attention, meaning that the channel saliency is specific to each image and cannot identify global channel attention over the entire dataset. As a result, there is still room for improvement and innovation in this field, and researchers are actively exploring new and more effective ways to implement channel attention in deep neural networks.

3. Global attention-based channel pruning

In this section, we will give a detailed formulation and theoretical explanation of the proposed Global Channel Attention Pruning (GlobalPru). As illustrated in Section 3.1, GlobalPru is formulated as a static alternative regularized by channel importance ranking. Then, the training pipeline of GlobalPru can be divided into two stages including Global Channel Attention Election (stage 1) and Learn-to rank Regularization Pruning (stage 2), which will be illustrated and theoretically analyzed in Sections 3.3 and 3.4, respectively. Specifically, in stage 1, GlobalPru explicitly elects the global channel attention that benefits most samples from the observations of the local image-specific channel dependencies. After that, in stage 2, the proposed learn-to-rank algorithm forces the channels to be ordered toward the global channel attention concurrently during the model training.

3.1. Preliminaries

Given a dataset with N samples as $X = \{x_i\}_{i=1}^{N}$ with the corresponding labels $Y = \{y_i\}_{i=1}^{N}$. For a convolution neural network (CNN) with L convolution layers and parameters set Θ, $W^l \in R^{c^l \times c^{l-1} \times k^l \times k^l}$ denotes the convolution parameters in the lth layer, where c^l is the channel numbers of lth layer and k^l represents the corresponding kernel size. $F^l(x_i) \in R^{b \times c^l \times h \times w}$ is the output feature map of the lth layer for the input x_i, where b, h, w are the batch size, height and width of the output feature map, respectively. Given the input feature $F^{l-1}(x_i)$, the output of layer l can be calculated as $F^l(x_i) = Relu(Bn(W^l \otimes F^{l-1}))$, where $Relu$ and Bn represent the activation and batch normalization operation respectively. Finally, we use $f(X, \Theta)$ to represent the output of the neural network over input samples X.

Static channel pruning eliminates the same redundancies and discovers a compact network among all images. Most static methods are guided by empirical risk minimization and numeric-based regular terms, i.e., parameter magnitudes, channel saliencies, or reconstruction errors, etc. Typical static methods can be formulated as follows:

$$\min_{\Theta} \sum_{i=1}^{N} \mathscr{L}\left(f(x_i, \Theta), y_i\right) + \lambda \cdot Norm(\Theta), \tag{1}$$

where $\mathscr{L}$ denotes the loss function and $Norm(\cdot)$ is the regularization term which is usually a human-designed criterion for inducing model sparsity. And λ is used as a knob to strike the different trade-offs between model accuracy and sparsity ratio.

Conversely, dynamic channel pruning discovers effective sub-networks for each input dependently during the inference stage, which is usually implemented through additional model path-finding functions. Consider a channel scoring module S (e.g., a squeeze and excitation channel attention module (Hu et al., 2017)). For a specific input x_i, the channel saliencies in layer l can be computed as $\pi^l(x_i) = S^l(x_i) \in \mathbb{R}^{c^l}$. Each element $\pi_j^l(x_i)$, herein, is a numerical value produced by a Sigmoid function and ranges from 0 to 1, representing the relative importance of the jth channel. A smaller π_j^l corresponds to a less significant channel j. This property allows $\pi^l(x_i)$ to serve as the probability of dropping the jth channel. To achieve adaptive model sparsity, these methods usually maintain a binary decision mask to indicate whether to drop or keep each channel, i.e., $mask_j^l = 0$ removes the jth channel on layer l. These mask elements are initialized to 1 and there are two common mask strategies. **Strategy 1:** given a pruning threshold ϵ^l, $mask_j^l$ is set to 0 when $\pi_j^l(x_i) < \epsilon^l$; **Strategy 2:** we can sample $mask_j^l$ from a Bernoulli distribution with the probability $\pi_j^l(x_i)$. However, both the comparison and sampling from π are non-differentiable, which impedes the end-to-end training. Thus, Gumbel-Softmax has been a popular trick for mask learning via relaxing the binary decisions to soft ones during training (Jang, Gu, & Poole, 2017), and tightening to hard ones for the inference. The local channel attention could be re-calculated as:

$$\hat{\pi}^l(x_i) = \pi^l(x_i) \otimes mask^l(x_i), \tag{2}$$

and the feature output of the lth layer would be:

$$F^l(x_i) = Relu\left(BN\left(\left(mask^l(x_i) \cdot \hat{F}^{l-1}(x_i)\right) \otimes W^l\right)\right). \tag{3}$$

Finally, a general dynamic pruning paradigm could be formulated as Eq. (4), wherein the regularization is used to induce the instant-wise network sparsity:

$$\min_{\Theta} \sum_{i=1}^{N} \mathscr{L}\left(f(x_i, \Theta), y_i\right) + \lambda \sum_{l=1}^{L} \|\pi^l(x_i)\|. \tag{4}$$

In this way, dynamic pruning has achieved higher compression rates than static methods by removing sample-specific model redundancy. However, this inference-time path routing depends on the complete referenced model along with additional path-addressing units being deployed on resource-limited devices. These inherent dependencies render dynamic methods inefficient in terms of memory and computation, limiting their practical applicability.

3.2. Problem formulation

To harness the benefits of both static and dynamic methods while circumventing their associated limitations, GlobalPru uses the strategy of uniting all samples to achieve a consensus on channel importance rankings during training. Specifically, GlobalPru initiates by scrutinizing local channel attentions across various samples through a pre-trained scoring module S (the squeeze and excitation module).

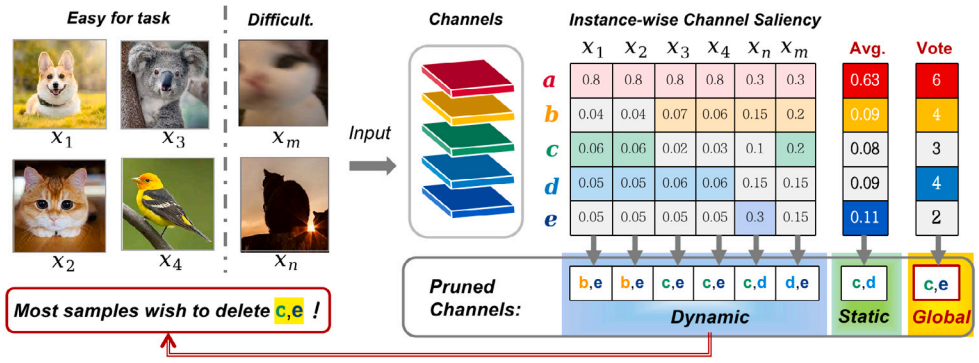


Fig. 1. An illustration of the motivation for GlobalPru combining the advantages of static pruning and dynamic pruning.

