International Journal of Advanced Engineering, Management and Science (IJAEMS)
Peer-Reviewed Journal
ISSN: 2454-1311 | Vol-11, Issue-6; Nov-Dec, 2025
Journal Home Page: https://ijaems.com/
DOI: https://dx.doi.org/10.22161/ijaems.116.2

Research on Recovery under Pruning Degeneration Using LoRA Technology

Hao-Lin Ye, Chih-Ying Chuang*

Guangdong-Taiwan College of Industrial Science & Technology, Dongguan University of Technology (DGUT), Dongguan 523000, China
*Corresponding Author, Email: ccyrandy@dgut.edu.cn

Received: 28 Sept 2025; Received in revised form: 31 Oct 2025; Accepted: 04 Nov 2025; Available online: 08 Nov 2025

***Abstract*—** *In several fields, Convolutional Neural Networks (CNNs) have demonstrated impressive progress in recent years. However, its adoption on devices with limited resources is limited by its enormous model scale and high computational requirements. Neural network pruning in particular has become one of the most important methods for resolving this problem. The choice of importance criteria has a significant impact on pruning's effectiveness. Without systematic comparisons of numerous criteria under the same pruning ratio, the majority of the research to far has been on the proposal of single criteria or comparisons under non-strict control. Furthermore, trimming frequently results in performance loss that must be fixed through fine-tuning. The advent of parameter-efficient fine-tuning algorithms like LoRA offers a fresh approach to addressing the high computational cost of conventional global fine-tuning. It is still unknown how they work together with various pruning criteria. This is accomplished by conducting controlled experiments on the CIFAR-10 dataset to evaluate the performance of three widely used pruning criteria: L_1-Norm pruning, SNIP pruning, and Taylor pruning, at pruning ratios ranging from 30% to 60%. LoRA is being methodically incorporated into the pruning recovery stage for the first time, demonstrating that it is a versatile and successful fine-tuning method that might significantly lessen the performance loss caused by trimming. Furthermore, in order to support the deployment of effective neural networks, this research offers empirical evidence for choosing suitable pruning and fine-tuning procedures for actual application objectives as seeking compression rate or accuracy.*

***Keywords*— L_1-Norm Pruning, SNIP Pruning, Taylor Pruning, LoRA, Model Sparsity.**

I. INTRODUCTION

Since the advent of artificial intelligence, Convolutional Neural Networks (CNNs) have achieved remarkable accomplishments across numerous domains. However, their substantial model size and computational demands significantly hinder deployment on resource-constrained devices. To address this challenge, model compression techniques, particularly neural network pruning, have emerged as key tools for reducing model scale and enhancing inference efficiency [1-3].

As a mainstream model compression technique, the core idea of pruning is to remove redundant parameters in the network while preserving model performance as much as possible. Based on whether the original network structure is retained after pruning, it can be categorized into unstructured pruning and structured pruning. Early pioneering work, such as the Optimal Brain Surgeon proposed by Hassibi and Stork, laid the foundation for unstructured pruning based on the Hessian matrix [4]. However, unstructured pruning requires specialized hardware to achieve acceleration. Consequently, structured pruning, which directly reduces the number of channels or filters and thereby enables acceleration on general-purpose hardware, has become a key research focus in recent years [5, 6]. Structured pruning methods are widely applied in

both CNNs and LLMs [7].

In unstructured pruning, how to define parameter importance is crucial to its effectiveness [8-11]. Different importance criteria have led to the formation of various pruning approaches. Among them, the most intuitive are norm-based criteria, such as using the L_1-Norm of filter weights to measure their importance [5]. This method is computationally efficient and requires no additional data, making it widely used as a baseline. However, its main limitation is that it is a static estimation that fails to account for the actual contribution of parameters during training dynamics. To assess parameter importance more accurately, gradient-based criteria were proposed. The first-order Taylor expansion criterion introduced by Molchanov was a milestone work [12]. This criterion uses the product of the gradient of the loss function with respect to the weight and the weight itself to approximate its importance, positing that parameters with less impact on the loss function are less important. By incorporating training dynamics, this method theoretically better identifies redundant parameters [13]. Han et al. have also demonstrated that unstructured pruning can effectively compress model networks and reduce model size [14]. To further reduce computational overhead, connection sensitivity-based criteria enable pruning early in the training phase, with Lee's proposed Single-shot Network Pruning (SNIP) criterion being a prominent example [15]. SNIP measures the sensitivity of each weight by calculating the magnitude of the gradient of the loss function with respect to it and prunes low-sensitivity connections in a single step before training begins. This method offers high efficiency, but its "preemptive" pruning strategy has also sparked discussions regarding its robustness.

