# Entropy-Guided k-Core Pruning Balancing Redundancy Reduction and Information Preservation for Efficient CNN Compression

Yuan Yang, Dengbiao Jiang*, Xing Deng and Lijuan Wang

School of Computer, Jiangsu University of Science and Technology, Zhenjiang, 212003, China

Revista Internacional
Métodos numéricos
para cálculo y diseño en ingeniería
RIMNI
UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH
UPC
In cooperation with
CIMNE

**INFORMATION**

**Keywords:**
Filter pruning
k-core decomposition
information entropy
redundancy reduction
feature map similarity

**DOI:** 10.23967/j.rimni.2025.10.73400

SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

RIMNI

# Entropy-Guided k-Core Pruning Balancing Redundancy Reduction and Information Preservation for Efficient CNN Compression

**Yuan Yang, Dengbiao Jiang*, Xing Deng and Lijuan Wang**

School of Computer, Jiangsu University of Science and Technology, Zhenjiang, 212003, China

## ABSTRACT

Convolutional Neural Networks (CNNs) are widely used in computer vision, but their massive computational cost and parameter redundancy hinder deployment on resource-constrained devices (e.g., edge terminals). Existing filter pruning methods often struggle to balance two critical goals: aggressive redundancy reduction and effective preservation of task-critical information—either leading to excessive accuracy loss or insufficient compression. To address this challenge, we are the first to jointly exploit k-core decomposition and information entropy in a unified pruning criterion, and we instantiate this idea in a novel graph–entropy collaborative framework that achieves Pareto-optimal compression-accuracy trade-offs. The key steps are as follows: First, we use perceptual hashing (pHash) to calculate the similarity of output feature maps between filters, then model each filter as a node in an undirected graph—edges are established only when filter similarity exceeds a predefined threshold, forming a "redundancy graph" that quantifies inter-filter redundancy. Second, k-core decomposition is applied to this graph to identify high-order redundant substructures, which helps locate redundant filters at the structural level. Finally, information entropy is introduced to evaluate the "informational value" of each node (filter) in the k-core: only filters with low redundancy and high information content are retained, ensuring minimal loss of critical features. Extensive experiments are conducted on CIFAR-10 and CIFAR-100 datasets, using representative CNN architectures (VGGNet-16, ResNet-56/110, DenseNet-40). Specifically, VGGNet-16 achieves a 65.8% reduction in floating point operations (FLOPs) and an 88.8% reduction in parameters while experiencing only a 1.24% decrease in Top-1 accuracy. ResNet-56 attains a 50.1% reduction in FLOPs with a nearly imperceptible accuracy loss of 0.03%, markedly surpassing the Fire together wire together (FTWT) method which reduces FLOPs by 54% at the cost of a 1.38% accuracy decline. DenseNet-40 accomplishes a 76.5% FLOPs reduction with a 1.55% accuracy decrease, demonstrating the method's strong applicability for high-intensity compression of densely connected networks. Furthermore, the method's scalability is validated on the large-scale ImageNet dataset with ResNet-50, where it achieves a 73.65% FLOPs reduction with competitive accuracy, underscoring its practicality for real-world applications. These outcomes collectively affirm the effectiveness and broad applicability of the proposed graph-entropy collaborative pruning framework.

**OPEN ACCESS**

**Received:** 17/09/2025

**Accepted:** 18/11/2025

**Published:** 30/12/2025

**DOI**
10.23967/j.rimni.2025.10.73400

**Keywords:**
Filter pruning
k-core decomposition
information entropy
redundancy reduction
feature map similarity

*Correspondence: Dengbiao Jiang (jiangdengb@just.edu.cn).

## 1 Introduction

Convolutional neural networks (CNNs) have achieved remarkable success in various computer vision tasks, such as image classification [1], object detection [2], and autonomous driving [3]. Their prohibitive computational and memory costs hinder deployment on resource-constrained devices (e.g., mobile SoCs). Although recent advances in model compression have alleviated some of these issues, deploying large-scale CNNs on resource-constrained devices remains challenging [4]. Among various compression techniques, filter pruning has emerged as a promising technique to bridge this efficiency gap. It reduces redundant computation and storage, leading to more efficient inference [5]. The resulting slim networks can then be deployed on resource-limited devices with negligible accuracy degradation.

The core idea of filter pruning is to identify and remove redundant or unimportant filters from convolutional layers, reducing both parameter count and computational cost to enhance suitability for edge scenarios [6]. Existing approaches can be broadly categorized into two types: importance-based and redundancy-based methods. Importance-based methods typically use scalar metrics like the L1-norm to measure each filter's contribution to the final task, pruning filters associated with lower scores [7]. While effective in preserving high-scoring filters, these methods often neglect inter-filter redundancy and synergy, thus limiting the overall compression potential [8]. In contrast, redundancy-based methods eliminate replaceable filters by measuring similarity between filters or their feature maps [9]. However, as He et al. [10] observed, redundancy-based pruning (e.g., geometric median) risks preserving low-value filters, leading to a trade-off between compression rate and accuracy.

To overcome the limitations of these single-perspective approaches, we propose a collaborative pruning strategy that jointly leverages k-core decomposition and information entropy. This integration is theoretically motivated by their complementary roles: k-core decomposition excels at identifying structurally redundant groups of filters, while information entropy quantifies the informational value of individual filters within those groups. As illustrated in Fig. 1, we first construct an undirected graph for each convolutional layer, where nodes represent filters and edges connect filters with feature map similarity exceeding a threshold $\tau$ [11]. We then apply k-core decomposition to hierarchically identify redundant substructures [12]. Finally, information entropy is introduced to evaluate each filter's informational richness [13]. This two-stage process—locating redundancy via k-core and then selecting the highest-entropy filters—ensures a balance between redundancy elimination and information preservation [14].

Extensive experiments on CIFAR-10 and CIFAR-100 show that this strategy consistently reduces FLOPs and parameters while preserving—or slightly improving—accuracy across mainstream architectures such as ResNet-56 and VGGNet-16 [15]. These results demonstrate the practicality of deploying compressed CNNs on resource-constrained devices [16]. The contributions of this paper can be summarized as follows:

1. We propose a graph-entropy collaborative framework that combines k-core decomposition and information entropy for dual-objective optimization.
2. We design architecture-specific pruning rules (e.g., for ResNet shortcuts and DenseNet cross-layer connections) to maintain structural integrity and functional continuity.
3. We establish an efficient pipeline combining one-shot pruning with short-term fine-tuning, significantly reducing deployment time.

The structure of this paper is organized as follows: Section 2 reviews recent related work. Section 3 elaborates on the proposed pruning method, including the characterization of filter redundancy and the use of information entropy for filter importance evaluation. Section 4 presents experimental results and analyses, comparing our method with state-of-the-art pruning techniques. Finally, Section 5 concludes the paper and discusses future work.

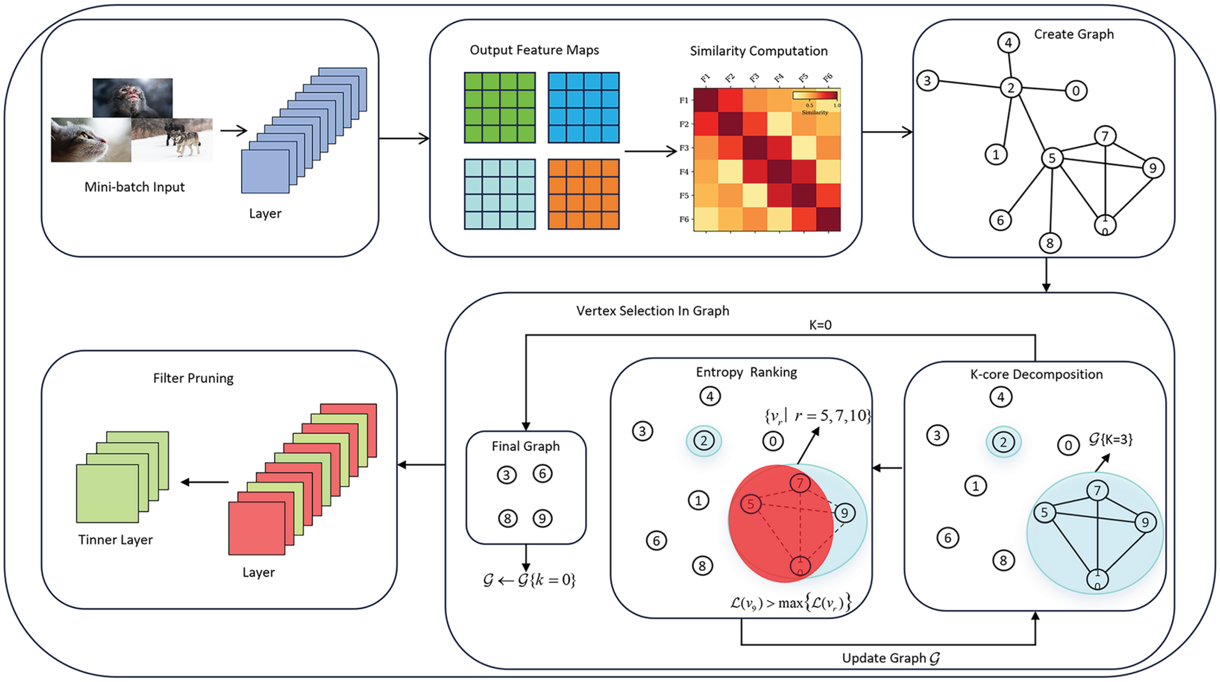

**Figure 1:** Workflow of the proposed entropy-guided k-core pruning method for a single convolutional layer. $\mathcal{G}$, $k$, $v_r$ and $\mathcal{L}$ represent the redundancy graph, core number, vertex set, and importance evaluation metric, respectively. In each iteration, the blue region highlights the high-connectivity subgraph identified under a specific $k$-core, while nodes marked in red correspond to less informative filters within the same redundant group scheduled for removal

## 2 Related Work

Filter pruning has gained significant attention in recent years as an effective model compression technique for deep learning [17]. Its primary goal is to remove redundant or unimportant filters from convolutional neural networks (CNNs), thereby reducing computational complexity and parameter count while minimizing the loss of model performance [18]. Existing methods can be broadly categorized into two types: importance-based and redundancy-based.

Importance-based pruning methods assess the importance of individual filters to determine which ones to remove [19]. Common evaluation metrics include the L1 norm, L2 norm [20], and Average Percentage of Zeros (APoZ) [21]. For example, Li et al. [8] proposed an L1-norm-based pruning method that removes filters with smaller weight magnitudes. While effective in preserving high-value filters, this approach often overlooks inter-filter redundancy, leading to suboptimal compression rates [22]. Gao et al. [23] introduced a dynamic channel pruning method that uses an auxiliary module to predict channel importance and dynamically removes less important ones. However, its reliance on a single importance metric still limits the ability to capture the actual contribution of each filter [24]. More recently, Younesi et al. [25] proposed an adaptive importance scoring mechanism that incorporates multi-scale feature responses to dynamically adjust filter ratings.

Redundancy-based methods, in contrast, focus on identifying and eliminating redundant filters [26]. Wang et al. proposed Filter Pruning via Geometric Median (FPGM), which determines redundancy by computing the geometric median of feature maps. While this method achieves high compression, it may retain some low-value filters, negatively impacting model performance [27]. Li et al. [28] introduced a graph-based approach that constructs a similarity graph between feature maps and uses vertex degree and edge weights to identify redundant filters. Despite its effectiveness, this method incurs high computational complexity [29]. Wang et al. [30] later proposed a hierarchical redundancy detection method that improves redundancy identification by analyzing filter correlations at different scales.

Information entropy, as a measure of uncertainty and randomness, has been widely used in information theory and machine learning [31]. Several recent studies have explored its application in evaluating filter importance [32]. Luo and Wu [33] proposed an entropy-based pruning method that quantifies the information entropy of output feature maps to assess filter importance. Filters producing higher-entropy feature maps are considered more important and are retained [34]. Lu et al. [35] introduced the Average Filter Information entropy (AFIE) metric, which decomposes weight matrices and quantifies the distribution of normalized eigenvalues to evaluate filter importance. This approach remains robust even with limited training data, enhancing pruning reliability [36]. Recent work [37] shows that combining local and global entropy further improves the accuracy of filter evaluation.