This initial step allows GlobalPru to discern global channel attention using the **Majority Vote Mechanism** (Section 3.3). Subsequently, GlobalPru compels all image-specific channel rankings to converge toward this identified global channel attention through the proposed **Learn-to-Rank** regularization method (Section 3.4). To sum up, GlobalPru's optimization objective encompasses three crucial components, including experience risk, channel sparsity regularization, and channel ranking regularization. The optimization target could be formulated as follows:

$$\min_{\Theta} \sum_{i=1}^{N} \sum_{l=1}^{L} \mathscr{L}(f(x_i, \Theta), y_i) + \alpha \left\| \pi^l(x_i) \right\|_1 + \beta \Phi(R_i^l, T^l). \tag{5}$$

Herein, R_i^l denotes the channel ranking derived from the channel saliency score $\pi^l(x_i)$, while T^l represents the target global channel attention. We aim to optimize the channel ranking loss, which involves maximizing the probability that the current channel (importance) orders match the target one. This is achieved by the function $\Phi(R^l, T^l)$ defined in Eqs. (6) and (7), facilitating the key learn-to-rank regularization during training. To be more specific, the probability that the jth object in R_l corresponds to the kth object in the actual ranking T_l is computed by: $P(R_j^l = T_k^l) = \frac{\exp \pi_j^l}{\sum_{i=k}^{c^l} \exp \pi_{T_i^l}^l}$. For example, consider the channel permutation $R^l = \langle c_1, c_2, c_3, c_4 \rangle$ and $T^l = \langle c_2, c_3, c_1, c_4 \rangle$. In this case, the probability that the current 3-rd object in R^l matches the 2-nd object in T^l is calculated as follows: $P(R_3^l = T_2^l) = \frac{\exp \pi_{R_3^l}^l}{\sum_{i=2}^{4} \exp \pi_{T_i^l}^l}$. In summary, we formulate Φ from a Bayesian perspective as follows:

$$\Phi(T^l, R^l) = -\log \prod_{j=1}^{c^l} P(R_j^l | T^l) = -\sum_{j=1}^{c^l} \log \frac{P(T^l | R_j^l) P(R_j^l)}{P(T^l)}. \tag{6}$$

Since T^l is defined as the prior of the channel rank and independent of the current channel ranking, we have $P(T^l) = 1$ and for each $j \in 1, 2, \ldots, c^l$, $P(T^l | R_j^l) = 1$, hence Eq. (6) can be rewritten as:

$$\Phi(T^l, R^l) = -\sum_{j=1}^{c^l} \log P(R_j^l = T_k^l) = -\sum_{j=1}^{c^l} \log \frac{\exp \pi_j^l}{\sum_{i=k}^{c^l} \exp \pi_{T_i^l}^l}. \tag{7}$$

To simulate the impact of model pruning and reduce computational overhead, we use $\hat{\pi}_j^l$ instead of π_j^l in actual optimization. We provide a detailed algorithmic procedure in Algorithm 1.

3.3. Global channel attention

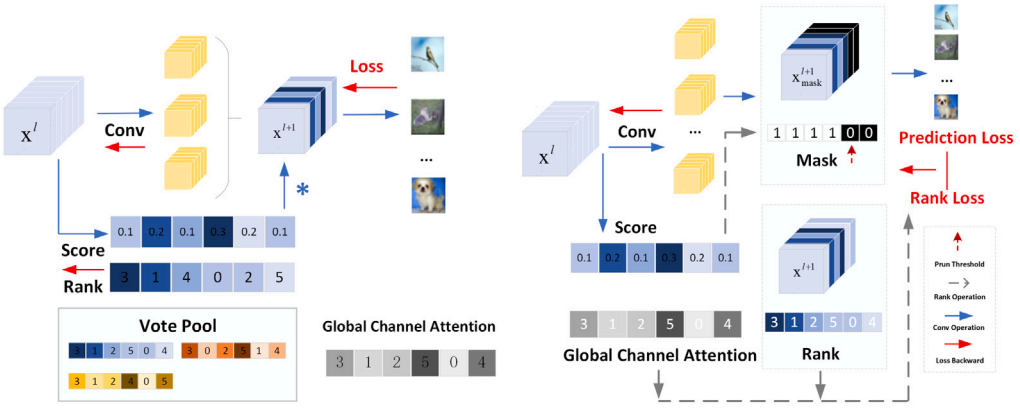
Definition 1. Channel local attention refers to the channel attention of a single input (channel importance ranking), while channel global attention refers to channel attention consistent with most samples for the current data domain.

Algorithm 1 Global Channel Attention-based Sparse Training

Input:
Dataset: $\{(x_i, y_i)\}_{i=1}^N$; The unpruned model F with L convolution layers, layer-wise channel scoring function $S^l, l \in L$, and parameters set Θ (f is a SENet in this work); Cross-entropy loss function: $\mathcal{L}_{ce}$; Pre-training Epochs: e; Ascending order sorting function: Sort; The vote count for channel j on layer l w.r.t input x_i: $vote_{i,j}^l$, initialize $vote_j^l = 0$ for each channel; The function that returns the index of the specified element in the list: Id; The function that returns the indices sorted in descending order SId.
Output:
Sparse Model with Channels Sorted by Importance;
for epoch e **do**
  pre-warm $\{S^l, l \in L\}$ by optimizing $\min \sum_{i=1}^N \mathscr{L}(F(x_i, \Theta), y_i)$;
end for
for each sample $(x_i, y_i)_{i=1}^N$ **do**
  perform one forward pass and compute $\pi^l(x_i) = S^l(x_i)$, $R_i^l = Sort(\pi^l(x_i))$;
  compute the vote count for each channel j, $vote_{i,j}^l = \mathrm{ID}(R_i^l[j])$;
  calculate the total vote count for each channel $vote_j^l += vote_{i,j}^l$;
end for
For each layer l, the channel ranking prior under maximum voting mechanism $T_l = \mathrm{SId}(vote_j^l)$;
while not converged **do**
  for each sample $(x_i, y_i)_{i=1}^N$ **do**
    Learn-to-Rank (channel-wise) regularized model sparse optimization via $\min \sum_{i=1}^N \sum_{l=1}^L \mathscr{L}(f(x_i, \Theta), y_i) + \alpha \left\| \pi^l(x_i) \right\|_1 + \beta \Phi(R_i^l, T^l)$
  end for
end while

Lots of previous works have proven that the importance ranking of channels (Tang et al., 2021) is highly sample-dependent, which means the model redundancy for different samples could be quite different with high probability. In fact, natural images are complex and diverse, composed of a mixture of different intrinsic features such as color, edges, textures, and others. The ratio of these features can vary widely between samples, which means that the model may require different amounts of each feature to effectively capture the information in each sample. This, in turn, means that the model's redundancy for different samples can be very different.

