2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Statistical Feature-Driven Regularization for Structured Model Pruning

Jielei Wang[1,3[0000-0003-2882-7053]], Dongnan Liu[1,2[0009−0006−5197−6232]], Heng Yin[1,2[0009−0001−0084−0427]], Kexin Li[1,3[0000−0003−3511−2582]], Guangchun Luo[1[0000−0001−7330−2139]], Guoming Lu[1,3[0000−0001−7477−5800]]

[1] The Institute of Intelligent Computing, University of Electronic Science and Technology of China, Chengdu, China
[2] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[3] Ubiquitous Intelligence and Trusted Services Key Laboratory of Sichuan Province
jieleiwang_uestc@163.com

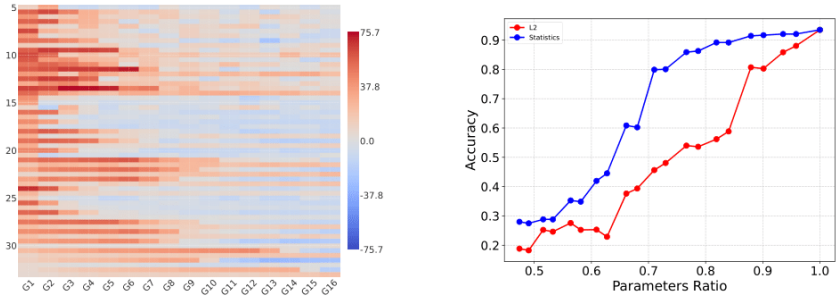
Abstract. Structured pruning is a highly effective model compression technique that balances accuracy and acceleration, making it widely adopted in the field of convolutional neural networks. Traditional pruning methods relying on magnitude-based criteria exhibit limitations in distinguishing critical channels because of narrow parameter distributions in sparse models. Building on this phenomenon, we propose a statistical feature-driven structured pruning framework that integrates dependency-aware group regularization. By incorporating a dependency graph to model inter-layer relationships and leveraging both the mean and variance of channel parameters, we design a dynamic regularization term to reduce both the norm and variance of channels, encouraging uniform shrinkage. Our approach has been validated through experiments across diverse datasets and model architectures, achieving only a 0.71% accuracy drop on ImageNet compared to the baseline model under similar FLOPs reduction ratios.

Keywords: Structured Pruning, Convolutional Neural Networks, Regularization, Statistical Feature

1 Introduction

With the widespread application of deep learning across various fields, the scale and complexity of neural network models have grown rapidly, resulting in a vast number of parameters, high computational resource demands, and significant storage space requirements. This limits the deployment of deep learning models on resource constrained devices, such as mobile devices and embedded systems, but also increases the energy consumption and time costs during training and inference processes. Therefore, various model compression techniques such as model distillation, quantization, and pruning have been proposed and extensively studied. Among these, structured pruning stands out as a primary technique, which not only greatly decreases the model's computational load and storage requirements but also aligns better with the architecture of

hardware accelerators. For this reason, this article concentrates on the domain of structured pruning [2, 6].



(a) Channels Sparsity Difference

(b) Accuracy Decay Comparison After Pruning (Top-1 Accuracy: 93.5%)

Fig. 1. Regularized vs Non-Regularized ResNet-56 on CIFAR-10.

Within structured pruning methods, regularization serves as a crucial technique to encourage model parameters to shrink towards zero, thereby facilitating the removal of redundant structures. Common regularization methods include l_1 regularization, l_2 regularization, and Elastic Net regularization [3, 17, 20]. l_1 regularization adds a penalty term based on the absolute values of the parameters, promoting sparsity; l_2 regularization penalizes the sum of the squares of the parameters, preventing them from becoming too large; Elastic Net regularization combines the advantages of both l_1 and l_2, encouraging sparsity while also controlling parameter magnitudes. However, these regularization methods typically operate under the assumption that channels with smaller norms are less informative [12]. In practice, after applying regularization, some channels may have overall small norms but exhibit high variance in their parameters, indicating that these channels still contain significant information in certain aspects.

