Indonesian Journal of Electrical Engineering and Computer Science
Vol. 43, No. 1, July 2026, pp. 314∼324
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v43.i1.pp314-324

Layer-wise adaptive structured pruning via genetic algorithms with taylor-based proxy fitness

Anh-Truong Vo[1,2], Hoang-Loc Tran[1,2], Dinh-Duy Phan[1,2], Duc-Lung Vu[1,2]
[1]Faculty of Computer Engineering, University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Feb 7, 2026
Revised Jun 6, 2026
Accepted Jun 27, 2026

Keywords:

Layer collapse
Model compression
Multi-objective optimization
Neural architecture search
NSGA-II
Structured pruning
Taylor expansion

ABSTRACT

Deploying deep convolutional neural networks (CNNs) on edge devices requires balancing model accuracy and computational efficiency. While structured pruning limits inference costs by removing redundant filters, most methods apply a rigid, global criterion, ignoring the distinct representational roles of individual layers. This yields suboptimal results, especially under aggressive compression where over-pruning degrades performance. To address this limitation, we propose an adaptive structured pruning framework based on genetic algorithms (GAs) that jointly optimizes layer-wise pruning ratios and strategies. Each layer independently selects between min-importance and median-rank pruning, enabling the exploration of tailored strategy combinations. A training-free taylor-based proxy fitness function ensures efficient candidate evaluation without repeated fine-tuning. After fine-tuning the selected architecture, experiments on VGG16 demonstrate that our method achieves $92.78 \pm 0.28\%$ accuracy (over 50 independent runs) with a $70.0 \pm 3.2\%$ MACs reduction on CIFAR-10, and maintains 71.82% accuracy on CIFAR-100. These results demonstrate competitive performance compared to existing pruning methods while achieving substantial computational cost reduction.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Duc-Lung Vu
Faculty of Computer Engineering, University of Information Technology
Ho Chi Minh City, Vietnam
Email: lungvd@uit.edu.vn

1. INTRODUCTION

In the era of ubiquitous artificial intelligence, deep convolutional neural networks (CNNs) drive state-of-the-art computer vision systems. However, popular architectures like VGG16 [1] and ResNet [2] demand billions of floating-point operations (FLOPs) and millions of parameters per inference. This mismatch between modern deep model complexity and the limited resources of edge device constitutes a critical deployment bottleneck [3]-[5].

To address these constraints, structured pruning (filter or channel pruning) has emerged as a prominent model compression technique. By physically eliminating redundant filters, it directly reduces memory footprint and inference latency on general-purpose hardware without specialized sparse computation support [6].

Despite its potential, current structured pruning faces two primary limitations. First, most methods utilize rigid heuristic strategies (e.g., L1-norm [7]) or uniform pruning ratios globally. This ignores diverse layer roles—shallow layers capture low-level textures, while deeper layers encode semantic abstractions—causing suboptimal feature preservation. Second, evaluation inefficiency plagues automated frameworks like AMC [8],

which require computationally prohibitive fine-tuning to evaluate candidates. Furthermore, aggressive global compression can completely eliminate filters in certain layers, causing layer collapse and emphasizing the need for adaptive pruning decisions.

To overcome the constraints of fixed global criteria, we address adaptive layer-wise strategy selection as a largely underexplored area. We propose an adaptive structured pruning framework based on genetic algorithms (GAs) [9]. Our method introduces a layer-wise hybrid strategy space, enabling each layer to independently select between min-importance pruning and median-rank pruning. The pruning process is formulated as single-objective and multi-objective optimization problems, explicitly balancing accuracy, MACs, and model size. To bypass costly fine-tuning, candidate architectures are evaluated using a training-free taylor-based proxy fitness function derived from pre-computed first-order importance scores.

The main contributions are:

- A novel layer-wise hybrid search space: we introduce a dual-variable space that jointly optimizes pruning ratios and strategies (min-importance vs. median-rank) per layer, unlike conventional methods confined to optimizing ratios under a fixed criterion.
- Topology-preserving constrained optimization: we frame pruning as a constrained multi-objective problem, enforcing structural constraints across the decoding and fitness evaluation phases to maintain topological connectivity and prevent layer collapse under aggressive compression.
- Adaptive strategy discovery: our framework autonomously identifies hybrid strategies balancing efficiency and feature preservation. Over 50 independent runs, the multi-objective variant achieved $92.78 \pm 0.28\%$ accuracy while reducing MACs by $70.0 \pm 3.2\%$.

To our knowledge, this work is among the first pruning frameworks that jointly optimize layer-wise pruning ratios and pruning strategies using evolutionary multi-objective optimization.

2. RELATED WORK

2.1. Structured pruning for efficient deep networks

Among various model compression techniques, structured pruning has gained particular attention for edge deployment [6], [10] because it physically removes redundant filters or channels, achieving direct reductions in memory footprint and inference latency on general-purpose hardware without requiring sparse computation support.