To address this challenge, channel attention has been proposed as a way to adjust the weight of each channel of the output features for each sample, effectively allowing the model to focus on the most important features for each sample. The channel attention mechanism works by learning a set of scalar values that represent the importance of each channel and using these values to rescale the output of the model for each sample. In this way, channel attention can help the model to better



(a) Global channel attention. (b) Learn-to-rank regularization.

Fig. 2. The overview of the training process of global attention-based channel rank network.

approximate the target feature mapping for each sample, and reduce the redundancy of the model in a sample-specific manner. The analysis is formalized in the following corollaries:

Corollary 1. *Conventional channel attention could indicate the quantitative channel redundancy for each sample through the learnable feature scaling factors.*

Corollary 2. *Local channel attention is highly sample-related, which implies that redundant channels are disordered and unpredictable for the whole sample domain.*

On the other hand, the quality of data also plays a crucial role in determining the number of channels needed to obtain satisfactory accuracy. When dealing with good data, a lower number of channels is sufficient and the variance of significant values between channels is substantial. On the other hand, poor data such as blurry images necessitates a higher number of channels and the variance of channel saliency values is reduced. As depicted in Fig. 1, dynamic pruning calculates channel redundancy for each sample $x_1, x_2, \ldots, x_m$ and presents the results in a vertical arrangement ($x_1 - x_m$ columns). Meanwhile, static pruning calculates the average value (The Avg. column) of the channel saliency values for all samples horizontally. Although the majority of the samples ($x_1 - x_4$) want to reserve channel d, it can still be removed in the static method due to the value distribution of a small number of fuzzy data samples (x_n, x_m). As a result, at high pruning ratios, static methods may suffer severe accuracy loss. Despite the fact that dynamic pruning maximizes the accuracy of the pruned model by computing model redundancy for each sample, it is deemed impractical because it requires preserving the entire model at the edge. To combine the benefits of static and dynamic pruning methods while avoiding their drawbacks, we propose a novel approach to measuring global channel attention across all samples in order to maximize the task accuracy of the pruned model for the majority of samples.

In Fig. 2(a), the process of identifying global channel attention is depicted, employing a majority vote mechanism. To provide further details, we commence the procedure by pre-warming a deep neural network equipped with channel attention modules. Specifically, in this study, we utilize SENet (Hu et al., 2017) with squeeze-and-excitation channel attention, subjecting it to a pre-training regimen spanning 30 epochs. Subsequently, during the training phase, we systematically aggregate image-specific local channel attention, effectively forming an 'electoral college' of sorts, which collectively contributes to the determination of global channel attention. For example, Channel 2 is considered the most important in 56% of the images, Channel 1 is deemed the second most important in 73% of the images, and Channel 63 is regarded as the third most important in 46% of the images... After the maximum voting, the channel ranking will be 2, 1, 63, ... This global attention mechanism precedes the subsequent channel importance ranking within the framework of learn-to-rank regularization. In this manner, serving as a static alternative, GlobalPru effectively derives the prevailing channel importance ordering within the current data domain. In fact, employing any channel ranking as a prior for global channel attention is effective. Channel ranking priors under the maximum voting mechanism offer the advantage of accelerating model convergence. Moreover, due to the reduced initial distance from the starting point to the model's convergence point, the maximum voting prior contributes to improving the generalization accuracy of the pruned model (He, Xie, Zhu, & Qin, 2022). The maximum voting mechanism strategic approach enables high-quality pruning akin to dynamic methods, leading to superior model compression rates and mitigated precision degradation resulting from the pruning process.

3.4. Learn-to-rank regularization

The learn-to-rank regularization is an explicit model derived from a Bayesian perspective, which models the joint probability maximization problem as defined in Eq. (6), (7), and EQ. (8). Fig. 2(b) illustrates the process of the proposed Learn-to-Rank regularization method. In each iteration, GlobalPru simultaneously computes the channel **Scores** using a squeeze-and-excitation attention module during the convolution operation. Subsequently, GlobalPru utilizes Strategy 1 (as detailed in Section 3.1) to derive the channel **Mask** and generates a temporary channel **Rank**. The resulting layer output is a masked feature map, which contributes to the final prediction loss. And the **Rank** is employed to calculate the Rank loss as in Eq. (7).

This approach considers the global channel attention as the prior for the channel importance ranking, and the channel ranking loss is formalized as the negative value of the probability that the current channel ranking matches the given prior. This leads to not only empirical risk minimization but also a training of a model distribution toward the most probable probabilistic model that generates the target channel ranking.

However, the vast number of channels in CNNs presents unique challenges when computing the ranking loss for each channel. To overcome this challenge, the authors leverage the strengths of model pruning to address the problem of computational costs in Eq. (7). By only maximizing the probability that the first $(1-p)c^l$ channels in T^l are ranked correctly (where p is the pruning rate and c_l is the number of channels on l_{th} layer), the computation cost can be further reduced. The proposed learn-to-rank regularization can be reformatted to Eq. (8) to

provide a more efficient method for ranking the importance of channels in deep neural networks as:

$$\Phi(T^l, R^l) = -\sum_{j=1}^{(1-p)c^l} \left\{ \log exp(\pi_j^l) - \log \sum_{j=1}^{(1-p)c^l} exp(\pi_j^l) \right\}. \quad (8)$$

4. Experiment

In this section, we aim to evaluate the efficacy of the proposed GlobalPru framework by conducting comprehensive empirical studies on three of the most widely adopted neural network architectures: plain architecture, residual structure, and lightweight depthwise convolution networks. The experiments are designed to demonstrate the effectiveness of both the Global Attention Mechanism and the Learn-To-Rank method, which are two core components of the GlobalPru framework. Through rigorous testing and analysis, we hope to provide insights into how the GlobalPru framework can be used to achieve better performance while reducing computational complexity in neural network models.