The core of unstructured pruning lies in identifying and removing redundant parameters in the network, and its effectiveness highly depends on the adopted importance criterion. Although criteria such as L_1-Norm, Taylor, and SNIP have been widely proposed and applied, existing research often focuses on the promotion of a single criterion or makes comparisons under different experimental settings. There is a lack of systematic comparison and in-depth analysis of the model sparsity patterns, compression efficiency, and final performance resulting from these criteria under strictly controlled conditions with identical pruning ratios. Furthermore, pruning inevitably leads to model performance degradation, making efficient performance recovery crucial. Due to the high computational cost of traditional global fine-tuning, recently emerged Parameter Efficient Fine-Tuning (PEFT) techniques, particularly Low-Rank Adaptation (LoRA) proposed by Hu et al., offer a new direction to address this issue [16]. Although significant progress has been made in the aforementioned research, most works focus on proposing or improving a single criterion. There is insufficient research conducting systematic and fair comparisons of different importance criteria under strictly controlled conditions with the same target pruning ratio [17]. Additionally, the synergistic effects of employing emerging PEFT techniques like LoRA as a standard recovery method after pruning, in conjunction with different pruning criteria, remain underexplored [18-20].

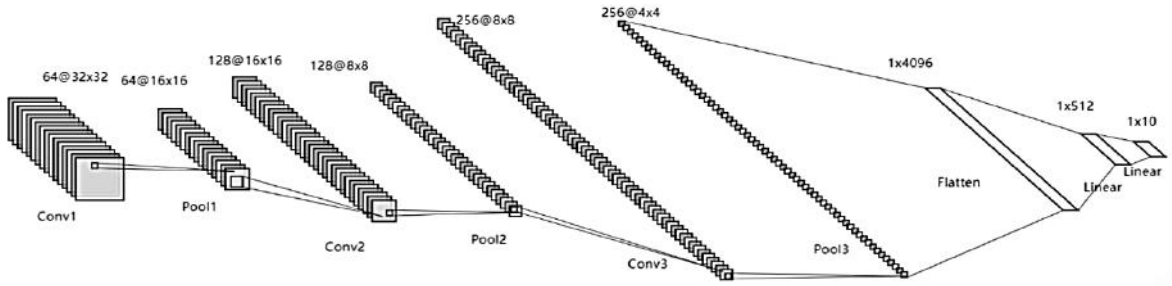

Fig.1. CNN Model Architecture (LeNet Style)

To address these research gaps, this paper designs a controlled experiment on the CIFAR-10 dataset to systematically evaluate the pruning effects of three mainstream importance criteria—L_1-Norm, Taylor, and SNIP—on CNN models at target pruning ratios ranging from 30% to 60%. This paper comprehensively assesses the post-pruning model accuracy retention and places particular emphasis on analyzing the effectiveness of LoRA fine-tuning as an efficient recovery method.

II. MODEL ARCHITECTURE

This paper employs an improved Convolutional Neural Network (Improved CNN) as the benchmark model, with its architecture inspired by the visual geometry group (VGG) network. The model consists of three convolutional blocks, each sequentially composed of a convolutional layer (Conv2d), a batch normalization layer (BatchNorm2d), a ReLU activation function, and a max-pooling layer (MaxPool2d). The output channel numbers for the three convolutional layers are 64, 128, and 256, respectively, with all convolutional kernels having a size of 3x3. Following the final pooling layer, the feature maps are flattened and fed into two fully connected layers (Linear), with a Dropout layer (dropout rate of 0.5) incorporated in between to prevent overfitting. The final output of the model is 10-dimensional, corresponding to the 10 classes of CIFAR-10. The model was first trained on the training set for 60 epochs using the Adam optimizer (learning rate 0.001, weight decay set to 1e-4) and a step learning rate scheduler (StepLR), ultimately achieving a test accuracy of 86.78%. This model serves as the baseline for all subsequent pruning experiments. The architecture of the convolutional neural network model is illustrated in Figure 1, where part of the depth is omitted due to the large number of layers.

The experiments utilized the CIFAR-10 dataset, which consists of 60,000 32x32 color images across 10 categories, with 50,000 images for training and 10,000 for testing. This study adopted a standard data preprocessing and augmentation pipeline to enhance model generalization. For the training set, preprocessing included random horizontal flipping and random cropping (with padding=4), followed by converting pixel values to tensors and normalization (with mean and standard deviation both set to 0.5). The data was loaded via PyTorch's DataLoader, with the batch size set to 128 for both training and testing.

III. PRUNING METHODS WITH PEFT

3.1 Pruning Methods

This paper implements three mainstream structured pruning methods on the pre-trained model and compares them across various target pruning ratios ranging from 30% to 60%. All pruning is performed on the weight parameters.