A central challenge in structured pruning is identifying redundant or unimportant filters. Early approaches relied on heuristic metrics such as weight magnitude, with L1-norm pruning [7] assuming that filters with small weights contribute less to model performance. Subsequent works proposed data-driven criteria to better capture feature redundancy. ThiNet [11] prunes filters based on statistics from the next layer, while network slimming [12] leverages channel-wise scaling factors from batch normalization. FPGM [13] introduces a geometric median-based criterion for filter selection. For example, HRank [14] measures the rank of feature maps to estimate filter importance, while Molchanov *et al.* [15], [16] introduced a theoretically grounded criterion based on the first-order taylor expansion of the loss function, approximated by the product of activation and gradient magnitude.

Despite theoretical advances such as the Lottery Ticket Hypothesis [17] and pruning at initialization [18], and the observation by Liu *et al.* [19] that pruned architectures can be retrained from scratch with competitive performance, most existing pruning methods still apply importance criteria in a greedy or globally uniform manner, ignoring that different layers require distinct preservation strategies.

2.2. Automated pruning via search

To avoid manual tuning of per-layer pruning ratios, recent studies have explored automated pruning frameworks based on search or optimization. AMC [8] formulates structured pruning as a reinforcement learning problem, where an agent learns layer-wise compression ratios. While effective, AMC relies on fine-tuning-based reward evaluation, resulting in substantial computational cost. ABCPruner [20] further models channel pruning as an optimization-based structure search problem, enabling flexible exploration of the pruning space. Similarly, Feng *et al.* [21] propose a task-driven sparsity approach that automatically learns layer-wise pruning rules via a differentiable search mechanism, reducing the reliance on manually designed criteria.

To reduce evaluation cost, proxy-based methods have been proposed. EagleEye [22] estimates sub-network accuracy without full training by leveraging adaptive batch normalization statistics, significantly accelerating the search process. MetaPruning [23] uses meta-learning to generate pruned network weights directly, while once-for-all (OFA) [24] trains a single super-network that can be specialized for diverse deployment scenarios. Efficient network design has also been advanced by architectures such as MobileNetV2 [25] and EfficientNet [26], which incorporate efficient building blocks. Despite these advances, most automated pruning methods primarily focus on optimizing how much to prune per layer, while implicitly assuming a fixed pruning criterion (e.g., L1-norm) throughout the network.

Moreover, prior evolutionary approaches typically formulate pruning as a single-objective problem (e.g., maximizing accuracy under a FLOPs constraint). While recent works have begun exploring multi-objective evolutionary pruning [27], [28], these methods primarily search for pruning ratios under a fixed, globally uniform criterion. In contrast, our work employs multi-objective optimization (MOO) to jointly explore both pruning ratios and per-layer pruning strategies, explicitly trading off accuracy, efficiency, and model size.

2.3. Gap analysis and motivation

Despite recent advances in automated pruning, existing methods still face limitations in both the design of the pruning search space and the efficiency of candidate evaluation during the search process.

- Rigidity of global pruning criteria: while automated frameworks like AMC [8] and ABCPruner [20] can search for distinct layer-wise ratios, they typically rely on a globally uniform pruning criterion across all layers. Such a uniform pruning criterion may not fully account for the heterogeneous representational roles of shallow and deep layers. Consequently, the search process often focuses primarily on optimizing how much to prune, while the choice of how to prune is usually predefined.
- High search latency and evaluation challenges: evolutionary and reinforcement learning-based pruning methods often incur high computational costs due to repeated fine-tuning or validation cycles. Proxy-based strategies, such as EagleEye [22], mitigate this burden by estimating sub-network performance without full retraining. However, these approaches typically require a calibration phase to recompute batch normalization statistics through several forward passes on a calibration set for each candidate sub-network. While relatively lightweight, this additional step can introduce non-negligible cumulative latency when evaluating a large population of candidates in extensive search spaces.

Our motivation. These limitations motivate the development of a framework that jointly optimizes how much to prune and how to prune at each layer. By integrating a hybrid strategy space with a training-free Taylor-based proxy, we aim to provide an efficient, adaptive, and topology-preserving solution for model compression.

3. METHOD

To address CNN resource optimization, this study first investigates global pruning strategies to identify structural limitations, then proposes a GA framework [9] that jointly searches layer-wise pruning ratios and pruning strategies.

3.1. Preliminary study on global strategies

We empirically evaluate global structured pruning to analyze the impact of filter selection criteria. The importance score $\mathcal{I}(f_i)$ for each filter f_i in VGG16 relies on the first-order taylor expansion [15], [16].

We evaluated three selection strategies: min-importance prunes lowest-score filters to preserve sensitive features; max-importance prunes highest-score filters (hypothesizing they cause overfitting); and median-rank preserves filters representing the central tendency $\left[\lfloor\frac{N-k}{2}\rfloor, \lfloor\frac{N+k}{2}\rfloor\right]$.