4.1. Experimental setup

4.1.1. Datasets

The proposed GlobalPru is evaluated on several widely-used image datasets in the field of model pruning, including ImageNet, CIFAR-10, CIFAR-100, and SVHN. These datasets have been widely adopted in previous works for model pruning and serve as a good benchmark to evaluate the effectiveness of GlobalPru.

ImageNet is a large-scale image recognition dataset consisting of over 14 million images with over 20,000 object categories. The images in the dataset have varying sizes, but most commonly they are resized to 256 × 256 or 224 × 224 pixels. It is widely used for training and evaluating computer vision models and has played a crucial role in advancing the field of deep learning. The dataset is commonly used as a benchmark for image classification tasks. CIFAR-10 and CIFAR-100 are two widely used datasets in computer vision and machine learning, which consist of color images of 32 × 32 pixels in size, each belonging to 10 and 100 classes, respectively. SVHN is a real-world image dataset that contains over 600,000 digit images of street view house numbers. The diversity of these datasets makes them a good choice for evaluating the broad applicability of GlobalPru.

4.1.2. Models

In order to thoroughly evaluate the performance of the proposed GlobalPru method, we have conducted experiments on three of the most widely used and well-known neural network architectures: plain architecture - VGG-16 and VGG16-BN, residual structure – ResNet with various depths, and lightweight depthwise convolution – MobileNetV2. These three models represent a comprehensive cross-section of the most popular convolutional networks currently in use. By testing these diverse models, we aim to demonstrate the broad applicability and effectiveness of GlobalPru in pruning redundant parameters while preserving model performance. Each of these models is carefully selected to showcase the potential of GlobalPru on different types of network structures and to showcase its robustness against various model complexities.

4.1.3. Metrics

Most model pruning works in the field only evaluate the accuracy of the pruned models on the image classification task. However, the robustness of the pruned networks is often ignored. To provide a comprehensive evaluation of the performance of our GlobalPru method, we not only evaluate the accuracy of the pruned networks on the image classification task but also evaluate the robustness of the pruned networks against adversarial perturbations. This is achieved by conducting an adversarial sample detection task, which is an important and challenging task in the field of deep learning. By demonstrating the robustness of the pruned networks on the adversarial samples detection task, we aim to further demonstrate the applicability of our GlobalPru method in various scenarios.

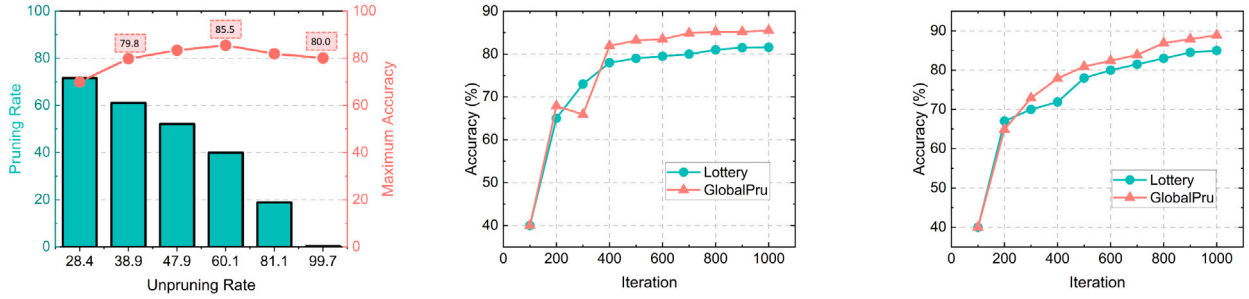
4.1.4. Implementation details

Standard data argumentation RandomSizedCrop and RandomHorizontalFlip are used in all datasets we used. The coefficient α to regulate the channel saliency score is set as 0.0001 in our work. Another coefficient β to balance the weight of the channel rank loss is also set as 0.0001 empirically. The stochastic gradient descent (SGD) is used for all training processes. Three pruning modes are tested in GlobalPru, named “Fixed”, “Mixed”, and “Compared” respectively. Under the fixed mode, all layers use the same pruning rate with the Lottery Ticket of model pruning which corresponds to the highest test accuracy (Frankle & Carbin, 2019). In order to investigate a reasonable pruning rate range, we test the proposed GlobalPru with different pruning rates, which is based on the classic three-stage “train-prune-finetune” pruning paradigm. We present an illustrative example using ResNet-18 applied to the CIFAR-10 dataset, and the corresponding results are visualized in Fig. 3. It can be seen that pruning rates within the range of $[38.9\%, 100.0\%]$ yield higher test accuracy compared to the unpruned model, with the optimal pruning rate resulting in a peak test accuracy of 61.1%. Based on the accuracy-pruning rate results, we investigate a pruning rate interval of $[40.1\% - 61.1\%]$ for convolutional layers. The lower bound represents the pruning rate at which peak generalization performance occurs, while the upper bound signifies the pruning rate at which decreased generalization accuracy begins.

Ultra-low pruning rate may lead to a significant decrease in the model’s generalization ability under fixed mode. To solve this problem, we propose to perform mixed pruning mode, where more sensitive layers maintain a lower sparsity. Dong et al. (2020) has theoretically proven that the average Hessian trace is the right sensitivity for model perturbation (quantization, pruning, low-rank decomposition...). Specifically, we use the matrix-free Hutchinson algorithm on the pre-trained model (these pre-trained parameters could be easily found in pytorch) to efficiently compute the layer-wise average Hessian matrix. The larger the Hessian trace value, the lower the pruning rate. Further insights into Hessian computation can be found in Avron and Toledo (2011), Bai, Fahey, and Golub (1996). Subsequently, with a predetermined budget, GlobalPru enables adaptive compression ratio assignment among all compressible components through a Pareto frontier-based method, as outlined in Dong et al. (2020). The compared mode is an extension of the mixed mode, where it fixes the overall pruning rate budget and then selectively allocates pruning rates to different layers. This mode was introduced to ensure a fair comparison of pruned model accuracy against state-of-the-art (SOTA) models at the same sparsity benchmark. All of the experiments are conducted on NVIDIA GeForce GPUs.