To validate our hypothesis, we quantified the sparsity level of channel groups (containing 1 or 2 channels) between the regularized and non-regularized models based on their l_2 magnitudes. Channel groups were sorted in ascending order of their l_2 values for comparison. As shown in the left image of Figure 1, the intermediate layers of the regularized model exhibit significantly higher sparsity compared to the non-regularized counterpart, with nearly half of the parameters approaching zero.

Furthermore, the right image of Figure 1 compares the accuracy decay of the model after pruning channels using traditional norm-based criteria versus pruning channels using our evaluation standard. We found that pruning based on the l_2 results in a greater loss of model accuracy. This phenomenon suggests that solely relying on parameter norms for pruning may lead to the unintended removal of informative channels.

Specifically, the conventional "smaller-norm-less-informative" assumption suffers from two critical limitations: Traditional metrics often rely on parameter magnitude, leads to misidentifying channels with high mean but low variance as informative, while overlooking those with moderate means but high variance that may encode dynamic features. Additionally, traditional methods apply uniform importance across all layers,

failing to account for the different layer sensitivity in deep networks, where shallow layers exhibit higher parametric variance for primitive feature extraction, while deeper layers demand stable, low-variance representations to preserve semantic fidelity [4, 5]. Motivated by these limitations, we propose Statistical Feature-Driven Regularization, a dual-constraint framework that simultaneously suppresses channel parameter magnitudes and stabilizes their distributions through adaptive regularization. The detailed algorithm flow is outlined in Algorithm 1. By explicitly controlling both first-order magnitude statistics and second-order distribution variance, it provides systematic yet gentle guidance for

Building upon the proposed regularization method based on parameter statistical features, the main contributions of our paper are as follows:

- We propose a regularization method based on model statistical features, which uniformly regularizes redundant channels, effectively reducing model redundancy and enhancing model compression performance.
- By combining mean and variance to assess channel importance, our approach enhances pruning effectiveness through dual-dimensional evaluation, maintaining model performance and alleviating the limitations of traditional methods that rely solely on norm magnitudes.

2 Related Work

The rise of structured pruning in neural network optimization has garnered significant attention in recent years. Unlike unstructured pruning, which removes individual weights based on their magnitudes, structured pruning focuses on removing entire channels, filters, or layers, leading to a more efficient and hardware-friendly model (4). This approach is particularly attractive for real-world deployment, as it often results in improved computational efficiency and reduced memory consumption. Early methods, such as magnitude-based pruning, laid the groundwork for structured pruning techniques, but they were often limited by their inability to account for the intricate dependencies and interactions among the channels [4, 7, 9, 11].

Recent advancements in channel pruning have moved beyond simplistic magnitude-based criteria by introducing sophisticated metrics to assess filter importance. Layer-adaptive sparsity for magnitude-based pruning designed a layer-adaptive global pruning scheme based on a relaxed output distortion minimization criterion [8], they extend single-layer magnitude-based pruning to the full model, and mathematically derive which channels have the minimal impact on the overall model distortion. In contrast, HRank [10] evaluates filters via the rank of their output feature maps, arguing that high-rank features retain richer information and thus prioritizing their corresponding filters. Meanwhile, the White-Box channel pruning method [19] introduces class-wise masks to integrate category label information into the model regularization training process, thereby quantifying the contribution levels of individual channels to different classes, which enables straightforward pruning.

Further diversifying evaluation criteria, entropy induced pruning [13] leverages information-theoretic principles, measuring parameter redundancy using the information

entropy of normalized eigenvalues, where lower entropy indicates higher redundancy. Specifically project the convolutional layer weight matrices into a low-rank space via SVD decomposition. REPrune [14] jointly optimize channel pruning through clustering and the maximum coverage problem (MCP). Specifically, the method first performs Ward's hierarchical clustering on kernels within each input channel to form clusters based on layer-specific thresholds, with each cluster representing a set of similar kernels. It then selects filters via the MCP framework to maximally cover clusters, prioritizing the retention of filters containing cross-channel diverse kernels, while balancing compression rate and accuracy through dynamically adjusted pruning ratios and channel regrowth strategies.

These methods collectively demonstrate a paradigm shift toward multi-dimensional importance metrics. Thereby enabling finer-grained trade-offs between model compression and performance preservation compared to traditional magnitude-only approaches. Therefore, we propose a dual-dimensional evaluation metric that originates from two critical statistical characteristics of convolutional layer channel parameters, specifically assessing each channel's global performance.