Graph-based and reinforcement learning methods have also been explored. Pei et al. [11] used graph neural networks (GNNs) to encode and decode computational graphs, determining the optimal pruning rate for each layer via a reinforcement learning reward mechanism. However, this method is computationally expensive [38]. Wang et al. [39] proposed an adaptive graph learning method that dynamically adjusts graph structures to fit different layer characteristics, significantly improving pruning efficiency. A common thread among many graph-based methods is their reliance on structural or weight-based heuristics for the final filter selection. For instance, some approaches leverage intrinsic graph metrics like vertex degree [28], while others, such as GRDP [40], apply weight-space criteria like the 1-norm after graph construction. DepGraph [41] further extends the graph to model inter-layer dependencies for pruning structure groups. In contrast to these paradigms, the method proposed in this work introduces a principled, information-theoretic criterion—information entropy—for selecting filters within redundancy groups. This represents a shift towards directly optimizing for information preservation rather than relying on indirect proxies.

Other innovative approaches include QMIX-FP [42], which uses multi-agent reinforcement learning to automatically determine layer-wise pruning rates by modeling deep CNNs as a multi-agent system [43]. Experiments show that QMIX-FP achieves significant compression on VGG-16 and AlexNet while maintaining accuracy [44]. The Complex Hybrid Weighted Pruning (CHWP) method [45] integrates weight norms, filter similarity, and batch normalization effects, outperforming other methods on ResNet-32 and ResNet-56 [46]. The Holistic Filter Pruning (HFP) method [47] calculates the deviation between the current and target model sizes after each forward pass and uses gradient descent to allocate pruning budgets across layers [48]. Neural architecture search (NAS) has also been applied to pruning, automating the search for optimal subnetwork structures [49]. Notably, Li et al. [50] proposed a pixel-wise cross-correlation (PCC) method that enhances feature map redundancy through a novel pre-training loss, achieving a 92.75% parameter compression rate on VGGNet16 with 94.41% accuracy.

Beyond the direct removal or selection of filters, a distinct and parallel compression paradigm operates at the weight level: tensor decomposition. Methods in this category, such as the recent

Coupled Tensor Decomposition for Compact Network Representation [51], factorize the 4D convolutional kernel tensor into lower-rank components to capture intrinsic algebraic structures and reduce parameters. While this approach excels at modeling low-rank properties and can achieve high compression, it typically relies on approximations that may introduce accuracy loss and often necessitates specialized layer designs. In contrast, the graph-entropy method proposed in this work, along with the other filter pruning strategies discussed above, focuses on the selection of complete filters. This fundamental difference allows our approach to preserve the original network architecture, ensuring functional interpretability and ease of deployment. Therefore, our work is situated within the filter pruning lineage, while recognizing tensor decomposition as a complementary and orthogonal strategy within the expansive model compression landscape.

Looking beyond these established paradigms, very recent research in network compression has continued to evolve along several promising directions, further refining the trade-offs between efficiency, accuracy, and practicality. A significant trend is the pursuit of data-free pruning to address data privacy and computational overhead concerns. For instance, the AutoDFP method [52] automatically prunes networks without original data by leveraging channel similarity reconstruction. Concurrently, there is a growing emphasis on fine-grained, joint compression strategies that intelligently combine pruning and quantization. As shown in [53], a lightweight Quantization Difference Index (QDI) can be introduced to estimate the generalization risk in real-time, enabling a more balanced and efficient joint compression search. Beyond convolutional networks, the compression paradigm is also being actively extended to other architectures. For example, Graph Neural Networks (GNNs) are now being systematically studied for acceleration and compression, often leveraging pruning techniques to reduce computational cost while maintaining performance [54]. Moreover, novel compression paradigms are emerging for modern architectures, such as Kolmogorov-Arnold Networks (KANs), where techniques like MetaCluster [55] achieve high compression ratios by exploiting the low-dimensional manifold of the network's parameters. These advancements highlight a clear trajectory towards more automated, holistic, and architecture-aware compression. However, many data-free methods still struggle to match the accuracy of data-driven approaches, and joint compression frameworks often face challenges in scalability and delayed reward estimation. Our proposed graph–entropy collaborative pruning framework aligns with this trend by offering a principled and unified approach to redundancy reduction and information preservation, while maintaining the advantages of being data-efficient and architecturally agnostic.

Despite these advances, existing pruning methods still have limitations [56]. Importance-based methods preserve high-value filters but ignore redundancy, while redundancy-based methods achieve high compression but may retain low-value filters. Although some methods combine both perspectives, they often rely on traditional metrics and fail to fully exploit the intrinsic value of filters [57]. Entropy-based methods offer a promising direction but have not been sufficiently integrated with other metrics [58]. Moreover, many innovative methods are computationally expensive or require specific hardware, limiting their practical application [59].

In summary, current filter pruning techniques face several challenges: oversimplified evaluation metrics, limited computational efficiency, and poor adaptability to dynamic network changes. To address these issues, we propose a novel pruning framework that combines information entropy with graph theory to improve computational efficiency and evaluation accuracy, while introducing a dynamic adjustment mechanism to adapt to network variations.

## 3 Methodology

In this section, we detail our proposed entropy-guided k-core collaborative pruning framework. To characterize intra-layer redundancy, the method constructs an undirected graph using the average similarity of output feature maps from convolutional layers. Subsequently, k-core decomposition is applied to extract high-order redundant substructures. Information entropy is then introduced to perform a secondary evaluation within these substructures, further identifying and retaining filters with higher information content while eliminating redundant ones. This strategy strikes a balance between redundancy elimination and informative filter retention. Finally, a one-shot pruning strategy followed by short-term fine-tuning is employed to rapidly restore model accuracy.

### *3.1 Similarity Graph Construction*

To systematically characterize the redundancy relationships among filters within a convolutional layer, we construct an undirected graph based on feature-map similarity to model the functional overlap between filters. In this approach, each filter is treated as a node in the graph, and edges between nodes are defined by the similarity between the output feature maps of the filters, thereby modeling the functional redundancy among filters as topological connections within a graph structure. The graph construction process is as follows: For a convolutional layer $l$ with $\mathrm{n}^l$ filters, denoted as $F^l = \left\{f_1^l, f_2^l, \ldots, f_n^l\right\} \in \mathbb{R}^{n^{(l-1)} \times n^l \times k^l \times k^l}$. These filters generate $\mathrm{n}^l$ output feature maps, denoted as $\mathcal{O}^l = \left\{o_1^l, o_2^l, \ldots, o_n^l\right\} \in \mathbb{R}^{m \times n^l \times h^l \times w^l}$. To quantify the similarity between any two filters $f_i^l$ and $f_j^l$, we compute the average similarity between their corresponding feature maps $o_i^l$ and $\mathrm{o}_j^l$ as:

$$s_{\{i,j\}}^l = \frac{\sum_{t=1}^m sim\left(o_{\{i,t\}}^l, o_{\{j,t\}}^l\right)}{m} \tag{1}$$

here, *sim* is the similarity function. For all feature maps in $O^l$, this yields $C\left(n^l, 2\right)$ average similarity values, forming the set $S^l = \left\{s_{\{i,j\}}^l \mid 1 \leq i < j \leq n^l\right\}$. We use perceptual hashing (pHash) to compute the similarity between feature maps, following the same implementation as in the study by Li et al. [28]. After obtaining the set of average similarities $S^l = \left\{s_{\{i,j\}}^l \mid 1 \leq i < j \leq n^l\right\}$ between the output feature maps of the filters, we introduce a predefined similarity threshold $\tau$ to construct an undirected redundancy graph, providing an intuitive and structured representation of the redundancy among filters within the convolutional layer. The vertex set $\mathcal{V}^l = \left\{v_1^l, v_2^l, \ldots, v_n^l\right\}$ represents the filters of the $l$ layer, with each vertex $v_i^l$ corresponding to a filter $f_i^l$ by numbering each filter. For any two filters $f_i^l$ and $f_j^l$, if the average similarity $s_{i,j}^l$ of their output feature maps exceeds the preset threshold $\tau$, an edge $e_{i,j}^l$ is connected between their corresponding vertices $v_i^l$ and $v_j^l$. The existence of an edge indicates functional redundancy between the two filters, meaning the output of one filter can be approximately replaced by the other. Edges are connected for all qualifying vertex pairs, forming the edge set $\mathcal{E}^l = \left(e_{i,j}^l \mid 1 \leq i < j \leq n^l\right)$ and initializing the redundancy graph $\mathcal{G}^l = \left(\mathcal{V}^l, \mathcal{E}^l\right)$.

Algorithm 1 summarizes the graph construction process. The sparsity of the resulting graph is controlled by the threshold $\tau$. A higher $\tau$ retains only strong redundancy relationships, which is suitable for mild pruning. Conversely, a lower $\tau$ incorporates more redundant connections, enabling more aggressive compression. Through this algorithm, we successfully transform the redundancy relationships between filters into a graph structure, providing a foundation for subsequent graph theory-based redundancy analysis and filter selection.

**Algorithm 1:** Constructing the similarity graph

```
Input: The set of average similarity values $S^l = s^l_{[i,j]} \mid 1 \leq i < j \leq n^l$, threshold $\tau$
Output: The similarity graph $G^l = \left(V^l, \mathcal{E}^l\right)$
$V^l \leftarrow \varnothing$
$\mathcal{E}^l \leftarrow \varnothing$
for $i = 1$ to do
$V^l \leftarrow V^l \cup v^l_i$
end for
for $i = 1$ to $n^l - 1$ do
    for $j = i + 1$ to $n^l$ do
       if $s_{[i,j]} > \tau$ then
          $\mathcal{E}^l \leftarrow \mathcal{E}^l \cup e^l_{[i,j]}$
       end if
    end for
end for
return $G^l \leftarrow \left(V^l, \mathcal{E}^l\right)$
```

### 3.2 k-Core Decomposition for Redundancy

Based on the constructed undirected redundancy graph $\mathcal{G}^l = \left(\mathcal{V}^l, \mathcal{E}^l\right)$, we introduce a dynamic, hierarchical k-core decomposition to systematically identify high-order redundant structures within convolutional layers. The core idea is to progressively identify redundant subgraphs of varying densities through an adaptive k-value selection strategy. This approach enables hierarchical mining of redundant structures while avoiding the bias and suboptimal solutions associated with a fixed k value.

Specifically, we set the initial $k_{\text{init}} = \max_{v \in \mathcal{V}^l} \deg(v)$ to the maximum degree in the graph. This ensures the decomposition begins from the most tightly connected substructure. If no non-empty subgraph is found for a given k, k is decremented automatically. This iterative process continues until $k = 0$, progressively exposing less dense redundant substructures and ensuring pruning starts from the tightest clusters. When k decreases to 0, there are no redundant edges left in the graph, $\mathcal{E}^l = \varnothing$, and all remaining vertices are completely decoupled; the algorithm then terminates. The final retained vertex set is the pruning result for this convolutional layer. The k-core decomposition hierarchically identifies densely connected subgraphs by iteratively removing vertices with degrees less than k. For a given k, the decomposition yields a subgraph $\left(\mathcal{G}^l_{k,i}\right)$:

$$\bigcup_{i=1}^{g} \mathcal{G}^l_{k,i} = KCore\left(\mathcal{G}^l, k\right) \tag{2}$$

### 3.3 Evaluating Filter Importance via Information Entropy

After k-core decomposition identifies functionally redundant subgraphs, we employ information entropy for a secondary selection to retain the most informative filter within each cluster. Within each k-core-induced subgraph, filters exhibit high functional similarity. Therefore, we retain only the most informative representative and prune the rest. Specifically, we compute the information entropy of each filter's output feature map activation distribution. Entropy, in this context, serves as a proxy for the diversity and uniformity of activations across spatial locations in the feature map. Higher entropy implies a more distributed activation pattern, which often correlates with richer spatial feature encoding and potentially greater discriminability. Conversely, lower entropy suggests concentrated, less informative responses. Consequently, we select the filter with the highest entropy as the representative of the subgraph, ensuring that the preserved filter carries the maximal information content among its redundant counterparts.