(1) L_1-Norm Pruning

L_1-Norm Pruning uses the L_1-Norm of the weights as the importance criterion. Its core idea is that a weight with a smaller absolute value is less important. The importance score $I_{L_1}(W_i)$ is defined as:

$$I_{L_1}(W_i) = |W_i| \quad (1)$$

Here, W_i represents the i-th weight in a given layer, meaning the importance score of each weight W_i is its absolute value. The approach adopted in this paper is as follows: for each layer, the L_1-Norm of its weight matrix is calculated, and the specified proportion of weights with the smallest norm values are removed. The specific implementation utilizes PyTorch's built-in torch.nn.utils.prune.l1_unstructured function, followed by prune.remove to permanently eliminate the pruned weights.

(2) Taylor Importance Pruning

The Taylor importance pruning method determines the significance of a weight by evaluating its impact on the loss function. If removing a weight leads to a substantial increase in the loss, the weight is considered highly important; conversely, if its removal results in negligible change or even a decrease in the loss, it is deemed unimportant. The specific importance score $I_{\text{Taylor}}(W_i)$ can be approximated by the absolute value of the product of the weight and its gradient:

$$I_{\text{Taylor}}(W_i) = \left|W_i \cdot \nabla_{W_i} L\right| \quad (2)$$

Here, $\nabla_{W_i} L$ represents the gradient of the loss function with respect to the weight. To achieve a stable estimation, this paper performs forward and backward propagation on the first 10 batches of the training set and accumulates the importance scores of each weight. Therefore, Equation (2) can be rewritten as:

$$I_{\text{Taylor}}^{(\text{cumulative})}(W_i) = \sum_{k=1}^{M} \left|W_i \cdot \nabla_{W_i} L^{(k)}\right| \quad (3)$$

where $\nabla_{W_i} L^{(k)}$ represents the gradient of the loss function with respect to the weight during the k-th pruning round, and M denotes the set of all weights in a given layer. Subsequently, based on the global importance scores, the specified proportion of weights with the lowest scores is removed.

(3) SNIP Pruning

SNIP pruning is a single-shot pruning algorithm based on connection sensitivity, which can be performed before training begins. First, forward and backward propagation are executed on a batch of data, and the absolute value of the gradient for each weight is calculated as its sensitivity score:

$$I_{\text{SNIP}}(W_i) = |g_i \odot W_i| \approx |g_i w_i| \quad (4)$$

Among them, $g_i \triangleq \frac{\partial L}{\partial W_i}$ represents the gradient of the loss function with respect to the weight, and $\odot$ denotes the element-wise multiplication between two vectors. Subsequently, a global sensitivity threshold is calculated, and all weights with sensitivity below this threshold are removed. For convolutional layers, their sensitivity scores can be averaged by the output channels. It is worth noting that while the mathematical formulation of SNIP pruning appears very similar to that of Taylor pruning—and indeed, Taylor's method serves as the theoretical foundation for SNIP—the key distinction adopted in this work is that SNIP performs pruning in a single step before training begins, whereas Taylor pruning calculates gradients and importance scores using the current model at each pruning iteration.

3.2 Parameter-Efficient Fine-Tuning

To address the prevalent issue of performance degradation following model pruning, this study departs from the traditional global fine-tuning approach—which is parameter-inefficient and prone to overfitting—and instead adopts the LoRA technique from the Parameter-Efficient Fine-Tuning (PEFT) paradigm for performance recovery [7]. The core concept of LoRA originates from the observation that the weight update matrices of large models, when adapted to downstream tasks, exhibit an intrinsic low-rank property.

Building on this insight, an improved CNN integrated with LoRA modules is designed. This design freezes all original parameters of the pruned network to preserve acquired knowledge, while only injecting trainable LoRA layers in parallel alongside the fully connected layers. These adaptation layers approximate the incremental update ΔW of the original weights through the product of two small matrices B*A. This strategy allows the model to optimize only a minimal number of newly added parameters during fine-tuning, thereby efficiently guiding the model to adapt to new tasks while significantly reducing computational and storage costs.The hyperparameters for LoRA are set as follows: rank = 8, scaling factor (alpha) = 16.0. Only these newly added parameters were trained for 10 epochs using the Adam optimizer with a learning rate of 0.001.

IV. EXPERIMENTAL RESULTS

This paper presents the effects of different pruning methods, including L_1-Norm, Taylor, and SNIP, on model performance, compression efficiency, and fine-tuning recovery under various pruning ratios. All experiments are based on the CIFAR-10 dataset and the Improved CNN model described in Section 2, and the following evaluation metrics are adopted to comprehensively assess the effectiveness of different pruning methods:

(1) Top-1 Accuracy (%): The classification accuracy of the model on the test set, serving as the core metric for evaluating performance retention.