3 Methodology

We propose a statistical feature driven methodology to identify structural redundancy in deep neural networks. The convolutional layer parameters' mean and variance characteristics are systematically integrated into this framework, with the comprehensive workflow formalized in Algorithm 1. We will elaborate on the research approach and specific implementation of our method.

Algorithm 1: Statistical Feature-Driven Regularization

Input: Pre-trained model $\mathcal{M} = \{W_l\}_{l=1}^{L}$, target sparsity p, $\gamma \in [0, 1]$
Output: Pruned model $\mathcal{M}'$
Build dependency graph D via Eq. 3;
Group layers $\{g_i\}$ where $g_i = \{j \mid G_{ij} = 1\}$;
`// Regularization Phase`
foreach $(x, y) \in \mathcal{D}_{reg}$ **do**
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{cross-entropy}} + \lambda \sum_{j=1}^{L} \sum_{k=1}^{K} I_j^{(k)} \|W_j^{(k)}\|_2$;
  Update $W \leftarrow W - \eta \nabla_W \mathcal{L}$;
end
`// Pruning Phase`
foreach $g \in \{g_i\}$ **do**
  foreach $W_g^{(k)}$ **do**
    $I_g^{(k)} = \gamma \cdot \frac{\mathbb{E}(W_g^{(k)})}{\max \mathbb{E}(W_g)} + (1 - \gamma) \cdot \frac{\mathrm{Var}(W_g^{(k)})}{\max \mathrm{Var}(W_g)}$;
    Prune channels in M with globally lowest I scores;
  end
end
$M' \leftarrow$ fine-tuning (Pruned M);
return M';

3.1 Preliminaries

We assume that a neural network consists of L layers, where the parameters of the i-th layer are denoted by W_i. The input and output channels of i-th layer are denoted by C_{in}^i and C_{out}^i respectively, and the width of the convolutional kernel is denoted by K. So The tensor representing the layer in a deep convolutional neural network (CNN) can be parameterized by:

$$W_i \in R^{C_{out}^i \times C_{in}^i \times K \times K} \tag{1}$$

With the help of DepGraph [3], we can construct a grouping matrix $G \in R^{L \times L}$ to find the coupled layers with inter-dependency to the i-th layer conveniently. The entries of the matrix G_{ij} means the presence of dependency between i-th layer and j-th layer, so the group corresponding to the i-th layer can be expressed by the formula below:

$$g(i) = \{j | G_{ij} = 1\} \tag{2}$$

And to reduce unnecessary computational overhead, the Dependency Graph D was introduced. The key difference between D and G is that D only registers the dependencies between adjacent layers. In other words, D is the transitive reduction of G. Notably D does not document the dependencies across different layers; instead, it focuses on the relationships between layer inputs and outputs. In detail, we refer to the input and output of W_i as W_i^- and W_i^+ respectively. And then define two simple rules for identifying their dependencies:

- Inter-layer Dependency: dependency $W_i^- \Leftrightarrow W_j^+$ always emerges in connected layers with $W_i^- \leftrightarrow W_j^+$.
- Intra-layer Dependency: dependency $W_i^- \Leftrightarrow W_i^+$ exists if and only if W_i^- and W_i^+ shares the same pruning schemes, denoted as $sch(W_i^-) = sch(W_i^+)$.

So the dependencies can be represented by the following formula:

$$D(W_i^-, W_j^+) = \underbrace{1[W_i^- \leftrightarrow W_j^+]}_{Inter-layerDep} \vee \underbrace{1[i = j \wedge sch(W_i^-) = sch(W_j^+)]}_{Intra-layerDep} \tag{3}$$

Utilizing the Dependency Graph D, we can effectively prune channels that are mutually dependent.

3.2 Relative importance of channels based on statistical feature

We begin by analyzing the parameter distribution of each layer in a sparse model. Ideally, we would expect the parameter distribution of a model to be bimodal, with redundant small parameters clustering in non-essential channels and significant parameters clustering in important channels, so that we can easily select unimportant channel through magnitude-based criteria. However, as shown in Fig.1, most of parameters in each layer is confined to a narrow range. Particularly in sparse models, a significant number of parameters approach zero, leading to many channels having similar norms.