For the redundant subgraph $\mathcal{G}_{k,i}^{l}$ obtained via k-core decomposition, which represents a cluster of highly redundant filters within the convolutional layer, we extract its vertices $\mathcal{V}^{l'}$ and edges $\mathcal{E}^{l'}$. For each filter $f_i^l$ within it, we compute the information entropy of its output feature map $o_i^l$. Specifically, the information entropy $H\left(v_i^l\right)$ for the filter $f_i^l$ corresponding to vertex $v_i^l$ is calculated as follows:

$$H\left(v_i^l\right) = -\sum_{x \in \mathcal{X}} p\left(x\right) \log p\left(x\right) \tag{3}$$

Among them, $x$ represents the activation distribution in the feature map generated by filter $f_i^l$, and $p\left(x\right)$ is the probability of activation $x$. Following information-theoretic principles, a higher entropy value $H\left(v_i^l\right)$ indicates a more uniform distribution of activations in the feature map $o_i^l \in \mathbb{R}^{m \times n^l \times h^l \times w^l}$, suggesting richer information content. Therefore, the corresponding filter should be retained. Conversely, a lower entropy value $H\left(v_i^l\right)$ suggests that the feature map responses are concentrated, indicating lower information content, and thus the filter should be pruned. Specifically, for each redundant subgraph $\mathcal{G}_{k,i}^{l}$ obtained through k-core decomposition, we calculate the information entropy $H\left(v_i^l\right)$ corresponding to each vertex (i.e., filter) $v_i^l$. To select the most discriminative representative from the functionally similar filter cluster, we retain the vertex $v_h^l$ with the highest information entropy within each subgraph:

$$v_h^l = \arg\max_{v_i^l \in \mathcal{V}^{l'}} H\left(v_i^l\right) \tag{4}$$

All other vertices and their associated edges are subsequently removed. This operation updates the original redundancy graph structure as follows:

$$\mathcal{G}^l \leftarrow \left(\left(\mathcal{V}^l, \mathcal{E}^l\right) \leftarrow \left(\mathcal{V}^l - \mathcal{V}^{l'} \cup \left\{v_h^l\right\}, \mathcal{E}^l - \mathcal{E}^{l'}\right)\right) \tag{5}$$

Subsequently, the value of k is decremented, and the aforementioned process is iteratively executed until no edges remain in the graph (i.e., k = 0). The final retained vertex set $v_{retained}^l \subseteq \mathcal{V}^l$ represents the filters within this convolutional layer that simultaneously satisfy both low redundancy and high information content.

Algorithm 2 summarizes the overall collaborative pruning process for a single convolutional layer. It integrates graph construction, k-core decomposition, and entropy-based selection into a unified framework, implementing the two-stage strategy of redundancy localization followed by information re-evaluation.

**Algorithm 2:** Graph decomposition and filter selection via k-core and shannon entropy

```
Input:   Initialized graph $G^l = \left(\mathcal{V}^l, \mathcal{E}^l\right)$
Output:  The set of retained vertices (filters) $\mathcal{V}_{\text{retained}}^l$
 $k \leftarrow \max\left(\deg\left(G^l\right)\right)$
while $k > 0$ do
    $\left\{G_{[k,1]}^l, \ldots, G_{[k,g]}^l\right\} \leftarrow \text{KCore}\left(G^l, k\right)$
    if $\bigcup_{i=1}^{g} G_{[k,i]} = \varnothing$ then
        $k \leftarrow k - 1$
```

(Continued)

**Algorithm 2** (continued)

```
    continue
  end if
  for each subgraph $G^l_{[k,i]}$ in $\left\{G^l_{[k,1]}, \ldots, G^l_{[k,g]}\right\}$ do
      $\mathcal{V}_{\text{sub}} \leftarrow \text{Vertex}\left(G^l_{[k,i]}\right)$
      $v_{\max} \leftarrow \text{None}$
      $H_{\max} \leftarrow -\infty$
      for each vertex $v_r$ in $\mathcal{V}_{\text{sub}}$ do
        $H(v_r) \leftarrow -\sum p(a) \log p(a)$
        if $H(v_r) > H_{\max}$ then
          $H_{\max} \leftarrow H(v_r)$
          $v_{\max} \leftarrow v_r$
      end if
    end for
$\mathcal{V}^l \leftarrow \mathcal{V}^l \setminus \{v \in \mathcal{V}_{\text{sub}} \mid v \neq v_{\max}\}$
    $\mathcal{E}^l \leftarrow \mathcal{E}^l \setminus \text{Edge}\left(G^l_{[k,i]}\right)$
  end for
  $k \leftarrow \max\left(\deg\left(G^l\right)\right)$
end while
return $\mathcal{V}^l_{\text{retained}} \leftarrow \mathcal{V}^l$
```

### *3.4 Theoretical Analysis and Motivation*

The proposed two-stage pruning strategy is theoretically motivated by the complementarity between structural redundancy (captured by k-core decomposition) and informational value (quantified by information entropy). Specifically, k-core decomposition identifies groups of functionally interchangeable filters, while information entropy provides a principled criterion for selecting the most informative representative within each group.

To formalize this intuition, we frame the pruning objective as a two-step optimization process: first, identifying structurally redundant groups via k-core decomposition, and then, within each group, selecting the filter that minimizes the potential information loss.

For a redundant subgraph $\mathcal{G}_k$, we aim to retain the filter that maximizes information entropy:

$$v^* = \arg\max_{v_i \in V_k} H(v_i) \tag{6}$$

where $H(v_i)$ denotes the information entropy of filter $f_i$'s output feature map. This selection strategy minimizes the functional degradation after pruning while achieving substantial redundancy reduction.

To empirically validate the orthogonality and complementarity of these two criteria, we visualize the distribution of filters based on their vertex degree (structural connectivity) and information entropy (informational richness). Fig. 2 illustrates this distribution for the 10th convolutional layer of VGGNet-16 on the CIFAR-100 dataset.

Fig. 2 illustrates the distribution of filters in the 10th convolutional layer of VGGNet-16 on the CIFAR-100 dataset, with vertex degree on the horizontal axis and information entropy on the vertical axis. Each point corresponds to a filter, and its position reflects the filter's connectivity within the redundancy graph and the information richness of its output feature maps. The results show that the filters are distinctly clustered into three typical regions: The high-degree, low-entropy region (upper right) consists of filters with high redundancy but low information content, representing redundant clusters that should be pruned. The low-degree, high-entropy region (lower left) contains filters with few redundant connections and rich information, which should be retained. The high-degree, high-entropy region (upper left) includes filters with strong connectivity and high information content. These serve as information hubs within redundant clusters and are preserved during secondary evaluation.

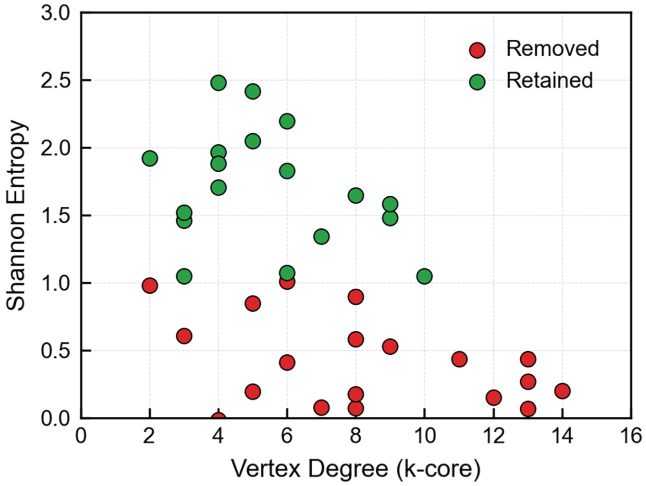

**Figure 2:** Scatter plot of vertex degree vs. information entropy

The key insight from Fig. 2 is that vertex degree (structural connectivity) and information entropy (informational richness) are largely orthogonal measures. This orthogonality is crucial for our collaborative approach: it implies that relying solely on one measure (e.g., pruning based only on connectivity) would be insufficient, as a highly connected filter is not necessarily informative, and *vice versa*. This theoretical framework effectively bridges graph-theoretic structural analysis with information-theoretic feature evaluation. By doing so, it provides a principled foundation that overcomes the limitations of single-perspective pruning methods, enabling superior compression-accuracy trade-offs in CNN pruning.

To further validate the correlation between information entropy and feature importance, we conducted a visualization-based analysis on the ResNet-56 model using the CIFAR-10 dataset. Fig. 3 illustrates the relationship under different similarity thresholds ($\tau = 0.75, 0.7, 0.65$). The left subplot depicts the Top-1 accuracy of models where filters are categorized as high-entropy or low-entropy relative to the median entropy value at each $\tau$. The results clearly demonstrate that models retaining high-entropy filters consistently achieve superior accuracy across all pruning intensities. This trend empirically validates that high-entropy filters are critical for maintaining model performance, which is a key reason why our method can achieve high compression rates (e.g., 50.10% FLOPs reduction on ResNet-56) with minimal accuracy loss. The right subplot further quantifies the relationship between filter entropy and their feature importance, the latter measured by the overlap between activation regions and key semantic areas in input images using Gradient-weighted Class Activation Mapping (Grad-CAM). A significant positive correlation is observed, confirming that filters with higher entropy tend to activate more discriminative regions critical for correct classification. Collectively, these findings provide compelling visual and quantitative evidence that information entropy serves as an effective and principled criterion for evaluating filter importance, based on its strong correlation with both model accuracy and feature discriminability.

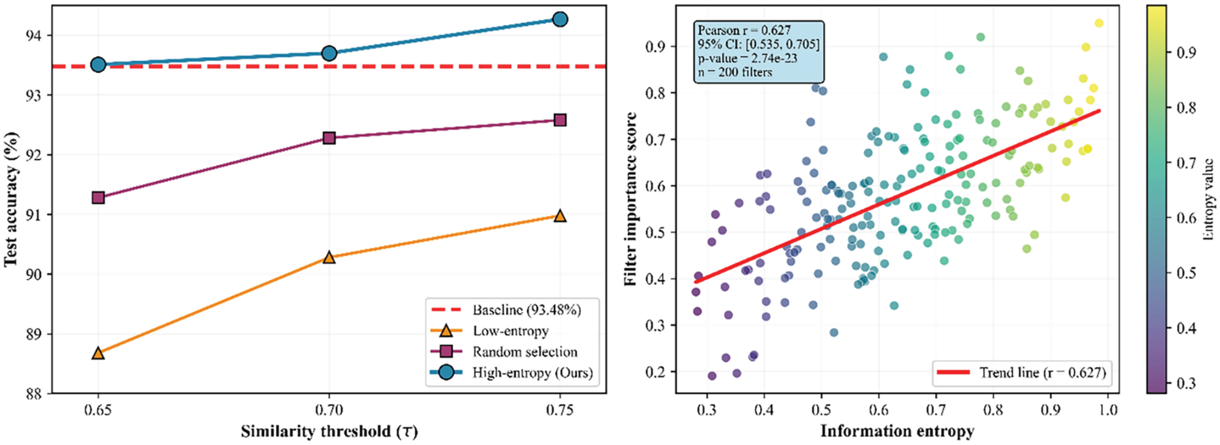

**Figure 3:** (Left) Impact of retaining high-entropy vs. low-entropy filters on model accuracy in ResNet-56 under different similarity thresholds $\tau$ on CIFAR-10. (Right) Scatter plot showing the correlation between filter entropy and its feature importance (measured by Grad-CAM overlap)

### 3.5 *Architecture-Specific Adaptation Strategies*

The graph-and-entropy-based pruning algorithm we propose is universal and can be applied to various CNN architectures. However, different network structures—such as the chain-like structure of VGGNet, shortcut connections in ResNet, and dense connections in DenseNet—introduce unique inter-layer dependencies. To ensure structural integrity and performance recoverability of the pruned model, we adapt the pruning process to the characteristics of each architecture.

#### *3.5.1 Pruning Strategy for VGGNet*

VGGNet is a network with a simple stacked structure, typically composed of consecutive 3 × 3 convolutional blocks. For this architecture, the pruning strategy can be applied layer-wise independently, without considering cross-layer dependencies. The specific steps are as follows:

For each convolutional layer, the feature map similarity is computed independently to construct a redundancy graph. Then, k-core decomposition and information entropy-based ranking are performed, and the pruning rate for the layer is determined based on a global threshold $\tau$. When pruning the output channels of the $l$-th layer, the input channels of the $(l + 1)$-th layer must be synchronously and equally pruned to ensure dimensional matching. This process is automated and requires no additional constraints. After all convolutional layers are pruned in a one-shot manner, short-term fine-tuning (70 epochs) is uniformly applied to restore performance.