(2) Pruning Ratio (%): The targeted proportion of weights to be removed, used as a controlled variable in the experiments.

(3) Accuracy Drop (%): The difference in accuracy between the pruned model and the original model, measuring the destructiveness of pruning.

(4) LoRA Improvement (%): The difference in accuracy after LoRA fine-tuning compared to the accuracy right after pruning, evaluating the effectiveness of recovery.

All experiments were implemented using the PyTorch framework and executed on a single NVIDIA GPU to ensure environmental consistency. The original unpruned Improved CNN model used in this study achieved an accuracy of 86.78% on the test set, serving as the baseline for all comparisons. The model size is 9.45 MB, and this performance will be used as the benchmark for evaluating the accuracy degradation caused by all pruning methods and the subsequent recovery effects.

4.1 The Impact of Pruning Ratio on Accuracy

Following the methodology proposed in this paper, the performance of the models after pruning and before fine-tuning was evaluated to directly assess the destructiveness of each pruning method. The experimental results are presented in Table 1. An initial observation reveals that the three pruning methods differ in their ability to preserve performance, which can be analyzed across different pruning ratios:

(1) Low Target Pruning Ratios (30%-40%):

At low target pruning ratios, the SNIP method demonstrated the best performance, maintaining the highest accuracy rates of 86.61% and 86.07%, which are nearly equivalent to the original model's 86.78%. The Taylor method ranked second, achieving a respectable 81.72% at a 30% pruning ratio but dropping significantly to 68.34% at 40%. In contrast, the L_1-Norm method performed the poorest, with accuracy already declining markedly to 65.17% and 60.64% even at these low pruning ratios.

(2) High Target Pruning Ratios (50%-60%):

When the target pruning ratio was set at 60%, the accuracy of the models pruned by all three pruning methods was less than 40%. The models obtained using Taylor and SNIP pruning methods even had an accuracy of less than 20%. Therefore, this experiment considers that a target pruning ratio of 60% is too high, significantly damaging the underlying structure of the original model. Under a target pruning ratio of 50%, the models pruned by Taylor and L_1-Norm methods had accuracies of less than 50%, suggesting that a 50% target pruning ratio is still too high for these two methods. However, the model obtained using the SNIP pruning method was able to maintain an accuracy of 70.42%, significantly better than the other two methods. Combined with subsequent fine-tuning operations, the model's accuracy can be further improved. Therefore, this experiment shows that a target pruning ratio of 50% is still a meaningful pruning ratio for the SNIP method. Both the Taylor and L_1-Norm methods experienced sharp performance degradation at high pruning ratios. Particularly at the 60% pruning ratio, the accuracy of both Taylor and SNIP dropped to approximately 16%.

Table 1. Accuracy of Three Pruned Models under Different Target Pruning Ratios

Pruning Ratio	L1-Norm	Taylor	SNIP
30%	0.6517	0.8172	0.8661
40%	0.6084	0.6834	0.8607
50%	0.4995	0.4540	0.7042
60%	0.3922	0.1654	0.1617

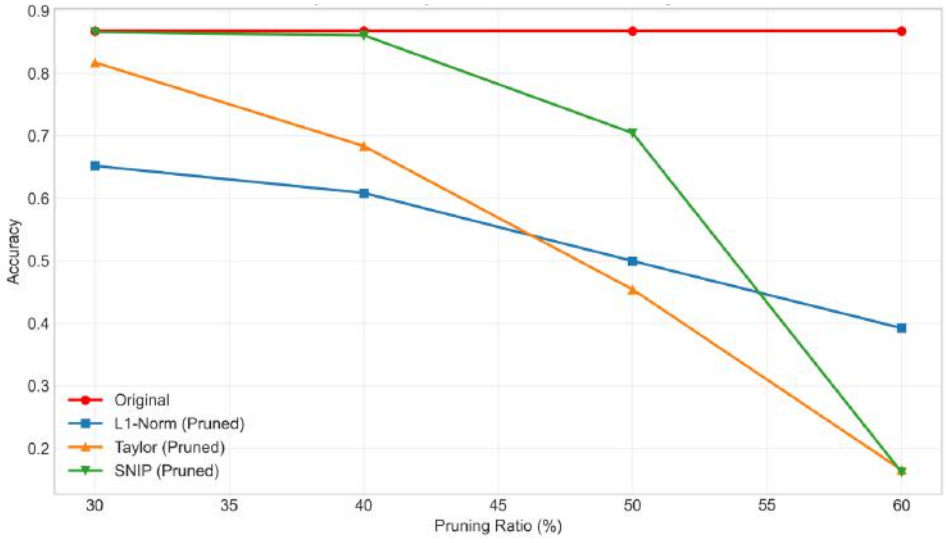

Fig.2. Accuracy vs. Pruning Ratio for Different Pruning Methods

Plotting the data from Table 1 into the line chart of Figure 2 clearly reveals the extent and pattern of impact that different pruning methods have on model performance. The chart shows that as the target pruning ratio increases from 30% to 60%, the accuracy of all pruning methods exhibits a declining trend. It is noteworthy that the three pruning methods—L_1-Norm, Taylor, and SNIP—