As a result, magnitude-based criteria can easily confuse important channels with redundant ones when evaluated at the channel level.

Building upon the empirically observed phenomena, we consider that when judging the importance of a channel, we need to consider that the distribution of parameters should not be concentrated in the interval close to 0.

Our proposed dual-component evaluation metric strategically incorporates:

The aggregate magnitude of parameters within each channel, and the distributional characteristics indicating whether parameters are tightly clustered near zero or exhibit broader dispersion. To implement this principle, we derive a statistical feature-driven importance metric by formally defining channel significance:

$$I_j^{(k)} = \gamma \cdot \frac{E\left(W_j^{(k)}\right)}{\max E\left(W_j^{(m)}\right)} + (1-\gamma) \cdot \frac{\mathrm{Var}\left(W_j^{(k)}\right)}{\max \mathrm{Var}\left(W_j^{(m)}\right)} \tag{4}$$

where $E\left(W_j^{(k)}\right)$ denotes mean of channel k parameters in layer j (measures magnitude), $\mathrm{Var}\left(W_j^{(k)}\right)$ denotes variance of channel k parameters (measures stability), and γ denotes weight coefficients.

Our method categorizes channels into three distinct types: (1) Important channels with high mean $E\left(W^{(k)}\right)$ and high variance $\mathrm{Var}\left(W^{(k)}\right)$, reflecting strong activation magnitudes and diverse parameter distributions; (2) redundant channels with low mean and low variance, where parameters cluster tightly near zero; (3) Noise-dominated channels, which exhibit either high mean with low variance or low mean with high variance, both contributing negligible task-relevant information.

3.3 Regularization base on Statistical importance

After identifying the importance of each channel, we can further incorporate this information with group lasso to apply appropriate regularization terms to the channels. Stronger regularization constraints should be imposed on redundant channels, while weaker regularization should be applied to important channels. Our approach reduces the actual contribution of redundant channels, thereby enhancing the distinction between important and redundant channels.

As a result, we can derive a consistent sparse model that improves the channel pruning performance. And the formula is expressed as follows:

$$\mathcal{L}(w) = \mathcal{L}_{\text{cross-entropy}} + \lambda \sum_{j=1}^{L} \sum_{k=1}^{K} I_j^{(k)} \left|W_j^{(k)}\right|_2 \tag{5}$$

Where ($\mathcal{L}_{\text{cross-entropy}}$) refers to the cross-entropy loss from the model's predictions, while γ scales the regularization term. Empirical results in Chapter 4 demonstrate that such adaptive regularization significantly enhances pruning efficacy without compromising model accuracy.

3.4 Global Pruning

It is important to highlight that the channel importance we measure is relative to each layer's average. However, a significant observation during global pruning is that convolutional layers located in the middle of the network tend to exhibit more sparsity than those at the beginning or end. This leads to excessive pruning of middle layers, resulting in increased distortion as more channels are pruned simultaneously [8]. Therefore, when conducting global pruning, the interaction between pruned channels and the remaining channels must be carefully considered. Drawing inspiration from LAMP, we propose a layer-adaptive pruning framework based on statistical importance.

The importance of channels is determined relative to their respective layers. We define the relative importance of a channel with the following equation:

$$I_k = \gamma \cdot \frac{E\left(W_j^{(k)}\right)}{\sum_{p>k} E\left(W_j^{(p)}\right)} + (1-\gamma) \cdot \frac{\mathrm{Var}\left(W_j^{(k)}\right)}{\sum_{p>k} \mathrm{Var}\left(W_j^{(p)}\right)} \tag{6}$$

In this formulation, The terms $\sum_{p>k} E\left(W_j\right)$ and $\sum_{p>k} \mathrm{Var}\left(W_j\right)$ serve as normalization factors that account for the remaining channels' overall magnitude and variability. The combination of these two components ensures that we adaptively balance the pruning process, thus avoiding the excessive pruning of critical channels.