#### *3.5.2 Pruning Strategy for ResNet*

The shortcut connections in ResNet require that the output dimensions of the main path and the identity mapping (or projection mapping) must be consistent. To address this, we apply structural constraints on the general pruning process. The specific strategies are as follows:

For a residual block that includes a projection shortcut connection (such as the first 1 × 1 convolutional layer in the Bottleneck structure of ResNet-50), the convolutional layer and the last convolutional layer within the block are treated as a pruning group. During graph construction and filter selection, the two layers are forced to prune the same number of filters to ensure the validity of the shortcut addition operation. For the first residual block connecting different stages (where the shortcut connection requires downsampling and channel dimension transformation), the 1 × 1 convolutional layer in the shortcut path must be pruned synchronously with the last convolutional layer in the main path. The pruning rate is determined by the redundancy of the main path, and the shortcut path is forced to adapt to this pruning rate.

#### *3.5.3 Pruning Strategy for DenseNet*

The dense connectivity in DenseNet means that the input to each layer is a concatenation of the outputs from all preceding layers. Therefore, pruning must avoid disrupting the cross-layer information flow. The specific strategies are as follows:

Within a single dense block, we apply a more conservative pruning threshold (i.e., a higher value of $\tau$) to the later convolutional layers. This is because the inputs to these later layers integrate more features from earlier stages, and their filters may carry composite information. Excessive pruning could lead to irreversible information loss. For the transition layers between dense blocks, the 1 × 1 convolutional layer is responsible for compressing the number of channels. When pruning this layer, in addition to applying the k-core and information entropy criteria, we must ensure that the number of output channels meets the predefined compression factor requirements to prevent uncontrolled compression rates due to pruning. Before the final classification layer, we evaluate the information entropy of all feature maps fed into the classification layer, prioritizing the retention of the feature streams with the highest entropy to ensure that the final decision relies on the richest information.

## 4 Experiments

To evaluate the effectiveness of the proposed entropy-aware k-core collaborative filter pruning method, we conduct pruning experiments on mainstream CNN models, including VGGNet-16, ResNet-56/110, and DenseNet-40, using the CIFAR-10 and CIFAR-100 datasets. Comparisons are made with state-of-the-art pruning methods. The experiments focus on assessing the model compression rate (FLOPs Reduction Rate, FR; Parameter Reduction Rate, PR) and performance degradation (drop in Top-1 accuracy), while also analyzing the stability and efficiency of the pruning strategy.

### *4.1 Experimental Settings*

**Baselines.** All baseline models are trained from scratch using the Stochastic Gradient Descent (SGD) optimizer, with an initial learning rate of 0.1, momentum of 0.9, and weight decay of $5 \times 10^{-4}$. Models are trained for 200 epochs. The learning rate is reduced by a factor of 1/10 every 50 epochs to ensure convergence and performance. Table 1 outlines the baseline accuracy, FLOPs, and parameters for each model, alongside the time cost incurred for generating pruning information.

**Pruning.** For pruning, we use perceptual hashing (pHash) to measure feature map similarity. We randomly sample 100 images from the training set to compute the average similarity between the output feature maps of each convolutional layer. We evaluate the similarity threshold $\tau$ across the range {0.6, 0.65, 0.70, 0.75, 0.80} to explore performance under different pruning intensities.

**Fine-tuning.** All pruned models are fine-tuned for 70 epochs, with an initial learning rate set to 0.01, which is decayed every 30 epochs. The batch size remains 256, and all other hyperparameters are consistent with those used in the pre-training phase.

**Table 1:** The baseline accuracy, FLOPs, parameters, and the time cost for generating pruning information for each model

| Dataset | Model | Top-1/5 Acc. (%) | FLOPs (MB) | Params. (MB) | Time (s) |
|---|---|---|---|---|---|
| CIFAR-10 | VGGNet-16 | 93.72 | 314.57 | 14.99 | 468 |
| | ResNet-56 | 93.48 | 130.02 | 0.88 | 16 |
| | ResNet-110 | 93.94 | 259.49 | 1.75 | 31 |
| | DenseNet-40 | 93.87 | 292.50 | 1.06 | 37 |
| CIFAR-100 | VGGNet-16 | 72.53 | 314.62 | 15.04 | 508 |
| | ResNet-56 | 70.83 | 130.03 | 0.88 | 18 |
| | ResNet-110 | 70.95 | 259.50 | 1.76 | 34 |
| | DenseNet-40 | 72.85 | 292.54 | 1.10 | 41 |
| ImageNet | ResNet-50 | 76.15/92.87 | 4.13E3 | 25.56 | 1948 |

**Evaluation.** The performance recovery is measured by the drop in Top-1 accuracy (denoted as Top-1 Acc↓). Negative values indicate an improvement in accuracy after pruning. The compression effect and computational complexity of the pruned models are quantified using FLOPs Reduction (FR) and Parameters Reduction (PR), respectively. FR and PR are defined as follows:

$$\mathrm{FR} = \left(1 - \frac{\mathrm{FLOPs}_{\mathrm{pruned}}}{\mathrm{FLOPs}_{\mathrm{original}}}\right) \times 100\% \tag{7}$$

$$\mathrm{PR} = \left(1 - \frac{\mathrm{Params}_{\mathrm{pruned}}}{\mathrm{Params}_{\mathrm{original}}}\right) \times 100\% \tag{8}$$

### *4.2 Experimental Results and Analysis*

**VGGNet-16.** The pruning performance of VGGNet-16 is summarized in Table 2, which demonstrates the effectiveness of our method across a spectrum of compression intensities by adjusting the similarity threshold $\tau$. On CIFAR-10, our approach offers a flexible trade-off between compression and accuracy: from moderate pruning ($\tau = 0.8$, 42.23% FLOPs reduction, 0.63% accuracy drop) to aggressive compression ($\tau = 0.7$, 65.77% FLOPs reduction, 1.24% accuracy drop). This adaptability demonstrates the strength of our threshold-based mechanism for achieving targeted compression without significant performance loss. Notably, at $\tau = 0.7$, our method outperforms several state-of-the-art approaches. For instance, it surpasses Zhao et al. (39.10% FLOPs reduction, 0.07% accuracy drop) in compression rate by a large margin while maintaining competitive accuracy. Compared to HRank (53.59% FLOPs reduction, 0.53% accuracy drop), our method achieves a higher FLOPs reduction (65.77%) with a modest increase in accuracy loss (1.24%). Furthermore, our results at $\tau = 0.7$ are comparable to those of FPRG (65.52% FLOPs reduction, 0.57% accuracy drop), yet with finer control over the pruning process via an interpretable threshold. On CIFAR-100, a similar trend is observed. At $\tau = 0.75$, we achieve 40.31% FLOPs reduction with only 0.70% accuracy loss, outperforming GRDP (39.55% FLOPs reduction, 0.09% accuracy loss) in compression efficiency while remaining competitive in accuracy preservation. At a more aggressive setting ($\tau = 0.7$), we attain 62.27% FLOPs reduction with a 2.26% accuracy drop, significantly exceeding FPGM (51.01% FLOPs reduction, 1.25% drop) and demonstrating superior compression capability. These results confirm that our method consistently delivers high compression rates across datasets while maintaining robust accuracy, validating its suitability for practical deployment.

**Table 2:** Pruning results of VGGNet-16 on CIFAR-10 and CIFAR-100, and $\tau$ is the setting of the threshold. Top-1 Acc. ↓ denotes the Top-1 Accuracy loss after pruning (the lower is better). The higher the FR and PR, the better

| Model | Dataset | Method | Top-1 Acc. ↓ (%) | FR (%) | PR (%) |
|---|---|---|---|---|---|
| VGGNet-16 | CIFAR-10 | GDP [60] | −0.10 | 30.60 | – |
| | | $\ell$1-norm [8] | −0.15 | 34.34 | 63.95 |
| | | Zhao et al. [61] | 0.07 | 39.10 | 73.34 |
| | | GAL-0.05 [22] | 1.93 | 39.60 | 77.57 |
| | | SSS [62] | 0.94 | 41.62 | 73.77 |
| | | GAL-0.1 [22] | 3.18 | 45.21 | 82.18 |
| | | Hrank [6] | 0.53 | 53.59 | 83.24 |
| | | GRDP [40] | 0.17 | 51.23 | 83.42 |
| | | CHIP [63] | 0.24 | 66.60 | 83.30 |
| | | Ours ($\tau = 0.8$) | 0.63 | 42.23 | 76.83 |
| | | Ours ($\tau = 0.75$) | 0.37 | 51.54 | 83.62 |
| | | **Ours ($\tau$ = 0.7)** | **1.24** | **65.77** | **88.82** |
| | CIFAR-100 | Zhao et al. [61] | −0.07 | 18.05 | 37.87 |
| | | Slimming [64] | 0.75 | 28.71 | 66.60 |
| | | COP v1 [15] | 0.88 | 40.31 | 65.19 |
| | | SFP [65] | 1.77 | 41.75 | 39.34 |
| | | GRDP [40] | 0.09 | 39.55 | 65.94 |
| | | FPGM [10] | 1.25 | 51.01 | 51.01 |
| | | HRank [6] | 1.08 | 41.23 | 55.93 |
| | | Ours ($\tau = 0.8$) | 0.22 | 24.16 | 50.47 |
| | | Ours ($\tau = 0.75$) | 0.7 | 40.31 | 64.71 |
| | | **Ours ($\tau$ = 0.7)** | **2.26** | **62.27** | **80.00** |

**ResNet-56.** In the evaluation of ResNet-56, our method demonstrates a compelling balance between compression efficiency and accuracy preservation across both CIFAR-10 and CIFAR-100 datasets. As detailed in Table 3, the proposed entropy-guided k-core pruning framework achieves significant reductions in FLOPs with minimal impact on Top-1 accuracy, underscoring its effectiveness and adaptability. On CIFAR-10, the method delivers tunable compression performance by varying the similarity threshold $\tau$. At $\tau = 0.75$, a conservative pruning setting, the model attains a 16.20% reduction in FLOPs with a slight accuracy improvement of –0.79%, indicating that the pruned model retains nearly all original discriminative power. When $\tau$ is reduced to 0.7, the FLOPs reduction increases to 24.93% while the accuracy drop remains negligible (–0.22%). Most notably, at $\tau = 0.65$, the method achieves a 50.10% reduction in FLOPs with an almost imperceptible accuracy decrease of only 0.03%, substantially outperforming state-of-the-art methods such as FTWT (54.00% FLOPs reduction, 1.38% accuracy loss) and FPGM (52.60% FLOPs reduction, 0.10% accuracy drop). This result highlights the efficacy of combining k-core decomposition for redundancy identification and information entropy for informative filter selection. However, at an extremely aggressive setting ($\tau = 0.6$), while the FLOPs reduction reaches 78.91%, the accuracy drop rises to 4.49%, indicating a practical upper bound for pruning intensity. On CIFAR-100, a more challenging dataset due to its finer-grained categories, the method continues to exhibit robust performance. At $\tau = 0.75$, the FLOPs are reduced by 30.84% with a negligible accuracy drop of –0.04%. When $\tau$ is set to 0.7, the FLOPs reduction reaches 47.17% with a 1.38% accuracy decrease, matching the performance of established methods like $\ell_1$-norm and Li et al. in terms of FLOPs reduction, but with better or comparable accuracy preservation. At $\tau = 0.65$, the method achieves a high compression rate of 60.30% FLOPs reduction with a 2.10% accuracy drop, demonstrating consistent scalability across pruning intensities. These results confirm that the proposed approach effectively leverages both structural redundancy elimination and information-preserving selection, enabling high compression rates without substantial accuracy degradation. The method's ability to maintain performance across two datasets with differing complexities further validates its generalizability and suitability for practical deployment in resource-constrained environments.