demonstrate distinct degradation characteristics across different pruning intervals: at low target pruning ratios, the performance decline of each method is relatively gradual, with minor differences between them; however, when reaching high target pruning ratios, the rate of accuracy drop accelerates significantly, and the performance gap between different methods widens progressively, reflecting the varying capabilities of each pruning criterion in identifying and preserving critical network connections. Particularly notable is that at the high target pruning ratio of 60%, the accuracy of all methods drops to a low level, a phenomenon that highlights the damage caused by excessive pruning to the model's representational capacity, while also emphasizing the necessity and urgency of employing PEFT methods like LoRA for performance recovery under such extreme sparsity conditions.

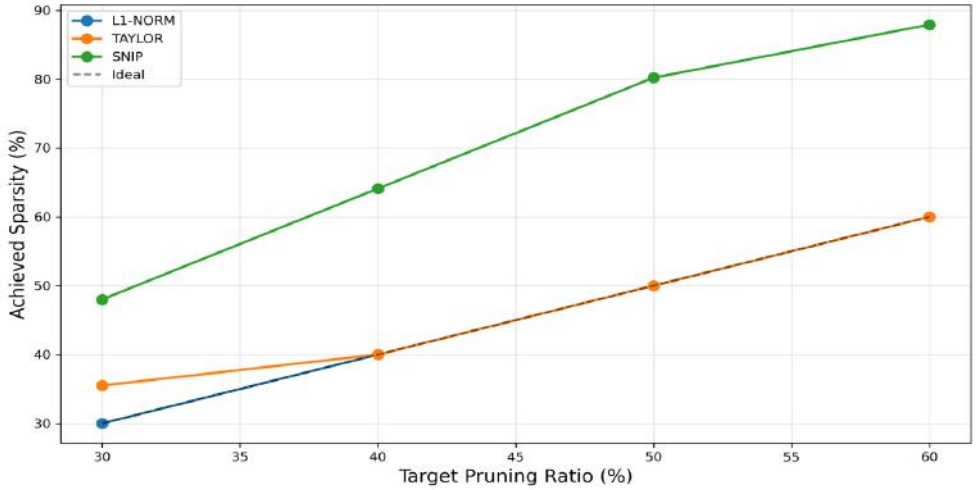

Fig.3. Achieved Sparsity vs. Target Pruning Ratio

The actual sparsity ratios of the pruned models obtained through the three pruning methods are shown in Figure 3. By comparing the actual sparsity levels achieved by different pruning methods under different target pruning rates, this figure provides profound insights into the core performance and control precision of each pruning algorithm. The ideal diagonal line in the figure represents the target sparsity, while the proximity of the three curves (L_1-Norm, Taylor, and SNIP) to this ideal line visually reflects the precision and reliability of each method. Through systematic analysis of the deviation patterns of these curves across different pruning intervals, it can be evaluated whether each method exhibits systematic biases—such as a tendency toward conservative under-pruning or aggressive over-pruning. This quantitative assessment of pruning precision is crucial. It can be observed that the L_1-Norm method achieves actual sparsity that perfectly matches the target sparsity, indicating that this method "follows instructions precisely." The Taylor method shows a slightly higher actual sparsity than the target at the 30% pruning ratio, while matching the target at other ratios, suggesting it possesses a certain level of intelligent judgment but overall still completes the task as required. In contrast, the SNIP method is the most aggressive: when it identifies a large number of redundant weights, it prunes quite boldly. Consequently, at every target pruning ratio, the actual sparsity of SNIP-pruned models significantly exceeds the target sparsity.

4.2 Performance Recovery via LoRA Fine-Tuning

According to the methodology proposed in this paper, the performance of the models after pruning and after fine-tuning was evaluated. These results are presentedalongside the performance of the models after pruning but before fine-tuning in Table 2. The table clearly demonstrates the effectiveness of LoRA fine-tuning, which produced significant performance recovery for all pruning methods and across all pruning ratios, with the sole exception of SNIP at the 60% target pruning ratio. This improvement is visually evident in Figure 4.