4.2. Comparison on plain architecture

We prune the neural networks of plain architecture (in this work, we use the most widely-used VGG-16 serious) on CIFAR-10 and ImageNet, and comprehensively compare with other state-of-the-art methods. For static pruning, we compare our method with ThiNet (Luo et al., 2017), L1-norm sparse (Li et al., 2017), CP (He et al., 2017b), NS (Liu et al., 2017b), and AGSPRL (Wei et al., 2022). For dynamic pruning methods, we compare with RNP (Lin et al., 2017), DNP (Wang et al., 2020), LIWS (Liu et al., 2019), FBS (Gao et al., 2019), and FTWT (Elkerdawy et al., 2022). We record the accuracy of the baseline and the pruned model, accuracy drop, FLOPs reduction, and parameter reduction for each pruning method.



(a) Lottery ticket search. (b) Pruning rate is 40.1%. (c) Pruning rate is 61.1%.

Fig. 3. Comparison of pruning performance under different pruning rates on ResNet-20 and CIFAR-10. (a) depicts the process of using the lottery hypothesis (Frankle & Carbin, 2019) to select the pruning rate range of GlobalPru. (b) and (c) represent the accuracy comparison between our method and lottery at the minimum and maximum pruning rates selected by Globalpru, where the lottery represents traditional weight-based three-stage pruning.

Table 2
Comparison of pruning performance on VGG16-BN architecture and CIFAR-10 dataset. The proposed method is compared with SOTAs including both static methods and sample-wise dynamic methods. '↓FLOPs (%)' and '↓Para. (%)' are the reduced percentage of FLOPs and the reduced percentage of Parameters, respectively. 'F' and 'M' represent the fixed pruning mode and mixed pruning mode, respectively. '*' denotes the proposed novel pruning paradigm. '–' represents incomparable items, typically due to different baseline accuracies reported by the comparison methods.

Method	Dynamic?	↓FLOPs (%)	↓Para. (%)	Accuracy (%)	↓ Acc. (%)
Baseline		0.00	0.00	93.5	0.00
DNP (Wang et al., 2020)	✓	50.40	–	93.45	0.05
LIWS (Liu et al., 2019)	✓	46.90	–	–	0.10
ThiNet (Luo, Wu, & Lin, 2017)	✗	50.00	47.92	93.36	0.14
L1-norm (Li, Kadav, Durdanovic, Samet, & Graf, 2017)	✗	34.00	–	93.00	0.50
CP (He, Zhang, & Sun, 2017b)	✗	50.00	47.92	93.18	0.32
NS (Liu, Li, Shen, Huang, Yan, & Zhang, 2017b)	✗	51.00	70.00	93.31	0.19
AGSPRL (Wei, Wang, Hua, Sun, & Zhao, 2022)	✗	50.80	73.84	92.83	0.67
RNP (Lin, Rao, Lu, & Zhou, 2017)	✓	50.00	–	92.65	0.85
FBS (Gao, Zhao, Dudziak, Mullins, & Xu, 2019)	✓	50.00	–	93.03	0.47
FTWT (r=0.92) (Elkerdawy, Elhoushi, Zhang, & Ray, 2022)	✓	56.00	–	–	0.09
GlobalPru (F)	*	55.00	71.34	93.29	0.21
GlobalPru (M)	*	56.70	80.07	93.49	0.01

Table 3
Comparison of pruning performance on Plain architecture: VGG-16 architecture and ImageNet dataset.

Method	Dynamic?	↓FLOPs (%)	Accuracy (%)	↓Acc (%)	
Baseline		0.00	0.00	89.90	0.00
L1-norm (Li et al., 2017)	✗	75.00	–	86.54	3.36
CP (He et al., 2017b)	✗	80.00	–	88.10	1.80
NS (Liu et al., 2017b)	✗	75.00	82.50	84.72	5.18
AMC (He, Lin, et al., 2018)	✗	80.00	–	88.50	1.40
RNP (Lin et al., 2017)	✓	80.00	–	86.32	3.58
LIWS (Wang et al., 2020)	✓	81.20	–	88.57	1.33
GlobalPru (M)	*	82.30	89.59	88.90	1.00

The comparison results of VGG16-BN pruning on CIFAR-10 are summarized in Table 2. It can be seen that when using the fixed pruning rate, our method achieves 55.2% FLOPs decrease and 71.34% parameters reduction with a negligible accuracy drop of −0.21%. When using the empirically mixed pruning rate for each layer, GlobalPru could further improve the FLOPs and parameters reduction to 56.7% and 80.07%, respectively, while with less accuracy degradation (−0.01%). To be specific, compared to the static SOTAs, GlobalPru wins a significant advantage in pruning rate and only has a slight accuracy drop. When compared to the pioneering dynamic methods, GlobalPru achieves less accuracy degradation than the most popular (Liu et al., 2019) and outperforms (Gao et al., 2019; Lin et al., 2017) in both compression rate and accuracy. The work that aligns with our method in terms of FLOPs reduction is FTWT (Elkerdawy et al., 2022), which achieved a substantial 56.0% reduction. However, its pruned model exhibits an unsatisfied 0.09% accuracy drop, which is 0.08% higher than that of our method.

The pruning results of VGG-16 on ImageNet are presented in Table 3. It can be seen that GlobalPru achieves the highest pruning ratio of 82.30% among all SOTAs while maintaining a competitive accuracy drop. In summary, GlobalPru excavates more model redundancy than SOTAs on plain network architecture while maintaining competitive model performance. In summary, the results demonstrate that GlobalPru could work efficiently on plain network architecture.

4.3. Comparison on residual architecture

We further investigate the effectiveness of GlobalPru on neural networks of residual architecture and compare the results against state-of-the-art static and dynamic methods. In the realm of static techniques, GlobalPru is pitted against SFP (He, Kang, et al., 2018), FPGM (He et al., 2019), DSA (Ning et al., 2020), Hinge (Li, Gu, et al., 2020), DHP (Li, Gu, et al., 2020), DLRFC (He, Qian, et al., 2022), and DCFF (Lin et al., 2023). In the realm of dynamic pruning methods, we conduct comparisons with Maninp (Tang et al., 2021), FBS (Gao et al., 2019), and DSP (Park et al., 2023). These experiments are performed under two settings including ResNet-20 on CIFAR-10 and ResNet-50 on ImageNet.

Table 4
Comparison of pruning performance on Residual architecture: ResNet-20 architecture and CIFAR-10 dataset. 'C' represents the compared pruning mode.