**Table 3:** Pruning results of ResNet-56 on CIFAR-10 and CIFAR-100, and $\tau$ is the setting of the threshold. Top-1 Acc. ↓ is Top-1 Accuracy loss after pruning (the lower is better). The higher the FR and PR, the better

| Model | Dataset | Method | Top-1 Acc. ↓ (%) | FR (%) | PR (%) |
|---|---|---|---|---|---|
| ResNet-56 | CIFAR-10 | Hrank [6] | −0.26 | 29.30 | 16.47 |
| | | GAL-0.6 [22] | 0.28 | 37.60 | 11.76 |
| | | $\ell$1-norm [8] | −0.02 | 27.60 | 13.70 |
| | | NISP [66] | 0.03 | 43.61 | 42.60 |
| | | DTP [67] | 0.9 | 72.10 | – |
| | | DepGraph [41] | −0.24 | 52.40 | – |
| | | FPGM [10] | 0.10 | 52.60 | 50.60 |
| | | DCP [34] | 0.31 | 49.80 | – |
| | | CUP-SS [26] | 0.31 | 52.83 | – |
| | | FTWTJ [68] | 1.38 | 54.00 | – |
| | | GRDP [40] | 0.38 | 41.96 | 18.93 |
| | | Li et al. [28] | 0.18 | 49.90 | 44.00 |
| | | FTWTD [68] | 1.03 | 66.00 | – |
| | | Ours ($\tau = 0.75$) | −0.79 | 16.20 | 2.68 |
| | | Ours ($\tau = 0.7$) | −0.22 | 24.93 | 7.34 |
| | | **Ours ($\tau = 0.65$)** | **−0.03** | **50.10** | **24.46** |
| | | Ours ($\tau = 0.6$) | 4.49 | 78.91 | 66.44 |
| | CIFAR-100 | $\ell$1-norm [8] | 1.37 | 47.02 | 17.50 |
| | | Li et al. [28] | 0.84 | 47.02 | 17.50 |
| | | GRDP [40] | 0.71 | 47.02 | 17.50 |
| | | Ours ($\tau = 0.75$) | −0.04 | 30.84 | 9.31 |
| | | Ours ($\tau = 0.7$) | 1.38 | 47.17 | 17.61 |
| | | **Ours ($\tau = 0.65$)** | **2.10** | **60.30** | **32.53** |

**ResNet-110.** The pruning results for ResNet-110, presented in Table 4, further validate the scalability and robustness of our method across deeper network architectures. On CIFAR-10, our approach achieves competitive compression rates across multiple operating points, with FLOPs reductions ranging from 36.32% at $\tau = 0.75$ to 62.51% at $\tau = 0.65$. This demonstrates the method's ability to adaptively balance compression intensity and accuracy preservation. Specifically, at $\tau = 0.65$, our method reduces FLOPs by 62.51% with a corresponding accuracy drop of 2.68%, offering performance on par with leading methods such as FPRG (62.70% FLOPs reduction, 1.67% accuracy drop) while providing a tunable compression mechanism via an interpretable threshold. On CIFAR-100, our method exhibits particularly strong performance at $\tau = 0.7$, achieving a 52.57% reduction in FLOPs with only a 0.96% decrease in Top-1 accuracy. This result compares favorably against both L1-norm (52.42% FLOPs reduction, 1.07% accuracy drop) and Li et al. (52.42% FLOPs reduction, 0.68% accuracy drop), underscoring the advantage of our entropy-guided selection in preserving discriminative features on more complex datasets. Even at a higher compression setting ($\tau = 0.65$), the method attains a 64.74% FLOPs reduction with a 2.50% accuracy drop, demonstrating a consistent trade-off between compression and performance across varying intensities. These results highlight the method's effectiveness in handling deeper networks like ResNet-110, where structural complexity and inter-layer dependencies pose significant challenges to pruning. The integration of k-core decomposition with information entropy ensures that both redundancy elimination and information retention are optimally balanced, leading to superior compression performance without substantial accuracy degradation.

**Table 4:** Pruning results of ResNet-110 on CIFAR-10 and CIFAR-100, and $\tau$ is the setting of the threshold. Top-1 Acc. ↓ is Top-1 Accuracy loss after pruning (the lower is better). The higher the FR and PR, the better

| Model | Dataset | Method | Top-1 Acc. ↓ (%) | FR (%) | PR (%) |
|---|---|---|---|---|---|
| ResNet-110 | CIFAR-10 | MIL [69] | 0.19 | 34.20 | – |
| | | Zhao et al. [61] | 0.25 | 36.44 | 41.27 |
| | | $\ell$1-norm [8] | 0.23 | 38.70 | 32.40 |
| | | NISP [66] | 0.18 | 43.78 | 43.25 |
| | | HRank [6] | −0.73 | 41.20 | 39.40 |
| | | SFP [65] | 0.30 | 40.80 | – |
| | | GRDP [40] | 0.44 | 45.54 | 15.32 |
| | | Ours ($\tau = 0.75$) | 1.16 | 36.32 | 8.88 |
| | | Ours ($\tau = 0.7$) | 1.49 | 48.40 | 17.17 |
| | | **Ours ($\tau = 0.65$)** | **2.68** | **62.51** | **32.62** |
| | CIFAR-100 | Li et al. [28] | 0.68 | 52.42 | 20.72 |
| | | $\ell$1-norm [8] | 1.07 | 52.42 | 20.72 |
| | | GRDP [40] | 1.95 | 63.08 | 46.78 |
| | | Ours ($\tau = 0.75$) | 0.87 | 38.82 | 12.00 |
| | | Ours ($\tau = 0.7$) | 0.96 | 52.57 | 21.74 |
| | | **Ours ($\tau = 0.65$)** | **2.50** | **64.74** | **37.39** |

**DenseNet-40.** The pruning results for DenseNet-40 on both CIFAR-10 and CIFAR-100 are presented in Table 5, demonstrating the effectiveness and adaptability of our method in handling the complex connectivity inherent in dense networks. On CIFAR-10, our approach achieves a flexible trade-off between compression and accuracy by varying the similarity threshold $\tau$. At $\tau = 0.75$, the method reduces FLOPs by 38.01% with a minimal accuracy drop of only 0.29%, indicating conservative yet effective pruning. When $\tau$ is lowered to 0.7, a more aggressive compression is achieved with 51.34% FLOPs reduction and a negligible accuracy loss of 0.28%, significantly outperforming both Li et al. (43.09% FLOPs reduction, 1.19% accuracy drop) and Zhao et al. (44.78% FLOPs reduction, 0.95% drop). This result highlights the advantage of our entropy-guided selection in preserving informative features while eliminating redundancy. At the most aggressive setting ($\tau = 0.65$), the method attains a remarkable 76.45% reduction in FLOPs at the cost of a 1.55% accuracy decrease, demonstrating its capability for high compression rates while maintaining reasonable model performance. This performance is competitive with GRDP (74.51% FLOPs reduction, 0.94% accuracy drop), yet offers finer control through an interpretable threshold mechanism. On CIFAR-100, a similar trend is observed, though with a slightly higher accuracy degradation due to the increased complexity of the dataset. At $\tau = 0.75$, our method reduces FLOPs by 20.67% with only a 0.73% accuracy drop, providing a conservative pruning option suitable for scenarios where accuracy preservation is critical. When $\tau$ is set to 0.7, the FLOPs reduction increases to 43.96% with a 1.27% accuracy loss, outperforming $\ell_1$-norm (41.87% FLOPs reduction, 0.96% drop) and Li et al. (similar FLOPs reduction but 0.78% drop) in terms of compression efficiency. At $\tau = 0.65$, the method achieves a high compression rate of 68.56% FLOPs reduction, albeit with a more significant accuracy drop of 2.48%. This result is comparable to GRDP (67.08% FLOPs reduction, 1.95% drop), confirming the robustness of our approach under aggressive pruning conditions. The consistent performance across both datasets underscores the ability of our graph-entropy framework to effectively navigate the dense connectivity pattern, balancing compression and accuracy through a tunable threshold.

**Table 5:** Pruning results of DenseNet-40 on CIFAR-10 and CIFAR-100, and $\tau$ is the setting of the threshold. Top-1 Acc. ↓ is Top-1 Accuracy loss after pruning (the lower is better). The higher the FR and PR, the better

| Model | Dataset | Method | Top-1 Acc. ↓ (%) | FR (%) | PR (%) |
|---|---|---|---|---|---|
| DenseNet-40 | CIFAR-10 | GAL-0.01 [22] | 0.52 | 35.13 | 35.58 |
| | | Li et al. [28] | 1.19 | 43.09 | 24.03 |
| | | GRDP [40] | 0.94 | 74.51 | 66.41 |
| | | Zhao et al. [61] | 0.95 | 44.78 | 59.67 |
| | | HRank [6] | 1.13 | 60.94 | 53.85 |
| | | Ours ($\tau = 0.75$) | 0.29 | 38.01 | 18.05 |
| | | Ours ($\tau = 0.7$) | 0.28 | 51.34 | 36.18 |
| | | **Ours ($\tau$ = 0.65)** | **1.55** | **76.45** | **69.06** |
| | CIFAR-100 | Zhao et al. [61] | 2.45 | 22.67 | 37.73 |
| | | Li et al. [28] | 0.78 | 41.87 | 18.22 |
| | | GRDP [40] | 1.95 | 67.08 | 46.78 |
| | | $\ell$1-norm [8] | 0.96 | 41.87 | 18.22 |
| | | Ours ($\tau = 0.75$) | 0.73 | 20.67 | 6.22 |
| | | Ours ($\tau = 0.7$) | 1.27 | 43.96 | 19.54 |
| | | **Ours ($\tau$ = 0.65)** | **2.48** | **68.56** | **46.71** |

**ResNet-50.** To evaluate the scalability of our method on larger and more complex datasets, we conducted pruning experiments on ResNet-50 using the ImageNet dataset. The results, summarized in Table 6, demonstrate that our method maintains competitive performance even when applied to large-scale visual recognition tasks. At a moderate pruning intensity ($\tau = 0.7$), our approach reduces FLOPs by 50.48% with a corresponding Top-1 accuracy drop of 2.27% and Top-5 accuracy drop of 1.13%. This performance is comparable to established methods such as FPGM (42.20% FLOPs reduction, 0.65% Top-1 accuracy drop) while achieving higher compression rates. When employing more aggressive pruning ($\tau = 0.65$), our method attains a substantial 73.65% reduction in FLOPs with a 6.38% decrease in Top-1 accuracy. This result significantly outperforms GAL-1-joint (72.86% FLOPs reduction, 6.84% Top-1 accuracy drop) and is competitive with HRank (76.04% FLOPs reduction, 7.05% Top-1 accuracy drop), demonstrating the method's effectiveness in high-compression scenarios. These results on ImageNet validate that our graph-entropy collaborative framework generalizes well beyond smaller datasets like CIFAR, maintaining its capability to achieve favorable compression-accuracy trade-offs in challenging real-world applications.

**Table 6:** Pruning results of ResNet-50 on ImageNet dataset, and $\tau$ is the setting of the threshold. Top-1 Acc. ↓ and Top-5 Acc.↓ are Top-1 Accuracy loss and Top-5 Accuracy loss after pruning (the lower is better). The higher the FR and PR, the better

| Method | Top-1 Acc.↓ (%) | Top-5 Acc.↓ (%) | FR (%) | PR (%) |
|---|---|---|---|---|
| SSS-32 [62] | 1.97 | 0.96 | 31.05 | 27.06 |
| He et al. [20] | 3.88 | 2.07 | 33.25 | – |
| ThiNet-70 [46] | 0.84 | 0.47 | 36.79 | 33.72 |
| SFP [65] | 1.54 | 0.81 | 41.80 | – |
| FPGM [10] | 0.65 | 0.24 | 42.20 | – |
| GAL-0.5 [22] | 4.20 | 1.93 | 43.03 | 16.86 |
| SSS-26 [62] | 4.33 | 2.08 | 43.03 | 38.82 |
| SRR-GR [27] | 0.37 | 0.19 | 44.10 | – |
| Ours ($\tau = 0.7$) | 2.27 | 1.13 | 50.48 | 20.06 |
| CUP-SS [26] | 1.47 | 0.88 | 54.54 | – |
| GAL-1 [22] | 6.27 | 3.12 | 61.37 | 42.47 |
| GDP-0.5 [60] | 6.57 | 2.73 | 61.61 | – |
| GAL-1-joint [22] | 6.84 | 3.75 | 72.86 | 59.96 |
| Ours ($\tau = 0.65$) | 6.38 | 3.53 | 73.65 | 48.70 |
| HRank [6] | 7.05 | 3.29 | 76.04 | 67.57 |

### *4.3 Efficiency Comparison*

To evaluate the overall efficiency of the proposed method, we compared it against several state-of-the-art pruning schemes under the CIFAR-10 + ResNet-56 + RTX 3090 setup, as summarized in Table 7 and Fig. 4. The results demonstrate that our approach achieves a favorable balance among time cost, compression rate, and accuracy preservation.