Our proposed hybrid strategy balances immediate accuracy with global network integrity. While min-importance maintains accuracy in high-variance layers, its rigid application in deep, attenuated layers causes layer collapse. Consequently, median-rank serves as a distributional regularizer, maintaining feature diversity and topological connectivity. This dual space lets the GA adaptively determine optimal layer-wise trade-offs.

Quantitative analysis of layer-wise score imbalance: To understand the failure of global thresholds in deep layers, we analyzed the statistical distribution of taylor importance scores across all 13 convolutional layers of VGG16 (Table 1).

Table 1. Distribution of taylor scores across all VGG16 convolutional layers (calculated via a representative pre-computation pass on a CIFAR-10 data batch)

Layer	Max score	Min score	Mean score	Std. Dev	Attenuation (vs. L00 Peak)
Layer 00 (Conv1_1)	0.5680	0.0001	0.0768	0.0986	1.000 (100.0%)
Layer 01 (Conv1_2)	0.4023	0.0005	0.0956	0.0805	0.708 (70.8%)
Layer 02 (Conv2_1)	0.3023	0.0003	0.0692	0.0550	0.532 (53.2%)
Layer 03 (Conv2_2)	0.3049	0.0006	0.0680	0.0564	0.537 (53.7%)
Layer 04 (Conv3_1)	0.3067	0.0001	0.0474	0.0408	0.540 (54.0%)
Layer 05 (Conv3_2)	0.2568	0.0003	0.0487	0.0392	0.452 (45.2%)
Layer 06 (Conv3_3)	0.2325	0.0001	0.0487	0.0392	0.409 (40.9%)
Layer 07 (Conv4_1)	0.1899	0.0000	0.0334	0.0290	0.334 (33.4%)
Layer 08 (Conv4_2)	0.1840	0.0001	0.0334	0.0289	0.324 (32.4%)
Layer 09 (Conv4_3)	0.1737	0.0000	0.0346	0.0275	0.306 (30.6%)
Layer 10 (Conv5_1)	0.5022	0.0000	0.0291	0.0332	0.884 (88.4%)
Layer 11 (Conv5_2)	0.1570	0.0001	0.0354	0.0265	0.276 (27.6%)
Layer 12 (Conv5_3)	0.0703	0.0256	0.0437	0.0066	0.124 (12.4%)

Empirical data reveals stark variance in feature importance distribution. High-variance layers: early layers (e.g., Layer 00) and specific intermediate layers exhibit high max-to-mean variance, containing a few dominating filters with exceptionally high scores ($\sim$0.50–0.57). Signal attenuation: the deepest layer (layer 12) suffers from extreme score attenuation. Its maximum taylor score (0.0703) is nearly $8\times$ lower than that of Layer 00, representing a relative attenuation of just 12.4%. Its variance is extremely tight (std = 0.0066), meaning the scores are uniformly distributed near zero.

This severe disparity creates a conflict for global criteria. A modest global retention threshold of 0.10 would catastrophically eliminate every filter in layer 12 (max score = 0.0703), despite its critical high-level semantic features. This phenomenon, termed layer collapse, exposes the limits of rigid global heuristics and motivates our adaptive, multi-objective search.

To address this limitation, Figure 1 illustrates the overall workflow of the proposed adaptive GA-based pruning framework. Figure 2 shows the NSGA-II Pareto front, illustrating the trade-off between compression and accuracy used to select the optimal pruning solution.

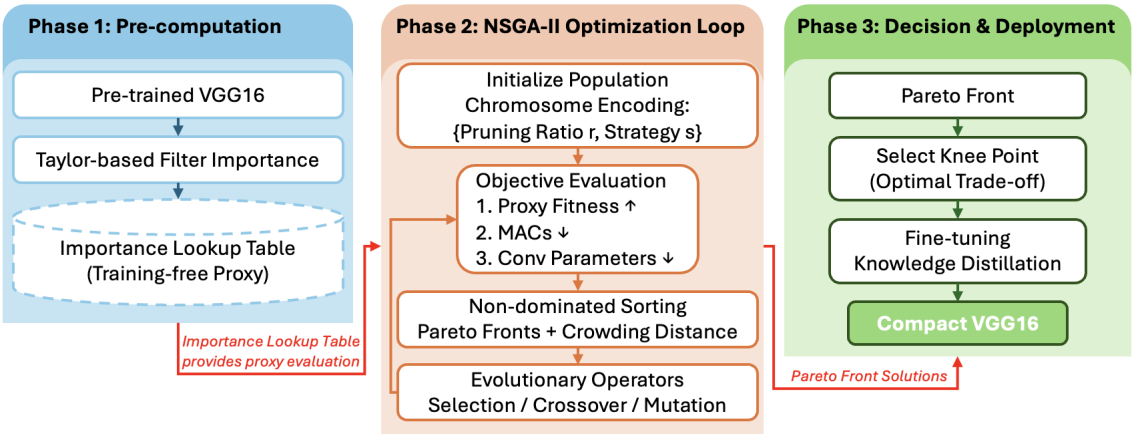

Figure 1. Overview of the proposed multi-objective evolutionary pruning framework. The pipeline consists of three stages: (1) Pre-computation: filter importance scores are estimated once via Taylor expansion and cached as a lookup table for training-free proxy evaluation. (2) NSGA-II Search: candidate pruning policies—each encoding layer-wise pruning ratios and strategies (min-importance or median-rank)—are evolved and evaluated against three objectives: proxy fitness, MACs, and convolutional parameters. (3) Decision and deployment: a knee-point solution is selected from the resulting Pareto front and fine-tuned with knowledge distillation to produce the compact model.