Method	Dynamic?	↓FLOPs (%)	Accuracy (%)	↓Acc (%)
Baseline		0.00	91.11	0.00
SFP (He, Kang, Dong, Fu, & Yang, 2018)	✗	42.20	89.18	1.39
FPGM (He, Liu, Wang, Hu, & Yang, 2019)	✗	54.00	88.79	1.78
DSA (Ning et al., 2020)	✗	50.30	89.73	0.84
Hinge (Li, Gu, Mayer, Gool, & Timofte, 2020)	✗	45.50	90.10	0.38
DHP (Li, Gu, Zhang, Van Gool, & Timofte, 2020)	✗	51.80	89.89	0.68
Maninp (Tang et al., 2021)	✓	54.20	90.40	0.17
FBS (Gao et al., 2019)	✓	53.10	89.32	1.25
DSP (Park, Kim, Kim, Choi, & Lee, 2023)	✓	55.76	–	−0.01
GlobalPru (F)	*	60.10	90.7	0.14
GlobalPru (M)	*	60.80	91.05	0.06
GlobalPru (C)	*	55.76	91.83	−0.72

Table 5
Comparison of pruning performance on Residual architecture: ResNet-50 architecture on ImageNet dataset.

Method	Dynamic?	↓FLOPs (%)	↓Para.(%)	Accuracy (%)	↓Acc (%)
Baseline		0.00	0.00	92.93	0.00
L1-norm (Li et al., 2017)	✗	50.00	–	74.39	18.54
CP (He et al., 2017b)	✗	50.00	–	90.80	2.13
ThiNet (Liu et al., 2017b)	✗	37.00	48.54	91.84	1.09
SFP (Liu et al., 2017b)	✗	41.80	–	92.06	0.87
LIWS (Wang et al., 2020)	✓	51.30	–	92.08	0.85
DLRFC (He, Qian, et al., 2022)	✗	54.00	40.00	92.64	0.29
DCFF (Lin, Chen, Chao, & Ji, 2023)	✗	76.01	71.00	90.41	2.52
GlobalPru (C)	*	54.00	68.90	92.77	0.20
GlobalPru (C)	*	76.00	88.31	91.50	1.47
GlobalPru (M)	*	51.40	63.23	92.81	0.12

The comparison results on ResNet-20 and CIFAR-10 are shown in Table 4. It can be seen that GlobalPru achieves the highest compression ratio (60.80%), with the most negligible accuracy drop (0.06%) among the popular SOTAs. We observe a slight improvement in the pruned model's generalization performance with the DSP method, but the sparsity is approximately 5% lower than GlobalPru(M). To make an equitable comparison with DSP, we set GlobalPru's pruning rate (fixed model) to approximate the FLOPs reduction reported by DSP. The results show that at lower pruning rates, GlobalPru also achieves a slight accuracy improvement (0.72%). We attribute this phenomenon to the pruning lottery effect as illustrated in Fig. 3.

The comparison results for ResNet-50 on the ImageNet dataset are presented in Table 5. On average, GlobalPru achieves superior performance with the highest FLOPs and parameters reduction at 51.40% and 63.23%, respectively, and only a minimal accuracy drop of −0.12% among all the methods. The state-of-the-art DLRFC and DCFF methods report FLOPs reductions of 54.0% and 76%, respectively. While DLFRC exhibits a slightly lower accuracy drop of 0.17% compared to our method, DCFF, despite its remarkable FLOPs reduction, incurs an unsatisfactory accuracy drop of 2.52%. To facilitate a more direct comparison, we set GlobalPru's sparsity to achieve an FLOPs reduction similar to DLRFC and DCFF respectively, and compare the pruned model's accuracy at an equivalent level. Notably, at a 54.0% FLOPs reduction, our method outperforms DLRFC by a significant margin of 0.09%; at 76.0% FLOPs reduction, our method outperforms DCFF by 1.05%. The results reaffirm the necessity of aggregating channel attention across different instances. In summary, the results demonstrate that GlobalPru could work efficiently on residual network architecture.

4.4. Comparison on DepthWise architecture

Another popular model architecture using "Depthwise Separable Convolution" is also included in our experiment to cover the ultra-modern compact architecture. We assess this line of research using the widely adopted MobileNetV2 on CIFAR-10 and ImageNet. We comprehensively compare GlobalPru with other state-of-the-art methods with respect to the accuracy drop, FLOPs reduction, and parameter reduction.

The comparison results on CIFAR-10 are shown in Table 6, we compare our method against DCP (Zhuang et al., 2018), WM (the width-shrinking variant of DCP), NPPM (Gao et al., 2021), and GMP (Belay, 2022). It can be seen that, among all the methods, GlobalPru consistently outperforms with an average FLOPs reduction of 60.14%, parameters reduction of 89.20%, and a minimal accuracy loss of only 0.55%. What sets it apart is that the latest work reported an unprecedented sparsity rate of up to 90%. Unfortunately, it suffered from a significant accuracy drop. Exploring ultra-compact models with minimal precision loss could be one of our future directions.

The comparison results on ImageNet are summarized in Table 7, the GlobalPru is compared with DPFPS (Ruan et al., 2021), CC (Li et al., 2021), and ManiDP (Tang et al., 2021). Among all the competitors, GlobalPru achieves superior performance and is the only method without accuracy loss. It reaches the highest FLOPs reduction of 39.29% with a slight increase in accuracy of 0.1%. In summary, the performance of our method on lightweight DepthWise architecture demonstrates that GlobalPru will likely be one of the key contributors to the direction of ultra-efficient model compression.

4.5. Generalization verification

A good pruning method must have good generalization and robustness in addition to model accuracy, which is often overlooked in previous network pruning works. To demonstrate that GlobalPru has these advantageous properties, we put it through its paces on variational visual tasks with adversarial samples.

As shown in Table 8, we verify the robustness of our method by observing the performance of GlobalPru when faced with adversarial perturbations. It can be seen that GlobalPru exhibits extraordinary perturbation resistance, even surpassing some typical adversarial sample detection methods. We speculate that this is because the model redundancy removed by GlobalPru is selected by a majority vote of all samples, thereby weakening the effect of small image disturbances

Table 6
Comparison of pruning performance on DepthWise architecture: MobileNetV2 architecture and CIFAR-10 dataset. WM is the width-shrinking version of the original DCP method.