4 Experiments

4.1 Experimental setup

Dataset In our experiments, we conducted comprehensive pruning and fine-tuning studies on mainstream deep learning architectures including MobileNet, ResNet and VGG. The evaluation utilized three benchmark datasets with varying scales:

CIFAR-10/100 Datasets: As medium-scale benchmarks, CIFAR-10 contains 60,000 32×32 RGB images (50,000 training/10,000 testing) across 10 classes, while CIFAR-100 extends this to 100 fine-grained categories with 600 images per class. During training, we applied standard data augmentation including random cropping (with 4-pixel padding) and horizontal flipping.

ImageNet-1K Dataset: This large-scale dataset comprises 1.28 million training images and 50,000 validation images across 1,000 object categories. Following established protocols, we pre-served the original resolution (224×224 for most CNNs) and employed identity trans-formation for validation images

Training strategy All experiments are conducted on NVIDIA A800 GPUs using PyTorch. We initialize baseline models (ResNet, VGG and MobileNet) with pre-trained weights from torchvision and timm repositories. For CNN-based architectures (ResNet, VGG, MobileNet), the training process spans 100 epochs with a batch size of 128. The learning rate is initialized at 0.01 and progressively reduced to 0.0001 using a linear step scheduler, coupled with the SGD optimizer (momentum=0.9). Weight decay is set to 1e-4 to regularize model parameters.

4.2 Main Result on CIFAR-10 and CIFAR-100

We evaluate our method against several prominent pruning techniques, including Network Slimming, HRank, DepGraph and so on [3, 10, 11], across various architectures on the CIFAR-10 and CIFAR-100 datasets.

For CIFAR-10, we evaluate our method on ResNet-56, achieving a pruning accuracy of 93.77%, representing a 0.27% improvement over the baseline, while reducing FLOPs by 55.1%. This result demonstrates the superiority of our approach over other pruning methods, highlighting the effectiveness of the statistical feature-based regularization in preserving model performance during compression. On CIFAR-100, we apply our method to VGG-19, where we achieve a pruning accuracy of 71.50%, a 2.55% decrease from the baseline, with a significant reduction in FLOPs by 88.7%. Despite the slight accuracy drop, the substantial decrease in computational load underscores the efficiency of our pruning strategy.

Table 1. Pruning results on CIFAR-10 and CIFAR-100

Dataset	Architecture	Method	Baseline	$\text{Acc}_{\text{pruned}}$	ΔAcc	ΔFLOPs
CIFAR 10	ResNet-18	OTOv2 [1]	93.02%	92.86%	-0.16%	79.7%
		ATO [18]	94.41%	94.51%	0.1%	79.8%
		ours	94.04%	**94.27**%	**0.23**%	**80.6**%
	ResNet-56	L1 [9]	92.80%	91.80%	-1.0%	50.0%
		LAMP [8]	93.53%	93.16%	-0.37%	53.1%
		L2 [3]	93.53%	93.77%	0.23%	51.3%
		White-Box [19]	93.26%	93.54%	0.28%	55.6%
		ATO [18]	93.50%	93.74%	0.24%	55.0%
		ours	93.50%	**93.77**%	**0.27%**	**55.1%**
CIFAR 100	VGG-19	EigenD [15]	73.3%	65.18%	-8.16%	88.6%
		Greg [16]	74.02%	67.75%	-6.27%	88.7%
		L2 [3]	73.50%	70.39%	-3.11%	88.7%
		ours	74.05%	**71.50%**	**-2.55%**	**88.7%**
	ResNet-18	OTOv2 [1]	-	74.96%	-	39.8%
		ATO [18]	77.95%	76.79%	-1.16%	40.1%
		ours	77.62%	**76.91**%	**-0.71**%	**44.75**%
	ResNet-34	OTOv2 [1]	-	74.96%	-	49.5%
		ATO [18]	78.43%	78.54%	0.11%	49.5%
		ours	78.13%	**78.44**%	**0.31**%	**50.3**%
	MobileNetv2	L2 [3]	71.11%	71.67%	0.56%	33.4%
		ours	70.78%	**71.59**%	**0.81**%	**50.34**%

What's more, as illustrated in Figure 2, our method consistently outperforms other methods in terms of both accuracy and compression, even at higher compression rates, demonstrating its robustness and effectiveness across a range of pruning levels.

These results validate the efficacy of our channel pruning method, which utilizes convolutional layer statistical features to identify and retain critical channels, leading to models that are both efficient and performant across different architectures and datasets.