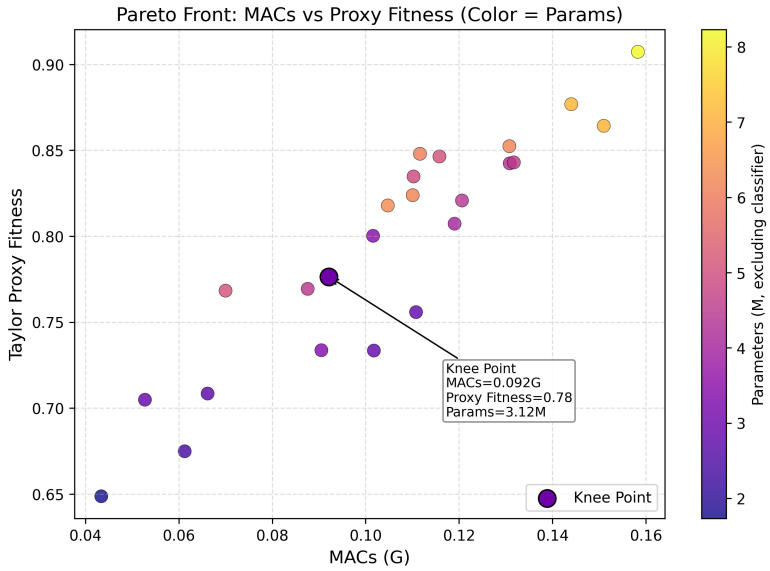

Figure 2. NSGA-II Pareto front on CIFAR-10 (representative run). Each point is a non-dominated pruning configuration. The front reveals a clear compression–accuracy trade-off: solutions toward the lower-left achieve aggressive compression at reduced proxy fitness, while those toward the upper-right preserve higher fitness at greater parameter cost. The knee point identifies the architecture offering the best balance between efficiency and accuracy preservation, selected for subsequent fine-tuning.

3.2. Search space and layer-wise adaptive pruning policy

Unlike most existing automated pruning methods that primarily search for pruning ratios, our framework introduces a dual search space that jointly optimizes pruning ratios (how much to prune) and pruning strategies (how to prune).

Given a CNN with L convolutional layers, each chromosome $\mathbf{z}$ represents,

$$\mathbf{z} = [(r_1, s_1), (r_2, s_2), \ldots, (r_L, s_L)], \tag{1}$$

where $r_l \in (0, 1]$ denotes the retention ratio of layer l, and s_l denotes the selected pruning strategy (min-importance or median-rank).

We exclude max-importance since removing highly important filters typically leads to severe accuracy degradation and contradicts standard pruning objectives.

3.3. Taylor-based proxy fitness

Evaluating candidate pruning configurations via fine-tuning is computationally expensive during evolutionary search. To enable efficient evaluation, we employ a taylor-based proxy fitness derived from the first-order taylor importance criterion [15], [16].

Given a pruning configuration $\mathbf{z}$ retaining filters $S_l(\mathbf{z})$ in each layer, the global taylor preservation ratio is defined as,

$$\mathcal{P}_T(\mathbf{z}) = \frac{\sum_{l=1}^{L} \sum_{i \in S_l(\mathbf{z})} T_{l,i}}{\sum_{l=1}^{L} \sum_{i=1}^{C_l} T_{l,i}} \tag{2}$$

where C_l is the number of filters in layer l. This ratio measures the proportion of taylor importance preserved after pruning and serves as an efficient proxy for ranking candidate configurations during the evolutionary search.

For the multi-objective formulation, we use a layer-wise normalized variant to mitigate dominance from high-importance layers:

$$\mathcal{P}_T^{norm}(\mathbf{z}) = \frac{1}{L} \sum_{l=1}^{L} \frac{\sum_{i \in S_l(\mathbf{z})} T_{l,i}}{\sum_{i=1}^{C_l} T_{l,i}} \tag{3}$$

3.4. Single-objective fitness formulation

To establish a baseline, we designed a single-objective genetic algorithm (GA-SO) aggregating objectives into a scalar fitness function:

$$F(\mathbf{z}) = \mathcal{P}_T(\mathbf{z}) - \lambda_1 \cdot \phi(\mathbf{z}) - \lambda_2 \cdot \psi(\mathbf{z}) - \lambda_3 \cdot r_{param}(\mathbf{z}) \tag{4}$$

where $\mathcal{P}_T(\mathbf{z})$ denotes the taylor-based proxy fitness defined in section 3.3.