**Table 7:** Efficiency comparison of different pruning schemes on CIFAR-10 + ResNet-56. Top-1 Acc. ↓ isTop-1 Accuracy loss after pruning (the low is better)

| Method | Pruning time (s) | Fine-tuning epochs | Total time (s) | FLOPs ↓ (%) | Top-1 Acc↓(%) |
|---|---|---|---|---|---|
| Ours | 486 | 70 | 556 | 59.8 | 0.83 |
| NSB [70] | ∼600 | 150 | ∼750 | 51.0 | 0.26 |
| Hrank [6] | 600 | 300 | 900 | 53.6 | 0.53 |
| ResRep [71] | ∼720 | 120 | ∼840 | 52.9 | 0.00 |
| SFP [65] | ∼120 | 100 | ∼220 | 52.6 | 1.33 |
| FPGM [10] | ∼90 | 100 | ∼190 | 52.6 | 0.66 |
| AMC [72] | ∼900 | 200 | ∼1100 | 50.0 | 0.9 |
| GRDP [40] | 468 | 70 | 538 | 57.1 | 0.84 |

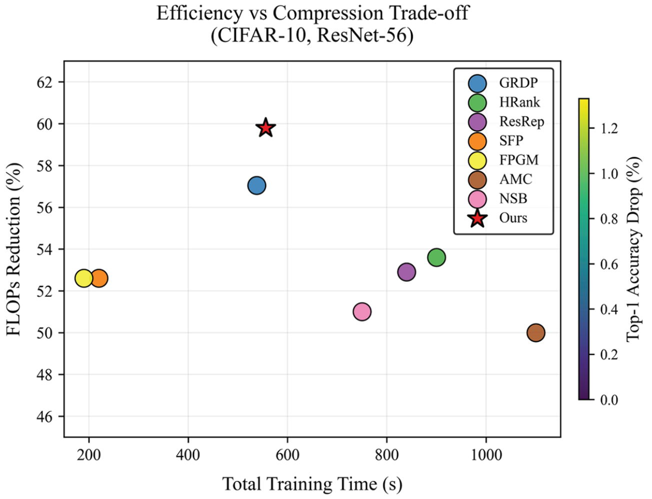

**Figure 4:** Compression performance comparison

The total processing time of our method is 556 s, which is longer than that of FPGM (190 s) and SFP (220 s), yet it delivers a substantially higher FLOPs reduction (59.8% vs. 52.6%) while maintaining a competitive accuracy drop of only 0.83%—lower than SFP (1.33%) and AMC (0.90%), and comparable to HRank (0.53%). Although the introduction of graph construction and entropy computation incurs a moderate time overhead compared to GRDP (486 vs. 468 s), it significantly reduces accuracy degradation under aggressive pruning settings (e.g., 0.03% vs. 0.84% at $\tau = 0.65$), highlighting the benefit of entropy-guided selection in retaining discriminative filters.

Compared to methods requiring extended fine-tuning, such as HRank and AMC, our approach completes pruning and recovery in only 70 fine-tuning epochs, reducing the total time cost by approximately 40% while achieving higher compression rates. This efficiency makes our method particularly suitable for deployment on resource-constrained edge devices. In the three-dimensional performance space of time, compression, and accuracy, our method occupies a Pareto-optimal region, offering high compression and low accuracy loss at a moderate computational cost.

### *4.4 Ablation Studies*

**Selection Criterion.** To systematically evaluate the contribution of each component in our proposed framework, we conducted comprehensive ablation studies on multiple architectures, including VGGNet-16, ResNet-56, and DenseNet-40, using the CIFAR-100 dataset. Three filter selection strategies were evaluated: (1) entropy-only pruning, where filters are retained or removed solely based on the information entropy of their output feature maps; (2) k-core-only pruning, which constructs a redundancy graph and randomly preserves one filter from each k-core subgraph; and (3) the combined k-core & entropy approach proposed in this work, which integrates graph-based redundancy decomposition with information-theoretic filter importance scoring. The comparative results demonstrate the effectiveness of the hybrid strategy in achieving a superior balance between model compression and accuracy retention.

All experiments were conducted with a fixed similarity threshold $\tau = 0.7$ and repeated five times to ensure statistical reliability. The results, including mean performance and standard deviation, are summarized in Fig. 5, the proposed hybrid method ($k$-core & Entropy) consistently outperforms the other two strategies across all architectures, achieving the highest compression rates (FLOPs and parameter reduction) while maintaining the lowest accuracy degradation. In contrast, the Entropy-only method preserves accuracy reasonably well but yields limited compression due to its neglect of structural redundancy. The $k$-core only approach achieves higher compression but suffers from greater accuracy loss and higher variance, attributable to its random selection within redundant clusters.

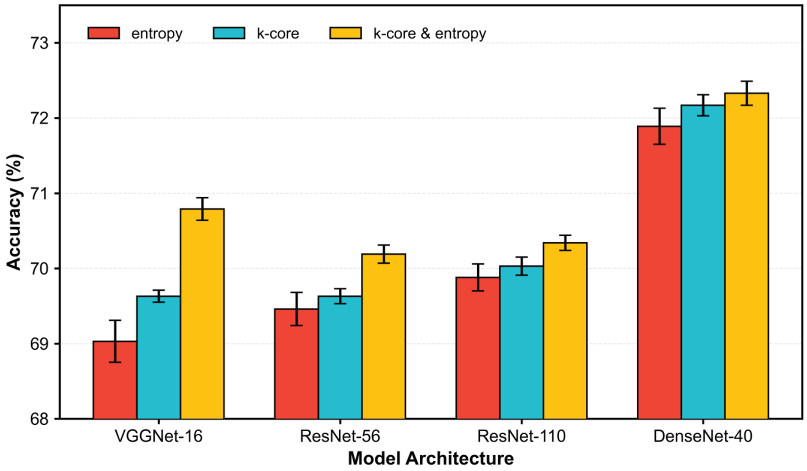

**Figure 5:** Performance comparison

Notably, the hybrid method demonstrates particularly strong performance on VGGNet-16 and ResNet-56, where it achieves a favorable balance between compression and accuracy. For DenseNet-40, all methods exhibit a slightly higher accuracy drop, reflecting the inherent challenge of pruning densely connected architectures. However, our method still delivers the most stable and robust results, with the smallest standard deviation, confirming its effectiveness in preserving informative features while eliminating redundancy.

These results underscore the importance of combining both structural redundancy analysis and information-theoretic importance scoring. The graph-based $k$-core decomposition effectively identifies redundant groups, while the information entropy provides a reliable measure of the informational value of each filter, together enabling more informed and stable pruning decisions across diverse network architectures.

**Sample Size.** To systematically evaluate the impact of sample size on the pruning performance, we conducted a series of experiments in which the number of randomly selected images used for computing feature map similarity was varied from 25 to 200 in increments of 25. All models were fine-tuned on the CIFAR-100 dataset under identical hyperparameter settings. As summarized in Table 8, both FLOPs reduction (FR) and parameter reduction (PR) rates exhibit a gradual decline as the sample size increases. This behavior can be attributed to the reduced estimation error in average similarity with larger samples, leading to a more conservative pruning strategy and consequently lower compression rates. Interestingly, although larger sample sizes result in higher parameter retention, this does not consistently translate into improved accuracy. The Top-1 accuracy loss initially decreases with increasing sample size, reaching an optimum around a sample size of 100, beyond which the accuracy degradation tends to rise again across all architectures. This non-monotonic relationship suggests that an intermediate sample size strikes an optimal balance between estimation reliability and representativeness of the dataset. Excessively large samples not only diminish pruning efficiency but also increase computational overhead during the filter selection phase, without yielding gains in model accuracy.

**Table 8:** Pruning results of the models (VGGNet-16, ResNet-56/110, and DenseNet-40) on CIFAR-100 dataset with varying sample sizes (from 25 to 200 in steps of 25). Top-1 Acc. ↓ isTop-1 Accuracy loss after pruning (the low is better). The higher the FR and PR, the better

| Sample size | Model | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | VGGNet-16 | | | ResNet-56 | | | ResNet-110 | | | DenseNet-40 | | |
| | Top-1 acc. ↓ (%) | FR(%) | PR (%) | Top-1 acc. ↓ (%) | FR(%) | PR (%) | Top-1 acc. ↓ (%) | FR(%) | PR(%) | Top-1 acc. ↓ (%) | FR (%) | PR (%) |
| 25 | 2.71 | 65.12 | 83.48 | 1.31 | 48.23 | 18.71 | 1.69 | 53.51 | 21.83 | 1.25 | 44.38 | 18.63 |
| 50 | 2.76 | 63.29 | 81.35 | 1.10 | 46.91 | 17.62 | 1.36 | 52.89 | 21.67 | 0.71 | 43.29 | 18.51 |
| 75 | 2.44 | 63.31 | 81.03 | 0.77 | 46.15 | 17.53 | 0.95 | 52.69 | 20.75 | 0.70 | 42.61 | 18.21 |
| **100** | **1.71** | **62.49** | **79.93** | **0.67** | **47.05** | **17.47** | **0.64** | **52.45** | **20.69** | **0.55** | **41.84** | **18.19** |
| 125 | 1.72 | 61.56 | 79.33 | 0.78 | 46.20 | 17.45 | 0.84 | 52.68 | 20.71 | 0.80 | 42.71 | 18.51 |
| 150 | 2.34 | 60.92 | 78.97 | 0.74 | 43.97 | 16.20 | 0.79 | 52.29 | 20.34 | 1.11 | 41.44 | 17.00 |
| 175 | 2.37 | 61.29 | 79.12 | 0.77 | 42.93 | 16.03 | 1.19 | 52.13 | 20.42 | 0.94 | 40.27 | 16.42 |
| 200 | 2.27 | 61.20 | 79.11 | 0.88 | 42.86 | 15.46 | 1.04 | 52.10 | 20.32 | 1.16 | 40.96 | 16.46 |

**Learning rate.** We investigated the impact of the learning rate on fine-tuning outcomes. All models were fine-tuned on the CIFAR-100 dataset with different initial learning rates. We evaluated four values: 0.1, 0.01, 0.001, and 0.0001, while keeping all other hyperparameters constant. The accuracy curves throughout fine-tuning are presented in Fig. 6. The results reveal a typical "U-shaped" convergence behavior across all networks. With a learning rate of 0.1, the models exhibited significant early-stage oscillation and a sharp accuracy drop within the first 20 epochs. A slight recovery occurred later, but the final performance remained suboptimal. A learning rate of 0.01 yielded the most stable training process, entering a steady ascent after approximately 10 epochs and nearing convergence by epoch 30, ultimately achieving the highest or near-highest Top-1 accuracy. This validates 0.01 as a suitable learning rate for fine-tuning pruned models. When the learning rate was reduced to 0.001, the models started with relatively high initial accuracy but converged slowly, failing to reach full convergence even after 70 epochs, which indicates insufficient gradient update strength. A learning rate of 0.0001 caused training to nearly stagnate, resulting in fine-tuning efficacy that was only marginally better than random initialization. Based on convergence speed, stability, and final accuracy, we selected a learning rate of 0.01 for all subsequent experiments.