Table 2. Accuracy of Three Pruning Methods Combined with LoRA Fine-Tuning under Different Target Pruning Ratios

Pruning Ratio	L1-Norm	L1-Norm + LoRA	Taylor	Taylor + LoRA	SNIP	SNIP + LoRA
30%	0.6517	0.8346	0.8172	0.8579	0.8661	0.8670
40%	0.6084	0.8105	0.6834	0.8418	0.8607	0.8629
50%	0.4995	0.7938	0.4540	0.8230	0.7042	0.8496
60%	0.3922	0.7622	0.1654	0.7583	0.1617	0.1917

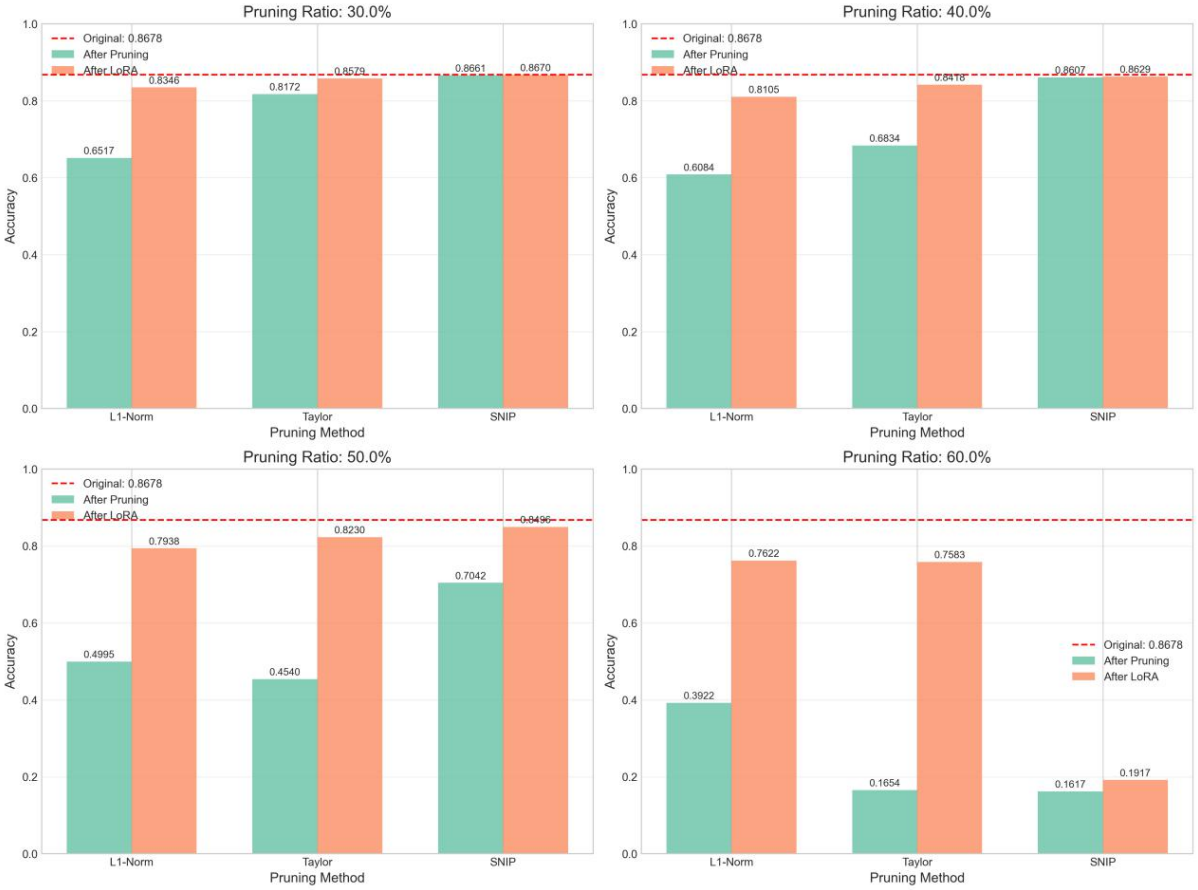

Fig.4. Pruning Methods Comparison at Different Pruning Rations

The experimental data reveal the synergistic effects between different pruning methods and LoRA. For highly destructive pruning methods like L_1-Norm and Taylor, LoRA acts as a "lifesaver." In the case of L_1-Norm + LoRA, even at the 60% target pruning ratio where the post-pruning accuracy plummets to 39.22%, LoRA is able to restore it to 76.22%, representing an improvement of 37 percentage points. For Taylor + LoRA, the most substantial recovery is observed at the 60% pruning ratio with an impressive 59.29% increase. This indicates that although Taylor pruning removes a significant number of weights, it effectively preserves the network's "skeleton" or "potential," enabling LoRA to efficiently reconstruct functionality on this foundation and demonstrating the strongest recovery capability.On the other hand, for pruning methods like SNIP that already maintain good performance, LoRA serves as "icing on the cake." At target pruning ratios of 30%–50%, SNIP alone maintains accuracy between 86.61% and 70.42%, leaving limited room for LoRA to bring improvements ranging from only 0.09% to 14.54%. Nevertheless, the final accuracy achieved by SNIP + LoRA is the highest among all combinations. However, at the 60% target pruning ratio, SNIP's accuracy drops drastically and can no longer be recovered through LoRA fine-tuning. This trend after fine-tuning is visually captured in Figure 5.