The MACs penalty is defined as:

$$\phi(\mathbf{z}) = \max(0, r_{mac} - \tau) + \beta \cdot \max(0, r_{mac} - \tau)^2 \tag{5}$$

The structural vulnerability penalty is defined as:

$$\psi(\mathbf{z}) = \sum_{l \in \mathcal{L}_{vuln}} \max(0, \rho - r_l)^2 \tag{6}$$

In addition to this soft penalty, the full structural safeguard mechanism enforces a hard floor of 10% retention per layer (with filter counts rounded to multiples of 8 for hardware alignment) during chromosome decoding. Any solution violating this hard boundary is rejected ($F = -\infty$).

3.5. Multi-objective formulation and taylor-based proxy

Rather than coalescing constraints into scalar fitnesses, we formulate pruning as a multi-objective optimization problem (MOP):

$$\min_{\mathbf{z}} \mathbf{F}(\mathbf{z}) = [f_{\text{acc}}(\mathbf{z}), f_{\text{mac}}(\mathbf{z}), f_{\text{param}}(\mathbf{z})] \tag{7}$$

where $f_{\text{acc}}(\mathbf{z}) = -\mathcal{P}_T(\mathbf{z})$ and $\mathcal{P}_T(\mathbf{z})$ is the Taylor-based proxy fitness defined in Section 3.3..

To solve (7), we employ the non-dominated sorting genetic algorithm II (NSGA-II) [29]. Individuals are ranked via non-dominated sorting based on Pareto dominance, where a solution $\mathbf{z}_1$ dominates $\mathbf{z}_2$ ($\mathbf{z}_1 \prec \mathbf{z}_2$) if it is no worse in all objectives and strictly better in at least one. Diversity within each front is maintained using crowding distance. During evolution, we apply single-point crossover to exchange genes between parent solutions. Mutation reinitializes ratio genes uniformly within $[\text{min\_ratio}, 1.0]$ and randomly reassigns strategy genes. The overall optimization procedure is summarized in Algorithm 1.

Algorithm 1 NSGA-II based hybrid structured pruning

```
Input: Pre-computed taylor scores, population size N, generations G.
Initialize population P_0 with random pruning ratios and strategies.
Evaluate objectives F(z) for all z ∈ P_0.
for t = 0 to G − 1 do
    Generate offspring Q_t via selection, crossover, and mutation.
    Evaluate objectives for all z ∈ Q_t.
    Merge populations R_t = P_t ∪ Q_t.
    Perform non-dominated sorting on R_t to obtain fronts F_1, F_2, ....
    Construct P_{t+1} using fronts and crowding distance.
end for
Output: Pareto front F_1.
```

Knee point selection. The NSGA-II algorithm produces a Pareto front $\mathcal{F}_1$ containing multiple non-dominated solutions. To obtain a single deployable model, we select the knee point $\mathbf{z}^*$, defined as the solution minimizing the Euclidean distance to the ideal point in the normalized objective space. This strategy provides a balanced trade-off between accuracy preservation and resource reduction.

$$\mathbf{z}^* = \arg\min_{\mathbf{z} \in \mathcal{F}_1} \sqrt{\hat{f}^2_{acc}(\mathbf{z}) + \hat{f}^2_{mac}(\mathbf{z}) + \hat{f}^2_{param}(\mathbf{z})} \tag{8}$$

4. RESULTS AND DISCUSSION

4.1. Experimental setup

We evaluate our framework on the VGG16 architecture [1] using two benchmark datasets: CIFAR-10 and CIFAR-100 [30]. Baseline accuracy, parameters, and MACs for these datasets are detailed in subsequent result tables.

The evolutionary search operates with a population size of 50 over 30 generations on a single NVIDIA RTX 4090 GPU. The entire search process completes in under 15 seconds because taylor importance scores are pre-computed exactly once. The 1,500 genetic evaluations are performed using these cached ranking tables via vectorized summation, eliminating iterative fine-tuning during search. Final architectures are fine-tuned for 150 epochs using SGD with knowledge distillation (KD) [31] ($T = 4.0, \alpha = 0.9$). Full implementation details are publicly available (see data availability).

4.2. Stability analysis: global vs. adaptive pruning

We first investigate the robustness of different pruning strategies under an extreme compression regime (targeting $< 0.1G$ MACs). Table 2 compares our adaptive framework with standard global pruning heuristics. As shown in Table 2, rigid global ranking strategies such as median and max fail under extreme compression, leading to layer collapse. In contrast, our adaptive GA frameworks consistently identify stable architectures that remain operable at similar compression levels.