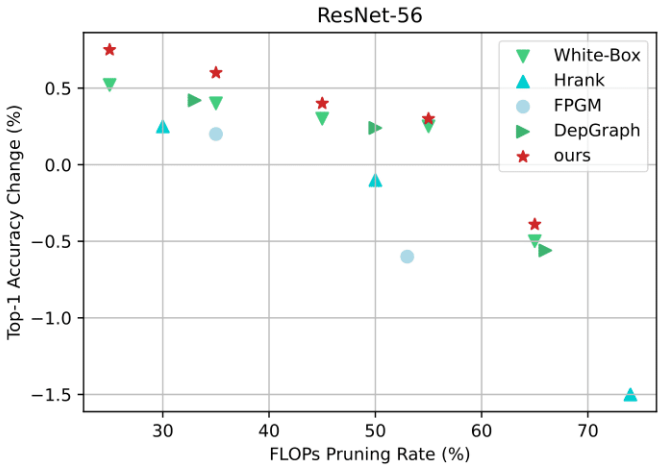


Fig. 2. Comparison of Top-1 accuracy between existing methods and our approach under varying FLOPs pruning rates. Experiments were performed using ResNet-56 architectures on the CIFAR-10 dataset.

4.3 Main Result on ImageNet

We also evaluate our pruning method on the ImageNet dataset using the ResNet-50 architecture and compare its performance with existing pruning techniques. As shown in Table 2, The original ResNet-50 model achieves a top-1 accuracy of 76.15% on the ImageNet validation set. After applying our pruning method, the pruned model achieves a top-1 accuracy of 75.34%, resulting in a decrease of 0.81% from the baseline. In terms of computational efficiency, the pruned model demonstrates a 54.8% reduction in FLOPs, significantly lowering the computational burden.

In summary, our channel pruning method effectively reduces computational complexity in ResNet-50 models on ImageNet, offering a competitive trade-off between accuracy and efficiency.

Table 2. Pruning results on ImageNet

Architecture	Method	Baseline	$\mathbf{Acc}_{\mathbf{pruned}}$	ΔAcc	ΔFLOPs
ResNet-50	BN [11]	76.10%	75.30%	-0.50%	50.0%
	HRANK [10]	76.15%	75.10%	-1.05%	43.9%
	L2 [3]	76.15%	75.18%	-0.97%	54.6%
	White-Box [19]	76.15%	75.32%	-0.83%	45.6%
	ours	76.15%	**75.44%**	**-0.71%**	**54.8%**

4.4 Ablation Study

To validate the effectiveness of the proposed statistical feature-based regularization method within the pruning framework, we conducted a systematic ablation study on the gamma parameter, which was evenly partitioned into ten incremental steps from 0 to 1, with detailed experimental results summarized in Table 3.

Through comparative analysis of pruning outcomes across these configurations, our objective is to demonstrate empirically whether the integration of both statistical features (mean and variance) provides superior pruning decision-making compared to scenarios where either feature is utilized independently.

This experimental design aligns with established methodologies in sparse neural network optimization, where parameter sensitivity analysis and feature fusion strategies have been shown to critically influence model compression efficacy. The structured exploration of gamma's role in balancing feature contributions further adheres to rigorous ablation protocols recommended in prior studies on pruning-induced regularization.

Table 3. Ablation Study of Structured Pruning on ResNet-56 for CIFAR-10 (All experiments use identical pruning ratios. ΔFLOPs reflects actual computation reduction. Bold value indicates best performance.)

γ	ΔFLOPs	Acc_{ori} (%)	Acc_{pruned}(%)	Acc_{fine}(%)
0.0	54.8	93.50	35.24	93.32
0.1	54.8	93.50	34.40	93.17
0.2	54.8	93.50	41.04	93.06
0.3	54.8	93.50	41.52	93.48
0.4	54.8	93.50	**41.86**	**93.74**
0.5	55.4	93.50	41.67	93.10
0.6	54.8	93.50	36.48	93.26
0.7	55.2	93.50	37.18	93.13
0.8	55.8	93.50	36.39	93.35
0.9	56.7	93.50	26.57	92.98
1.0	56.8	93.50	27.12	93.22

5 Conclusion

This study addresses the limitations of magnitude-based structured pruning in sparse models by proposing a statistical feature-driven regularization framework. Through systematic analysis of parameter distributions in sparse layers, we introduce a dual-component importance metric combining channel-wise mean and variance. The methodology incorporates dependency-aware group regularization via inter-layer dependency graphs, enabling adaptive parameter shrinkage while preserving critical feature representations. Extensive experiments on CIFAR and ImageNet benchmarks demonstrate superior compression-accuracy trade-offs compared to existing methods. We hope that our work provides a new perspective for channel pruning methods and inspires further innovation in this field.