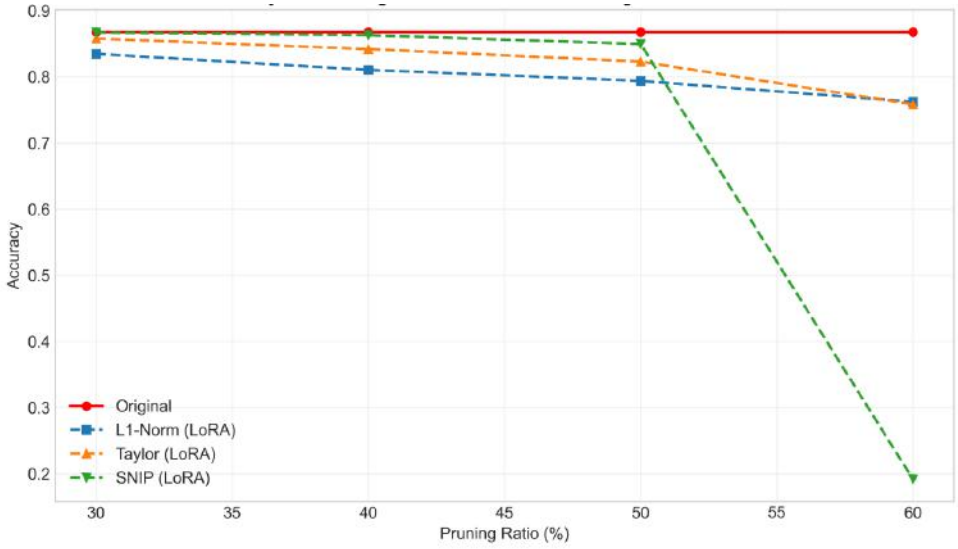

Fig.5. Accuracy vs. Pruning Ratio for Different Pruning Methods with LoRA

4.3 Discussion

The pruning effects of different importance criteria are directly determined by their underlying theoretical assumptions and computational approaches. The SNIP criterion demonstrates excellent performance retention at target pruning ratios of 30%-50%, which can be attributed to its foundation in connection sensitivity. By calculating gradient magnitudes through a single forward-backward pass during the model initialization phase, SNIP identifies weights with the least impact on the loss function. This "one-shot global pruning" strategy tends to remove a substantial number of redundant connections, resulting in actual sparsity that often significantly exceeds the target value, reflecting its strong compression aggressiveness. However, this aggressive approach may excessively remove structurally critical weights in the network at the high target pruning ratio of 60%, damaging the model's skeleton and consequently making subsequent fine-tuning ineffective for performance recovery. This finding aligns with Lee's emphasis on SNIP's efficiency and one-shot pruning advantage, while our experiments also reveal its potential risks under high sparsity demands, thereby supplementing the original research's insufficient discussion of extreme compression scenarios.

The Taylor importance criterion, when combined with LoRA fine-tuning, shows the strongest recovery capability at the 50% target pruning ratio. This is because the Taylor criterion dynamically evaluates parameter importance through the interaction between weights and their gradients, reflecting the actual contribution of parameters during the training process. This gradient-based evaluation enables better discrimination between important and unimportant weights, thereby preserving the functional skeleton of the model after pruning, while enhancing the accuracy of importance estimation through the incorporation of gradient information. Our experimental results further indicate that the Taylor method can achieve effective reconstruction through LoRA even at higher target pruning ratios, suggesting that the retained weight structure contains substantial representational potential.

Although L_1-Norm pruning is simple in criterion and computationally efficient, its sole reliance on weight magnitude while ignoring training dynamics leads to the poorest performance across all pruning ratios. Nevertheless, L_1-Norm pruning combined with LoRA still achieves considerable recovery at the high target pruning ratio of 60%, indicating that while its pruning approach is "blind," it does not destroy the most fundamental network structure. That is, norm-based methods, though imprecise, can still serve as stable compression baselines. Based on the presentation of experimental data, the summarized characteristics of the three pruning methods discussed above are systematically compared and described in Table 3.