Table 2. Stability comparison at extreme compression (CIFAR-10)

Strategy	MACs (G)	Params (M)	Acc (%)	Status
Baseline	0.33	33.65	93.62	-
Global Min	0.07	18.15	92.53	Stable
Global Median	-	-	CRASH	Layer collapse
Global Max	-	-	CRASH	Layer collapse
Adaptive GA-SO	0.087	23.21	92.33	Stable
Adaptive GA-MO	0.099 ± 0.010	21.77 ± 0.81	92.78 ± 0.28	Stable

4.3. Comparison with structured pruning methods

Table 3 compares our method with representative structured pruning methods that report results on VGG16/CIFAR-10. As shown in Table 3, L1-Norm and HRank achieve slightly higher absolute accuracy but operate at substantially lower compression levels (34–54% MACs reduction). ABCPruner [20] reaches a comparable MACs reduction (73.68%) while retaining higher accuracy (93.08%), as it performs optimization-based ratio search. However, ABCPruner searches only for per-layer pruning ratios under a fixed criterion.

Table 3. Comparison with representative structured pruning methods on VGG16 (CIFAR-10)

Method	Acc (%)	Params ↓ (%)	MACs ↓ (%)
Baseline	93.62	0	0
L1-Norm [7]	93.40	64.0	34.2
HRank [14]	93.43	82.9	53.5
ABCPruner [20]	93.08	88.68	73.68
Ours (GA-SO)	92.33	31.0	73.6
Ours (GA-MO)	92.78 ± 0.28	35.3 ± 2.4	70.0 ± 3.2

By jointly optimizing pruning ratios and strategies, our GA-MO variant achieves $70.0 \pm 3.2\%$ MACs reduction with $92.78 \pm 0.28\%$ accuracy across 50 runs; the absolute comparisons should be interpreted cautiously due to differing training protocols. AMC [8], EagleEye [22], and MetaPruning [23] search for per-layer ratios but do not report VGG16/CIFAR-10 results and require substantially more computation (e.g., EagleEye $\sim$25 GPU hours vs. our $<$15 seconds), while our method additionally explores *how* to prune at each layer.

Under aggressive compression on CIFAR-100 (Table 4), **Adaptive GA-MO** achieves 71.82% accuracy with only a 0.96% drop, outperforming all global baselines Since L1-Norm, HRank, and ABCPruner do not report results on the VGG16/CIFAR-100 dataset in their original publications, direct comparison with these methods is not included in Table 4.

Table 4. Performance comparison on CIFAR-100 (VGG16)

Method	MACs (G)	Reduct.	Acc (%)	Drop (%)	Params (M)	Drop (%)
Baseline	0.33	0.0%	72.78	-	34.02	-
Global Min	0.12	63.6%	69.91	-2.87	22.30	34.5%
Global Median	0.14	57.6%	71.02	-1.76	21.45	37.0%
Global Max	0.10	69.7%	68.57	-4.21	22.11	35.0%
Ours (GA-MO)	0.116	64.8%	71.82	-0.96	22.52	33.8%
Ours (GA-SO)	0.086	73.9%	70.68	-2.10	23.26	31.6%

4.4. Discussion

Our analysis in section 3 shows that layer-wise score imbalance causes global median and max strategies to suffer from layer collapse. Our framework avoids this through layer-wise decision making, learning local ratios and strategies simultaneously instead of relying on a single global threshold.

Our experiments reveal distinct behavioral differences between the GA variants. The single-objective formulation (GA-SO), driven by a weighted-sum fitness, converges to a 100% min-importance strategy across all layers. This greedy behavior maximizes the proxy fitness but limits exploration. In contrast, the multi-objective formulation (GA-MO) maintains strategy diversity throughout the search, as illustrated in the Pareto front (Figure 2). Rather than forcing a single threshold, this spread enables selection of architectures tailored to different hardware constraints.

What is new and why it matters. By introducing a dual-variable search space that simultaneously optimizes how much and how to prune, our method extends prior automated pruning approaches that search only for pruning ratios under fixed criteria. This broader search space allows the evolutionary process to discover pruning configurations that remain stable and avoid layer collapse even under aggressive compression regimes.

Role of the structural safeguard. An ablation experiment confirms that disabling the safeguard (soft penalty + hard constraints) does not degrade accuracy, acting as a cost-free safety mechanism. Operationally, the 8-filter alignment ensures hardware-friendly channel counts, improving MACs estimation consistency, while the minimum retention floor prevents search regions prone to collapse.

4.5. Proxy fidelity and statistical robustness

To validate proxy reliability, we correlate proxy fitness with post-fine-tuning accuracy across 32 non-dominated sub-networks from a single NSGA-II Pareto front (each pruned and fine-tuned for 150 epochs with identical KD settings). The Spearman correlation is $\rho = 0.8035$ ($p = 3.08 \times 10^{-8}$) and pearson $r = 0.8272$ ($p = 5.32 \times 10^{-9}$), confirming that the proxy reliably ranks candidates across different compression regimes, consistent with EagleEye [22].