Method	Dynamic?	↓FLOPs (%)	↓Para. (%)	Acc. (%)	↓Acc (%)
Baseline		0.00	0.00	97.85	0.00
WM (Zhuang et al., 2018)	✗	26.47	76.33	94.17	3.68
DCP (Zhuang et al., 2018)	✗	26.47	76.33	94.69	3.16
NPPM (Gao, Huang, Cai, & Huang, 2021)	✗	47.00	–	94.75	3.10
GMP (Belay, 2022)	✗	88.2	–	82.00	15.85
GlobalPru (M)	*	60.14	89.20	97.30	0.55

Table 7
Comparison of pruning performance on DepthWise architecture: MobileNetV2 architecture and ImageNet. The FLOPs and parameters reduction originates from Ruan, Liu, Li, Yuan, and Hu (2021).

Method	Dynamic?	↓FLOPs (%)	↓Para. (%)	Acc. (%)	↓Acc (%)
Baseline		0.00	0.00	71.80	0.00
DPFPS (Ruan et al., 2021)	✗	25.00	–	71.40	0.40
CC (Li et al., 2021)	✗	29.00	–	71.19	0.61
ManiDP (Tang et al., 2021)	✗	30.00	–	71.50	0.3
GlobalPru (M)	*	39.29	33.13	71.90	−0.10

Table 8
Comparison of AUROC (%) of GlobalPru with typical adversarial detection methods on Fast Gradient Sign Method (FGSM)-based CIFAR-100 and SVHN datasets.

AUROC (%) FGSM	Non prune		Prune	
	MAHA	FBS	DPIC	**GlobalPru**
SVHN	99.63	99.95	99.96	**100.00**
CIFAR-100	99.77	100.00	100.00	**100.00**

and making GlobalPru more robust than the previous pruning criterion only guided by task loss.

5. Conclusion

To sum up, we propose a novel pruning method, i.e., the Channel Attention-based Learn-to-Rank Network, based on the shortcomings of current dynamic pruning methods, such as the need to save the complete model locally and repeat the forward computation. Our approach first explores the channel saliency rank of each sample and then selects the most suitable channel rank supported by all current inputs through a majority voting mechanism. We define this channel rank as global channel attention. Next, we obtain a Channel Rank Network by proposing an efficient channel sorting algorithm to incorporate the knowledge of global channel attention into the training of the model, so as to quickly give an appropriate pruning response when new samples or sparse requirements arrive. While obtaining the advantages of dynamic pruning, our method avoids the defects of the original dynamic pruning method and achieves better pruning performance than most state-of-the-art methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: National Key R&D Program of China No. 2020AAA0108800, NSFC under Grant 62172326 and 62137002, the MOE Innovation Research Team No. IRT17R86, China University Innovation Fund NO. 2021FNA04003, and the Project of China Knowledge Centre for Engineering Science and Technology.

References

Avron, H., & Toledo, S. (2011). Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM*, *58*(2), 1–34.
Bai, Z., Fahey, G., & Golub, G. (1996). Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, *74*(1–2), 71–89.
Belay, K. (2022). Gradient and mangitude based pruning for sparse deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 11* (pp. 13126–13127).
Chen, X., Zhu, J., Jiang, J., & Tsui, C. (2020). Tight compression: Compressing CNN model tightly through unstructured pruning and simulated annealing based permutation. In *57th ACM/IEEE design automation conference* (pp. 1–6). IEEE.
Dong, X., Huang, J., Yang, Y., & Yan, S. (2017). More is less: A more complicated network with less inference complexity. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5840–5848).
Dong, Z., Yao, Z., Arfeen, D., Gholami, A., Mahoney, M. W., & Keutzer, K. (2020). HAWQ-V2: hessian aware trace-weighted quantization of neural networks. In *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020*.
Elkerdawy, S., Elhoushi, M., Zhang, H., & Ray, N. (2022). Fire together wire together: A dynamic pruning approach with self-supervised mask prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12454–12463).
Elsen, E., Dukhan, M., Gale, T., & Simonyan, K. (2020). Fast sparse ConvNets. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 14617–14626). Computer Vision Foundation / IEEE.
Frankle, J., & Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th international conference on learning representations*. OpenReview.net.
Gao, S., Huang, F., Cai, W., & Huang, H. (2021). Network pruning via performance maximization. In *IEEE conference on computer vision and pattern recognition* (pp. 9270–9280). Computer Vision Foundation / IEEE.
Gao, X., Zhao, Y., Dudziak, Ł., Mullins, R., & Xu, C.-z. (2018). Dynamic channel pruning: Feature boosting and suppression. arXiv preprint arXiv:1810.05331.
Gao, X., Zhao, Y., Dudziak, L., Mullins, R. D., & Xu, C. (2019). Dynamic channel pruning: Feature boosting and suppression. In *7th international conference on learning representations*. OpenReview.net.
He, Y., Kang, G., Dong, X., Fu, Y., & Yang, Y. (2018). Soft filter pruning for accelerating deep convolutional neural networks. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence* (pp. 2234–2240). ijcai.org.
He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., & Han, S. (2018). Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision* (pp. 784–800).

He, Y., Liu, P., Wang, Z., Hu, Z., & Yang, Y. (2019). Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4340–4349).

He, Z., Qian, Y., Wang, Y., Wang, B., Guan, X., Gu, Z., et al. (2022). Filter pruning via feature discrimination in deep neural networks. In *European conference on computer vision* (pp. 245–261). Springer.

He, Z., Xie, Z., Zhu, Q., & Qin, Z. (2022). Sparse double descent: Where network pruning aggravates overfitting. In *International conference on machine learning* (pp. 8635–8659). PMLR.

He, Y., Zhang, X., & Sun, J. (2017a). Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1389–1397).

He, Y., Zhang, X., & Sun, J. (2017b). Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1389–1397).

Hu, J., Shen, L., & Sun, G. (2017). Squeeze-and-excitation networks. CoRR, abs/1709.01507.

Hua, W., Zhou, Y., De Sa, C. M., Zhang, Z., & Suh, G. E. (2019). Channel gating neural networks. *Advances in Neural Information Processing Systems*, *32*.

Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with Gumbel-softmax. In *International conference on learning representations*. URL https://openreview.net/forum?id=rkE3y85ee.

Khan, R. A., Fu, M., Brent, B., Luo, Y., & Wu, F.-X. (2023). A multi-modal deep neural network for multi-class liver cancer diagnosis. *Neural Networks*, *165*, 553–561.