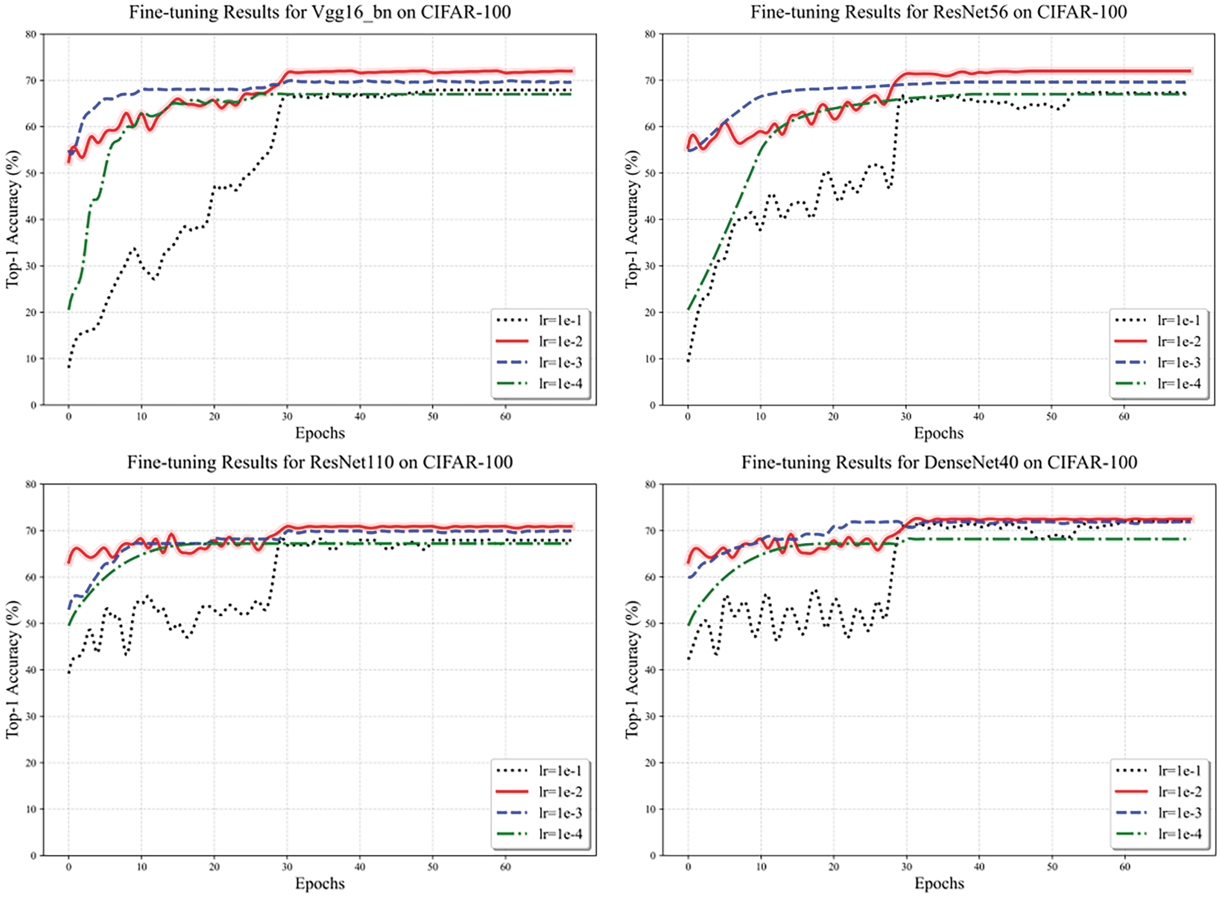

**Figure 6:** Accuracy change curves

### 4.5 *Discussion and Future Work*

Our proposed "graph-entropy collaborative pruning" framework offers compelling benefits in three key aspects: compression rate, accuracy retention, and computational efficiency. By synergistically integrating k-core decomposition for redundancy elimination and information entropy for feature preservation, our method achieves high compression rates (e.g., up to 76.45% FLOPs and 88.82% parameter reduction) with minimal accuracy loss (as low as 0.03% on ResNet-56), significantly outperforming state-of-the-art methods like L1-norm, FTWT, and HRank. The architecture-specific strategies and the efficient "one-shot pruning + short-term fine-tuning" pipeline further ensure the practical deployability of our approach, reducing total time cost by over 60% compared to methods requiring prolonged retraining.

Notwithstanding these strengths, we identify several limitations that point to promising future research directions:

**Adaptive Thresholding for Scalability.** The current global similarity threshold $\tau$, while effective for CIFAR-scale datasets, offers limited granularity for controlling pruning intensity across layers in larger, more heterogeneous networks (e.g., on ImageNet). Future work will develop adaptive $\tau$ scheduling strategies, such as layer-wise or block-wise thresholding, to achieve finer-grained compression control and maintain robustness on complex datasets.

**Extension Beyond Convolutional Layers.** Our method currently prunes only convolutional layers, leaving parameter-heavy linear layers (e.g., in classification heads) and emerging components like self-attention modules untouched. To maximize the impact on modern architectures (e.g., Vision Transformers, CNN-Transformer hybrids), we plan to generalize the graph–entropy framework to linear and attention-based layers, enabling whole-model compression.

**Handling Complex Connectivity Patterns.** The k-core decomposition algorithm assumes a relatively uniform connectivity structure. In highly fragmented or attention-augmented graphs, this may cap the achievable redundancy reduction. Investigating alternative graph-theoretic measures or hierarchical community detection algorithms could enhance redundancy identification in these challenging scenarios.

**Computational Overhead and Scalability.** The graph construction phase, reliant on perceptual hashing (pHash), introduces a one-time overhead. While this cost is acceptable for the models and datasets studied here (Tables 1 and 7) given the substantial gains in compression and accuracy with reduced fine-tuning, it may become a bottleneck for very high-resolution inputs or extremely large models. To enhance scalability, future work will explore strategies such as approximate & hierarchical graph construction (e.g., via sampling or lower-fidelity feature representations) and in-training redundancy estimation, where the similarity calculation is integrated into the training process to incrementally build the graph with minimal overhead.

Addressing these aspects will further solidify the generality and effectiveness of our approach for next-generation efficient deep learning models.

## 5 Conclusion

In this study, we proposed a novel graph-entropy collaborative framework for filter pruning in convolutional neural networks (CNNs). The core contribution is a Pareto-optimal balancing mechanism that simultaneously maximizes structural redundancy removal via k-core decomposition while minimizing information loss through entropy-based filter scoring. This approach enables effective compression of CNNs, making them suitable for deployment in resource-constrained environments.

Specifically, our method constructs an intra-layer redundancy graph using perceptual hashing to quantify feature map similarity. Through adaptive k-core decomposition, high-density redundant substructures are identified hierarchically. Information entropy is then employed to evaluate the informational value of filters within each cluster, ensuring that only the most informative nodes are retained. This two-stage process guarantees the removal of filters that are redundant both structurally and functionally, while preserving discriminative features.

Comprehensive experiments on CIFAR-10 and CIFAR-100 demonstrate the efficacy and generality of the proposed framework. Across multiple architectures—including VGGNet-16, ResNet-56/110, and DenseNet-40—our method achieved significant compression rates, with average reductions of

42.3% in FLOPs and 76.8% in parameters, accompanied by only a 0.63% average decrease in Top-1 accuracy. Notably, on ResNet-56 with $\tau = 0.65$, we attained a 50.10% reduction in FLOPs with a negligible accuracy loss of 0.03%, substantially outperforming existing methods such as FTWT (54.00% FLOPs reduction, 1.38% accuracy loss). Compared to pruning strategies based solely on importance metrics (e.g., L1-norm) or redundancy criteria (e.g., GRDP), our framework achieves a superior balance between compression and accuracy retention. The incorporation of architecture-specific constraints—such as synchronous pruning in residual connections and conservative thresholds in dense blocks—ensures functional integrity across diverse network topologies. The method's effectiveness extends to large-scale tasks, as evidenced by the ResNet-50 results on ImageNet, where it achieved a high compression rate of 73.65% FLOPs reduction, confirming its scalability beyond small-scale datasets. Furthermore, the one-shot pruning pipeline followed by 70 epochs of fine-tuning reduces the total time cost to approximately 40% of that required by iterative pruning methods like HRank, striking an efficient trade-off among compression, accuracy, and computational overhead.

Despite these promising results, certain limitations remain. Under high pruning intensities (e.g., $\tau = 0.65$), DenseNet-40 exhibits a slightly higher accuracy degradation, indicating a need for further refinement in dynamic threshold adaptation. Moreover, the current method is applied only to convolutional layers; future work will extend it to linear and attention-based layers, and validate its scalability on larger datasets such as ImageNet and Transformer-CNN hybrid architectures. Integrating advanced techniques like knowledge distillation or reinforcement learning may further enhance the generalization and convergence of pruned models.

In summary, this work presents a principled and efficient framework that unifies structural redundancy analysis with information-theoretic filtering, offering a practical and effective solution for compressing CNNs in edge-device applications.

**Acknowledgement:** Not applicable.

**Funding Statement:** This paper is supported by the National Natural Science Foundation of China (NSFC) (No. 61902158).

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Yuan Yang, Dengbiao Jiang; data collection: Yuan Yang, Xing Deng; analysis and interpretation of results: Dengbiao Jiang, Lijuan Wang; draft manuscript preparation: Yuan Yang, Dengbiao Jiang. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The original CIFAR-10 and CIFAR-100 datasets are publicly available from the official source at https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 16 September 2025).

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Hasan MA, Bhargav T, Sandeep V, Reddy VS, Ajay R. Image classification using convolutional neural networks. Int J Mech Eng Res Technol. 2024;16(2):173–81. doi:10.1109/ICEEICT53079.2022.9768622.

2. Naseer A, Mudawi NA, Abdelhaq M, Alonazi M, Alazeb A, Algarni A, et al. CNN-based object detection via segmentation capabilities in outdoor natural scenes. IEEE Access. 2024;12(2):84984–5000. doi:10.1109/ACCESS.2024.3413848.
3. Saika MH, Avi SP, Islam KT, Tahmina T, Abdullah MS, Imam T. Real-time vehicle and lane detection using modified overfeat cnn: a comprehensive study on robustness and performance in autonomous driving. J Comput Sci Technol Stud. 2024;6(2):30–36. doi:10.32996/jcsts.2024.6.2.4.
4. Xie C, Zhai X, Chi H, Li W, Li X, Sha Y, et al. A novel fusion pruning-processed lightweight CNN for local object recognition on resource-constrained devices. IEEE Trans Consum Electron. 2024;70(4):6713–24. doi:10.1109/TCE.2024.3475517.
5. Dantas PV, Silva WSD, Cordeiro LC, Carvalho CB. A comprehensive review of model compression techniques in machine learning. Appl Intell. 2024;54(22):11804–44. doi:10.1007/s10489-024-05747-w.
6. Lin M, Ji R, Wang Y, Zhang Y, Zhang B, Tian Y, et al. HRank: filter pruning using high-rank feature map. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 1529–38. doi:10.1109/CVPR42600.2020.00160.
7. Geng L, Niu B. Apssf: adaptive cnn pruning based on structural similarity of filters. Int J Comput Intell Syst. 2024;17(1):129. doi:10.1007/s44196-024-00518-4.
8. Li H, Kadav A, Durdanovic I, Samet H, Graf HP Pruning filters for efficient convnets. arXiv:1608.08710. 2016. doi:10.48550/arXiv.1608.08710.
9. Lee E, Hwang Y. Amap: automatic multihead attention pruning by similarity-based pruning indicator. IEEE Trans Neural Netw Learn Syst. 2025;1–14. doi:10.1109/TNNLS.2025.3606750.
10. He Y, Liu P, Wang Z, Hu Z, Yang Y. Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 4340–9. doi:10.1109/CVPR.2019.00447.
11. Pei J, Huang Z, Zhu J. Pruning rate-controlled filter order-information structure similarity graph clustering for dcnn structure optimization methods. Multimed Tools Appl. 2024;83(32):78407–31. doi:10.1007/s11042-024-18615-z.
12. Batagelj V, Zaversnik M. An O(m) algorithm for cores decomposition of networks. arXiv:cs/0310049. 2003. doi:10.48550/arXiv.cs/0310049.
13. Hur C, Kang S. Entropy-based pruning method for convolutional neural networks. J Supercomput. 2019;75(6):2950–63. doi:10.1007/s11227-018-2684-z.
14. Valverde JM, Shatillo A, Tohka J. Sauron U-Net: simple automated redundancy elimination in medical image segmentation via filter pruning. Neurocomputing. 2024;594(6):127817. doi:10.1016/j.neucom.2024.127817.
15. Wang W, Yu Z, Fu C, Cai D, He X. COP: customized correlation-based filter level pruning method for deep CNN compression. Neurocomputing. 2021;464:533–45. doi:10.1016/j.neucom.2021.08.098.
16. Zawish M, Davy S, Abraham L. Complexity-driven model compression for resource-constrained deep learning on edge. IEEE Trans Artif Intell. 2024;5(8):3886–901. doi:10.1109/TAI.2024.3353157.
17. Pham N-S, Shin S, Xu L, Shi W, Suh T. Cross-filter structured pruning for efficient sparse cnn acceleration. IEEE Access. 2025;13:129461–75. doi:10.1109/ACCESS.2025.3587027.
18. Bibi U, Mazhar M, Sabir D, Butt MFU, Hassan A, Ghazanfar MA, et al. Advances in pruning and quantization for natural language processing. IEEE Access. 2024;12:139113–28. doi:10.1109/ACCESS.2024.3465631.
19. Kallakuri U, Humes E, Mohsenin T. Resource-aware saliency-guided differentiable pruning for deep neural networks. In: Proceedings of the Great Lakes Symposium on VLSI 2024; 2024. p. 694–9. doi:10.1145/3649476.3658699.
20. He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 1389–97. doi:10.1109/ICCV.2017.155.