Over 50 independent runs with different random seeds, the full GA-MO pipeline achieves mean = 92.78%, std = 0.28%, min = 92.01%, max = 93.33%, indicating stable convergence to high-quality solutions independent of random initialization.

4.6. Limitations

This work introduces limitations warranting future investigation: (i) Architecture scope: a preliminary evaluation on ResNet-56 (CIFAR-10) yields 90.07 $\pm$ 0.88% accuracy at 83.7 $\pm$ 1.7% MACs reduction over 88 runs (baseline: 93.44%), suggesting transferability to residual architectures. (ii) Dataset scale: validation is limited to CIFAR. Larger datasets (e.g., ImageNet) may exhibit distinct pruning dynamics. (iii) Proxy approximation: the taylor proxy relies on a first-order approximation and may miss second-order filter interactions. (iv) Strategy space: expanding beyond the two current strategies (e.g., geometric-median) may yield improvements.

5. CONCLUSION

In this paper, we present an adaptive structured pruning framework using a multi-objective evolutionary algorithm (NSGA-II) with a layer-wise hybrid strategy. By jointly optimizing pruning ratios and pruning strategies at each layer, the proposed framework moves beyond rigid global heuristics that apply a single criterion uniformly. The multi-objective formulation enables explicit exploration of trade-offs between accuracy, computational cost, and model size.

Experimental results on VGG16 show that our method achieves 92.78 $\pm$ 0.28% accuracy with 70.0 $\pm$ 3.2% MACs reduction on CIFAR-10 while maintaining stable architectures under aggressive compression. On CIFAR-100, the framework retains 71.82% accuracy with a 64.8% MACs reduction, indicating good generalization to more challenging classification tasks.

Proxy fidelity analysis confirms that the Taylor-based proxy reliably ranks candidate architectures (strong rank correlation), and statistical validation over 50 independent runs demonstrates consistent reproducibility (std $= 0.28\%$). Ultimately, these empirical advantages suggest that the proposed framework holds significant potential to facilitate robust compression for mobile AI and edge devices, reducing theoretical inference costs while preserving accuracy.

While the framework demonstrates strong empirical advantages on VGG16, a preliminary evaluation on ResNet-56 suggests transferability to residual architectures. Full validation on larger architectures and datasets remains as future work.

Future work will focus on: (i) full statistical validation on residual architectures (ResNet-50/ImageNet) and Transformer-based models; (ii) expanding the strategy search space with geometric or activation-based criteria; and (iii) deploying compressed models on real-world edge devices to measure physical inference latency and energy efficiency.

ACKNOWLEDGMENTS

The authors would like to thank the MMLab, University of Information Technology, Vietnam National University Ho Chi Minh City, for providing the computational resources used in this research.

FUNDING INFORMATION

This research was funded by University of Information Technology, Vietnam National University Ho Chi Minh City under grant number D1-2026-06.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Anh-Truong Vo	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Hoang-Loc Tran			✓	✓	✓	✓					✓			✓
Dinh-Duy Phan			✓	✓	✓	✓	✓				✓			
Duc-Lung Vu	✓	✓			✓	✓	✓			✓		✓	✓	✓

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

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

DATA AVAILABILITY

The experiments in this study were conducted using publicly available datasets (CIFAR-10 and CIFAR-100). The source code implementing the proposed framework is publicly available at: `https://github.com/truongva1111/adaptive-ga-pruning`.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Apr. 2015, [Online]. Available: http://arxiv.org/abs/1409.1556

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[3] F. N. Peccia, S. Pavlitska, T. Fleck, and O. Bringmann, "Efficient Edge AI: deploying convolutional neural networks on FPGA with the gemmini accelerator," in *2024 27th Euromicro Conference on Digital System Design (DSD)*, IEEE, Aug. 2024, pp. 418–426. doi: 10.1109/DSD64264.2024.00062.

[4] V. Kamath and A. Renuka, "Deep learning based object detection for resource constrained devices: systematic review, future trends and challenges ahead," *Neurocomputing*, vol. 531, pp. 34–60, Apr. 2023, doi: 10.1016/j.neucom.2023.02.006.

[5] M. Y. Shabir, G. Torta, and F. Damiani, "TinyML model compression: a comparative study of pruning and quantization on selected standard and custom neural networks," *Telecommunication Systems*, vol. 88, no. 4, p. 132, Dec. 2025, doi: 10.1007/s11235-025-01363-2.

[6] Y. He and L. Xiao, "Structured pruning for deep convolutional neural networks: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 2900–2919, 2024, doi: 10.1109/TPAMI.2023.3334614.

[7] H. Li, K. Asim, D. Igor, S. Hanan, and G. H. Peter, "Pruning filters for efficient convnets," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, vol. 1608.08710, 2016.

[8] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "AMC: AutoML for model compression and acceleration on mobile devices," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, 2018, pp. 815–832. doi: 10.1007/978-3-030-01234-2_48.