Kuang, S., Woodruff, H. C., Granzier, R., van Nijnatten, T. J., Lobbes, M. B., Smidt, M. L., et al. (2023). MSCDA: Multi-level semantic-guided contrast improves unsupervised domain adaptation for breast MRI segmentation in small datasets. *Neural Networks*, *165*, 119–134.

Kwon, S. J., Lee, D., Kim, B., Kapoor, P., Park, B., & Wei, G. (2020). Structured compression by weight encryption for unstructured pruning and quantization. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 1906–1915). Computer Vision Foundation / IEEE.

Li, Y., Gu, S., Mayer, C., Gool, L. V., & Timofte, R. (2020). Group sparsity: The hinge between filter pruning and decomposition for network compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8018–8027).

Li, Y., Gu, S., Zhang, K., Van Gool, L., & Timofte, R. (2020). Dhp: Differentiable meta pruning via hypernetworks. In *Computer Vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, Part VIII 16* (pp. 608–624). Springer.

Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017). Pruning filters for efficient ConvNets. In *5th international conference on learning representations, ICLR 2017, Toulon, France, April 24-26, 2017, conference track proceedings*. OpenReview.net.

Li, Y., Lin, S., Liu, J., Ye, Q., Wang, M., Chao, F., et al. (2021). Towards compact CNNs via collaborative compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6438–6447).

Li, X., Wang, W., Hu, X., & Yang, J. (2019). Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 510–519).

Liebenwein, L., Baykal, C., Lang, H., Feldman, D., & Rus, D. (2019). Provable filter pruning for efficient neural networks. In *International conference on learning representations*.

Lin, M., Chen, B., Chao, F., & Ji, R. (2023). Training compact CNNs for image classification using dynamic-coded filter fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lin, J., Rao, Y., Lu, J., & Zhou, J. (2017). Runtime neural pruning. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 2178–2188).

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017a). Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision* (pp. 2736–2744).

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017b). Learning efficient convolutional networks through network slimming. In *IEEE international conference on computer vision* (pp. 2755–2763). IEEE Computer Society.

Liu, C., Wang, Y., Han, K., Xu, C., & Xu, C. (2019). Learning instance-wise sparsity for accelerating deep models. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 3001–3007). ijcai.org.

Luo, J.-H., Wu, J., & Lin, W. (2017). Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision* (pp. 5058–5066).

Molchanov, P., Mallya, A., Tyree, S., Frosio, I., & Kautz, J. (2019). Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11264–11272).

Ning, X., Zhao, T., Li, W., Lei, P., Wang, Y., & Yang, H. (2020). Dsa: More efficient budgeted pruning via differentiable sparsity allocation. In *Computer Vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, Part III 16* (pp. 592–607). Springer.

Park, J., Kim, Y., Kim, J., Choi, J., & Lee, S. (2023). Dynamic structure pruning for compressing CNNs. In *Thirty-seventh AAAI conference on artificial intelligence, AAAI 2023, thirty-fifth conference on innovative applications of artificial intelligence, IAAI 2023, thirteenth symposium on educational advances in artificial intelligence* (pp. 9408–9416). AAAI Press.

Rao, Y., Lu, J., Lin, J., & Zhou, J. (2018). Runtime network routing for efficient image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(10), 2291–2304.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, *28*.

Ruan, X., Liu, Y., Li, B., Yuan, C., & Hu, W. (2021). DPFPS: dynamic and progressive filter pruning for compressing convolutional neural networks from scratch. In *Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence* (pp. 2495–2503). AAAI Press.

Tang, Y., Wang, Y., Xu, Y., Deng, Y., Xu, C., Tao, D., et al. (2021). Manifold regularized dynamic network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5018–5028).

Tang, Y., Wang, Y., Xu, Y., Tao, D., Xu, C., Xu, C., et al. (2020). Scop: Scientific control for reliable neural network pruning. *Advances in Neural Information Processing Systems*, *33*, 10936–10947.

Wang, Y., Wang, J., Zhang, W., Zhan, Y., Guo, S., Zheng, Q., et al. (2022). A survey on deploying mobile deep learning applications: A systemic and technical perspective. *Digital Communications and Networks*, *8*(1), 1–17.

Wang, Y., Zhang, X., Hu, X., Zhang, B., & Su, H. (2020). Dynamic network pruning with interpretable layerwise channel selection. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 04* (pp. 6299–6306).

Wei, X., Du, W., Wan, H., & Min, W. (2023). Feature distribution fitting with direction-driven weighting for few-shot images classification. In *Proceedings of the thirty-seventh AAAI conference on artificial intelligence* (pp. 10315–10323).

Wei, H., Wang, Z., Hua, G., Sun, J., & Zhao, Y. (2022). Automatic group-based structured pruning for deep convolutional networks. *IEEE Access*, *10*, 128824–128834. http://dx.doi.org/10.1109/ACCESS.2022.3227619.

Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016). Learning structured sparsity in deep neural networks. *Advances in Neural Information Processing Systems*, *29*.

Zhang, Y., Lin, M., Lin, Z., Luo, Y., Li, K., Chao, F., et al. (2022). Learning best combination for efficient N: M sparsity. In *NeurIPS*.

Zhang, Y., Luo, Y., Lin, M., Zhong, Y., Xie, J., Chao, F., et al. (2023). Bi-directional masks for efficient N: m sparse training. In *Proceedings of machine learning research: vol. 202, International conference on machine learning* (pp. 41488–41497). PMLR.

Zhang, X., Xu, H., Mo, H., Tan, J., Yang, C., Wang, L., et al. (2021). DCNAS: densely connected neural architecture search for semantic image segmentation. In *IEEE conference on computer vision and pattern recognition* (pp. 13956–13967). Computer Vision Foundation / IEEE.

Zhou, A., Ma, Y., Zhu, J., Liu, J., Zhang, Z., Yuan, K., et al. (2021). Learning N: m fine-grained structured sparse neural networks from scratch. In *9th international conference on learning representations*. OpenReview.net.

Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Guo, Y., Wu, Q., et al. (2018). Discrimination-aware channel pruning for deep neural networks. In *Advances in neural information processing systems 31: Annual conference on neural information processing systems 2018* (pp. 883–894).

Zhuang, T., Zhang, Z., Huang, Y., Zeng, X., Shuang, K., & Li, X. (2020). Neuron-level structured pruning using polarization regularizer. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems, vol. 33* (pp. 9865–9877). Curran Associates, Inc.