Acknowledgments. This work was supported by the Postdoctoral Fellowship Program of CPSF (under Grant No. GZB20240113), the Sichuan Science and Technology Program (granted No. 2024ZDZX0011 and No. 2025ZNSFSC1472), and the Sichuan Central-Guided Local Science and Technology Development Program (under Grant No. 2023ZYD0165).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Chen, T., Liang, L., Ding, T., Zhu, Z., Zharkov, I.: OTOv2: Automatic, Generic, User-Friendly. (2022)
2. Cheng, H., Zhang, M., Shi, J.Q.: A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations. IEEE Trans. Pattern Anal. Mach. Intell. 46(12) (2024)
3. Fang, G., Ma, X., Song, M., Mi, M.B., Wang, X.: Depgraph: Towards any structural pruning. In: CVPR. pp. 16091–16101 (2023)
4. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. In: NeurIPS. vol. 1, pp. 1135–1143 (2015)
5. He, Y., Kang, G., Dong, X., Fu, Y., Yang, Y.: Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks. In: IJCAI. pp. 2234–2240 (2018)
6. He, Y., Xiao, L.: Structured Pruning for Deep Convolutional Neural Networks: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 46(5), 2900–2919 (2024)
7. Hu, H., Peng, R., Tai, Y.W., Tang, C.K.: Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. In: ECCV. pp. 45–60 (2016)
8. Lee, J., Park, S., Mo, S., Ahn, S., Shin, J.: Layer-adaptive Sparsity for the Magnitude-based Pruning. arXiv preprint arXiv:2007.00389 (2020)
9. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: ICLR (2017)
10. Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., Shao, L.: HRank: Filter Pruning Using High-Rank Feature Map. In: CVPR. pp. 1529–1538 (2020)
11. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: ICCV. pp. 2755–2763 (2017)

12. Liu, Z., Sun, M., Zhou, T., Huang, G., Darrell, T.: Rethinking the Value of Network Pruning. In: ICLR (2018)
13. Lu, Y., Guan, Z., Yang, Y., Zhao, W., Gong, M., Xu, C.: Entropy Induced Pruning Framework for Convolutional Neural Networks. In: AAAI. 38(4), 3918–3926 (2024)
14. Park, M., Kim, D., Park, C., Park, Y., Gong, G.E., Ro, W.W., Kim, S.: REPrune. In: AAAI. 38(13), 14545–14553 (2024)
15. Wang, C., Grosse, R., Fidler, S., Zhang, G.: EigenDamage: Structured Pruning in the Kronecker-Factored Eigenbasis. In: ICML. pp. 6566–6575. PMLR (2019)
16. Wang, H., Qin, C., Zhang, Y., Fu, Y.: Neural Pruning via Growing Regularization. arXiv preprint arXiv:2001.10576 (2020)
17. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: NeurIPS. vol. 29, pp. 2074–2082 (2016)
18. Wu, X., Gao, S., Zhang, Z., Li, Z., Bao, R., Zhang, Y., Wang, X., Huang, H.: Auto-Train-Once: Controller Network Guided Automatic Network Pruning from Scratch. In: CVPR. pp. 16163–16173 (2024)
19. Zhang, Y., Lin, M., Lin, C.W., Chen, J., Wu, Y., Tian, Y., Ji, R.: Carrying out CNN channel pruning in a white box. IEEE Trans. Neural Netw. Learn. Syst. 34(10), 7946–7955 (2023)
20. Zou, H., Hastie, T.: Regularization and Variable Selection Via the Elastic Net. J. R. Stat. Soc. Series B Stat. Methodol. 67(2), 301–320 (2005)