[9] J. H. John and J. Henry, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," University of Michigan Press, 1992.

[10] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning: taxonomy, comparison, analysis, and recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10558–10578, Dec. 2024, doi: 10.1109/TPAMI.2024.3447085.

[11] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: a filter level pruning method for deep neural network compression," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2017, pp. 5068–5076. doi: 10.1109/ICCV.2017.541.

[12] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2017, pp. 2755–2763. doi: 10.1109/ICCV.2017.298.

[13] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2019, pp. 4335–4344. doi: 10.1109/CVPR.2019.00447.

[14] M. Lin *et al.*, "HRank: filter pruning using high-rank feature map," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020, pp. 1526–1535. doi: 10.1109/CVPR42600.2020.00160.

[15] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[16] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2019, pp. 11256–11264. doi: 10.1109/CVPR.2019.01152.

[17] J. Frankle and M. Carbin, "The lottery ticket hypothesis: finding sparse, trainable neural networks," *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[18] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," in *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[19] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[20] M. Lin, R. Ji, Y. Zhang, B. Zhang, Y. Wu, and Y. Tian, "Channel pruning via automatic structure search," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, California: International Joint Conferences on Artificial Intelligence Organization, Jul. 2020, pp. 673–679. doi: 10.24963/ijcai.2020/94.

[21] K.-Y. Feng, X. Fei, M. Gong, A. K. Qin, H. Li, and Y. Wu, "An automatically layer-wise searching strategy for channel pruning based on task-driven sparsity optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 5790–5802, Sep. 2022, doi: 10.1109/TCSVT.2022.3156588.

[22] B. Li, B. Wu, J. Su, and G. Wang, "EagleEye: fast sub-net evaluation for efficient neural network pruning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12347 LNCS, 2020, pp. 639–654. doi: 10.1007/978-3-030-58536-5_38.

[23] Z. Liu *et al.*, "MetaPruning: meta learning for automatic neural network channel pruning," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3295–3304, 2019, doi: 10.1109/ICCV.2019.00339.

[24] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: train one network and specialize it for efficient deployment," in *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[26] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in *36th International Conference on Machine Learning, ICML 2019*, 2019, pp. 6105–6114.

[27] W. Hong, G. Li, S. Liu, P. Yang, and K. Tang, "Multi-objective evolutionary optimization for hardware-aware neural network pruning," *Fundamental Research*, vol. 4, no. 4, pp. 941–950, Jul. 2024, doi: 10.1016/j.fmre.2022.07.013.

[28] J. Poyatos, D. Molina, A. Martínez-Seras, J. Del Ser, and F. Herrera, "Multiobjective evolutionary pruning of deep neural networks with transfer learning for improving their performance and robustness," *Applied Soft Computing*, vol. 147, p. 110757, Nov. 2023, doi: 10.1016/j.asoc.2023.110757.
[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002, doi: 10.1109/4235.996017.
[30] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Department Computer Science , University of Toronto*, Nov. 2009.
[31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015, [Online]. Available: http://arxiv.org/abs/1503.02531

BIOGRAPHIES OF AUTHORS

Anh-Truong Vo received the Engineer's degree in Information Technology in 2022. He is currently pursuing the M.S. degree at the University of Information Technology, VNU-HCM, Vietnam. His research interests include model compression, neural architecture search, and evolutionary algorithms. He can be contacted at truongva.18@grad.uit.edu.vn.

Hoang-Loc Tran received a B.S. degree in Computer Engineering from the University of Information Technology, Vietnam National University Ho Chi Minh City (UIT VNU-HCMC) in 2018. Currently, he is pursuing an M.Sc. degree in Computer Science from UIT VNU-HCMC. From 2018 he started working as a researcher and teaching assistant at the Faculty of Computer Engineering, UIT VNU-HCMC. His research interests include machine learning, computer vision, edge computing, and embedded system. He can be contacted at email: locth@uit.edu.vn.

Dinh-Duy Phan received his B.Eng. degree in Computer Engineering in 2011 and his M.Sc. degree in Computer Science in 2014 from the University of Information Technology, Vietnam National University – Ho Chi Minh City, Vietnam. In 2011, he joined the Faculty of Computer Engineering at the University of Information Technology, as a lecturer. He is currently a doctoral student at the same university. His research interests include embedded system design, IC design, machine learning, and computer vision, as well as their applications on personal computers and embedded systems. He can be contacted at duypd@uit.edu.vn.

Duc-Lung Vu received the B.S. and M.Sc. degrees in Computer Engineering from Peter the Great St.Petersburg Polytechnic University in 1998 and 2000, respectively. He got his Ph.D. in Computer Science from Saint Petersburg Electrotechnical University in 2006. He has been working at the University of Information Technology, Vietnam National University Ho Chi Minh City, as an Associate Professor since 2015. His research interests include machine learning, human-computer interaction, embedded systems and digital system design on FPGA. He can be contacted at lungvd@uit.edu.vn.