21. Li X, Gong J, Lv H, Wen J, Liu K, Wang Z. Convolution kernel pruning algorithm based on average percentage of zeros and data distribution similarity. In: 2024 IEEE International Conference on Unmanned Systems (ICUS). IEEE; 2024. p. 1260–5. doi:10.1109/ICUS61736.2024.10840150.
22. Lin S, Ji R, Yan C, Zhang B, Cao L, Ye Q, et al. Towards optimal structured cnn pruning via generative adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 2790–9. doi:10.1109/CVPR.2019.00290.
23. Gao X, Zhao Y, Dudziak Ł, Mullins R, Xu C-Z. Dynamic channel pruning: feature boosting and suppression. arXiv:1810.05331. 2018. doi:10.48550/arXiv.1810.05331.
24. Sharma M, Heard J, Saber E, Markopoulos PP. Convolutional neural network compression via dynamic parameter rank pruning. IEEE Access. 2025;13:18441–56. doi:10.1109/ACCESS.2025.3533419.
25. Younesi A, Ansari M, Fazli M, Ejlali A, Shafique M, Henkel J. A comprehensive survey of convolutions in deep learning: applications, challenges, and future trends. IEEE Access. 2024;12(2):41180–218. doi:10.1109/ACCESS.2024.3376441.
26. Duggal R, Xiao C, Vuduc R, Chau DH, Sun J. CUP: cluster pruning for compressing deep neural networks. In: 2021 IEEE International Conference on Big Data (Big Data). IEEE; 2021. p. 5102–6. doi:10.1109/BigData52589.2021.9671980.
27. Wang Z, Li C, Wang X. Convolutional neural network pruning with structural redundancy reduction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 14913–22. doi:10.1109/CVPR46437.2021.01467.
28. Li J, Shao H, Zhai S, Jiang Y, Deng X. A graphical approach for filter pruning by exploring the similarity relation between feature maps. Pattern Recognit Lett. 2023;166(12):69–75. doi:10.1016/j.patrec.2022.12.028.
29. Amirabadi MA, Nezamalhosseini SA, Kahaei MH, Chen LR. A comprehensive survey on machine and deep learning for optical communications. IEEE Access. 2025;13(1):88794–846. doi:10.1109/ACCESS.2025.3569559.
30. Wang R, Zuo P, Fu X, Yang H, Liu Y, Zhang L, et al. Structure characteristic-aware pruning strategy for convolutional neural networks. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE; 2019. p. 1533–40. doi:10.1109/HPCC/SmartCity/DSS.2019.00211.
31. Shannon CE. A mathematical theory of communication. Bell Syst Tech J. 1948;27(3):379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.
32. Shi C, Hao Y, Li G, Xu S. Vngep: filter pruning based on von neumann graph entropy. Neurocomputing. 2023;528(1):113–24. doi:10.1016/j.neucom.2023.01.046.
33. Luo J-H, Wu J. An entropy-based pruning method for CNN compression. arXiv:1706.05791. 2017. doi:10.48550/arXiv.1706.05791.
34. Mohammad Amini S, Abbasi-Moghadam D, Sharifi A. Classification of tomato plant leaf disease with entropy filter and convolutional neural network. CABI Agricul Biosci. 2025;6(1):0050. doi:10.1079/ab.2025.0050.
35. Lu Y, Guan Z, Yang Y, Zhao W, Gong M, Xu C. Entropy induced pruning framework for convolutional neural networks. Proc AAAI Conf Artif Intell. 2024;38(4):3918–26. doi:10.1609/aaai.v38i4.28184.
36. Ozdemir C. Adapting transfer learning models to dataset through pruning and Avg-TopK pooling. Neural Comput Appl. 2024;36(11):6257–70. doi:10.1007/s00521-024-09484-6.
37. Mahmoodi J, Abbasi-Moghadam D, Sharifi A, Nezamabadi-Pour H, Esmaeili M, Vafaeinejad A. Dessanet model: hyperspectral image classification using an entropy filter with spatial and spectral attention modules on deepnet. IEEE J Sel Top Appl Earth Obs Remote Sens. 2024;17:14588–613. doi:10.1109/JSTARS.2024.3439592.
38. Lu Y, Hu Z, Zhao W, Guan Z, Gong M, Zhang M. Layer-interaction adaptive pruning for remote sensing scene classification. IEEE Trans Geosci Remote Sens. 2025;63:1–12. doi:10.1109/TGRS.2025.3570653.

39. Wang M, Zhang Q, Yang J, Cui X, Lin W. Graph-adaptive pruning for efficient inference of convolutional neural networks. arXiv:1811.08589. 2018. doi:10.48550/arXiv.1811.08589.
40. Li J, Shao H, Deng X, Jiang Y. Efficient filter pruning: reducing model complexity through redundancy graph decomposition. Neurocomputing. 2024;599(4):128108. doi:10.1016/j.neucom.2024.128108.
41. Fang G, Ma X, Song M, Mi MB, Wang X. DepGraph: towards any structural pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 16091–101. doi:10.48550/arXiv.2301.12900.
42. Li Z, Zuo X, Song Y, Liang D, Xie Z. A multi-agent reinforcement learning based approach for automatic filter pruning. Sci Rep. 2024;14(1):31193. doi:10.1038/s41598-024-82562-w.
43. Faqir N, Ennaji Y, Chakir L, Boumhidi J. Hybrid CNN-LSTM and proximal policy optimization model for traffic light control in a multi-agent environment. IEEE Access. 2025;13(1):29577–88. doi:10.1109/ACCESS.2025.3541042.
44. Medhat S, Abdel-Galil H, Aboutabl AE, Saleh H. Iterative magnitude pruning-based light-version of AlexNet for skin cancer classification. Neural Comput Appl. 2024;36(3):1413–28. doi:10.1007/s00521-023-09111-w.
45. Geng X, Gao J, Zhang Y, Xu D. Complex hybrid weighted pruning method for accelerating convolutional neural networks. Sci Rep. 2024;14(1):5570. doi:10.1038/s41598-024-55942-5.
46. Luo J-H, Wu J, Lin W. ThiNet: a filter level pruning method for deep neural network compression. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 5058–66. doi:10.48550/arXiv.1707.06342.
47. Enderich L, Timm F, Burgard W. Holistic filter pruning for efficient deep neural networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2021. p. 2596–605. doi:10.48550/arXiv.2009.08169.
48. Lee N, Ajanthan T, Torr PH. SNIP: single-shot network pruning based on connection sensitivity. arXiv:1810.02340. 2018. doi:10.48550/arXiv.1810.02340.
49. Klein A, Golebiowski J, Ma X, Perrone V, Archambeau C. Structural pruning of pre-trained language models via neural architecture search. arXiv:2405.02267. 2024. doi:10.48550/arXiv.2405.02267.
50. Li G, Shao H, Deng X, Jiang Y. Adaptive convolutional network pruning through pixel-level cross-correlation and channel independence for enhanced model compression. Eng Appl Artif Intell. 2025;154(12):110920. doi:10.1016/j.engappai.2025.110920.
51. Zniyed Y, Nguyen TP. Nguyen etal, Coupled tensor decomposition for compact network representation. IEEE Trans Neural Netw Learn Syst. 2025;1–15. doi:10.1109/TNNLS.2025.3609797.
52. Li S, Chen J, Xiang J, Zhu C, Liu Y. AutoDFP: automatic data-free pruning via channel similarity reconstruction. arXiv:2403.08204. 2024. doi:10.48550/arXiv.2403.08204.
53. Williams M, Aletras N. On the impact of calibration data in post-training quantization and pruning. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2024. p. 10100–18. doi:10.18653/v1/2024.acl-long.544.
54. Gurevin D, Shan M, Huang S, Hasan MA, Ding C, Khan O. PruneGNN: algorithm-architecture pruning framework for graph neural network acceleration. In: 2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA). IEEE; 2024. p. 108–23. doi:10.1109/HPCA57654.2024.00019.
55. Raffel M, Renjith A, Chen L. MetaCluster: enabling deep compression of kolmogorov-arnold network. arXiv:2510.19105. 2025. doi:10.48550/arXiv.2510.19105.
56. Bhuiyan SB, Adib MSH, Bhuiyan MA, Kabir MR, Farazi M, Rahman S, et al. Z-Pruner: post-training pruning of large language models for efficiency without retraining. arXiv:2508.15828. 2025. doi:10.48550/arXiv.2508.15828.

57. Louati H, Louati A, Mansour K, Kariri E. Achieving faster and smarter chest X-ray classification with optimized CNNs. IEEE Access. 2025;13:10070–82. doi:10.1109/ACCESS.2025.3529206.

58. Liao Z, Qu'etu V, Nguyen V-T, Tartaglione E. NEPENTHE: entropy-based pruning as a neural network depth's reducer. arXiv:2404.16890. 2024. doi:10.48550/arXiv.2404.16890.

59. Yadulla AR, Konda B, Yenugula M, Kasula VK, Rakki SB, Banoth R. Lightweight neural networks for adversarial defense: a novel NTK-guided pruning approach. In: 2025 37th Conference of Open Innovations Association (FRUCT). IEEE; 2025. p. 331–7. doi:10.23919/FRUCT65909.2025.11008002.

60. Guo Y, Yuan H, Tan J, Wang Z, Yang S, Liu J. GDP: stabilized neural network pruning via gates with differentiable polarization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 5239–50. doi:10.48550/arXiv.2109.02220.

61. Zhao C, Ni B, Zhang J, Zhao Q, Zhang W, Tian Q. Variational convolutional neural network pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 2780–9. doi:10.1109/CVPR.2019.00289.

62. Huang Z, Wang N. Data-driven sparse structure selection for deep neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 304–20. doi:10.48550/arXiv.1707.01213.

63. Sui Y, Yin M, Xie Y, Phan H, Zonouz A, Yuan B. CHIP: channel independence-based pruning for compact neural networks. Adv Neural Inf Process Syst. 2021;34:24604–16. doi:10.48550/arXiv.2110.13981.

64. Liu Z, Li J, Shen Z, Huang G, Yan S, Zhang C. Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 2736–44. doi:10.48550/arXiv.1708.06519.

65. He Y, Kang G, Dong X, Fu Y, Yang Y. Soft filter pruning for accelerating deep convolutional neural networks. arXiv:1808.06866. 2018. doi:10.48550/arXiv.1808.06866.

66. Yu R, Li A, Chen C-F, Lai J-H, Morariu VI, Han X, et al. NISP: pruning networks using neuron importance score propagation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 9194–203. doi:10.48550/arXiv.1711.05908.

67. Li Y, Gemert JCVan, Hoefler T, Moons B, Eleftheriou E, Verhoef B-E. Differentiable transportation pruning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023. p. 16957–67. doi:10.48550/arXiv.2307.08483.

68. Elkerdawy S, Elhoushi M, Zhang H, Ray N. 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; 2022. p. 12454–63. doi:10.48550/arXiv.2110.08232.

69. Dong X, Huang J, Yang Y, Yan S. More is less: a more complicated network with less inference complexity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 5840–8. doi:10.48550/arXiv.1703.08651.

70. Ganesh MR, Blanchard D, Corso JJ, Sekeh SY. Slimming neural networks using adaptive connectivity scores. IEEE Trans Neural Netw Learn Syst. 2022;35(3):3794–808. doi:10.1109/TNNLS.2022.3198580.

71. Ding X, Hao T, Tan J, Liu J, Han J, Guo Y, et al. ResRep: lossless CNN pruning via decoupling remembering and forgetting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 4510–20. doi:10.48550/arXiv.2007.03260.

72. He Y, Lin J, Liu Z, Wang H, Li L-J, Han S. AMC: AutoML for model compression and acceleration on mobile devices. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 784–800. doi:10.1007/978-3-030-01234-248.