Table 3. Comparative Summary of Characteristics of the Three Pruning Methods

Pruning Method	L_1-Norm	Taylor	SNIP
Characteristics	Precise Executor	Mild Over-achiever	Aggressive Over-executor
Performance	The L_1-Norm method precisely achieved the target sparsity across all pruning ratios	It slightly exceeded the target sparsity by 5.5% at the 30% pruning ratio, but accurately met the target at other ratios	It significantly exceeded the target sparsity, with the deviation magnitude increasing as the target ratio rose
Rationale	L_1-Norm pruning, based on simple weight magnitude ranking, can remove weights strictly according to the specified ratio	The Taylor importance criterion, being gradient-based, may identify a larger proportion of "less important" weights in certain layers	SNIP employs a global sensitivity threshold, exhibiting a tendency to remove more connections deemed unimportant
Advantages	It provides predictable and controllable compression outcomes	It demonstrates a certain degree of intelligent adaptive capability	It achieves the highest accuracy rates at target pruning ratios of 30%-50%

V. CONCLUSION

Through systematic comparative experiments, this study reveals the distinct performance of three mainstream pruning criteria—L_1-Norm, Taylor, and SNIP—under identical target pruning ratios, and validates the effectiveness of LoRA fine-tuning in restoring model performance after pruning.

In terms of strategy selection, the optimal choice depends on the target pruning ratio. At low target pruning ratios (30%-40%), the recommended strategy is SNIP + LoRA, as it can almost fully restore the original model performance, representing the solution with minimal accuracy loss. At high target pruning ratios, if the target is 50%, the recommended strategies are Taylor + LoRA or SNIP + LoRA. The former combination demonstrates remarkable recovery effectiveness, while the latter achieves the highest accuracy. However, if the target pruning ratio is 60%, the recommended strategies become Taylor + LoRA or L_1-Norm + LoRA, as they can still restore accuracy to over 75%, demonstrating stronger robustness. In contrast, SNIP appears to damage the most fundamental structure of the model under extremely high pruning ratios, making it impossible for LoRA to effectively recover its accuracy.

In summary, the conclusions of this study both support and supplement existing research. It has been successfully obtained a performance comparison of the three pruning methods—L_1-Norm, Taylor, and SNIP—across target pruning ratios of 30% to 60%, and introduced LoRA fine-tuning into the pruning recovery phase. This demonstrates that LoRA, as a universal and efficient recovery method, can significantly mitigate the performance loss caused by different pruning techniques, providing important insights for future research. The final selection of a pruning strategy should be based on practical application requirements. SNIP + LoRA and Taylor + LoRA are superior at target pruning ratios of 30%-50%. At a target pruning ratio of 60%, although the accuracy of SNIP + LoRA cannot be salvaged, the accuracy of L_1-Norm + LoRA and Taylor + LoRA can still be restored to a trustworthy range through LoRA fine-tuning.

REFERENCES

[1] M. Gethsiyal Augasta and T. Kathirvalavakumar, "Pruning algorithms of neural networks—a comparative study," Cent. Eur. J. Comp. Sci., vol. 3, no. 3, pp. 105–115, 2013.

[2] F. Hu, J. Zhong, S. Gao, Y. Lin, W. Zhou, and R. Wang, "An efficient training-from-scratch framework with BN-based structural compressor," Pattern Recognit., vol. 153, p. 115646, 2024.

[3] K. Guo, Y. Li, D. Fu, Y. Zheng, S. Ren, H. Hu, and J. Liang, "Multi-method fusion for convolutional neural network model compression," J. Xidian Univ., pp. 233–241, 2025.

[4] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in Advances in Neural Information Processing Systems, Denver, CO, USA, 1993, pp. 164–171.
[5] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in International Conference on Learning Representations (ICLR), Toulon, France, Apr. 2017.
[6] J. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 2017, pp. 5058–5066.
[7] Z. Zhou, Z. Zhao, D. Cheng, Z. Wu, J. Gui, Y. Yang, F. Wu, Y. Cheng, and H. Fan, "Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts LLMs," in Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, Nov. 4–9, 2025.
[8] R. Zhu, J. Zhang, J. Huang, R. Kang, and K. Chen, "Research progress on crop leaf disease detection based on convolutional neural networks," Trans. Chin. Soc. Agric. Eng., vol. 41, no. 17, pp. 15–28, 2025.
[9] X. Wang, P. Liu, S. Xiang et al., "Unstructured pruning method based on neural architecture search," Pattern Recognit. Artif. Intell., vol. 36, no. 5, pp. 448–458, 2023.
[10] P. Zhang, C. Tian, L. Zhao, and Z. Duan, "A multi-granularity CNN pruning framework via deformable soft mask with joint training," Neurocomputing, vol. 572, p. 127189, 2024.
[11] Y. Lian, P. Peng, K. Jiang, and W. Xu, "Cross-layer importance evaluation for neural network pruning," Neural Netw., vol. 179, no. 11, pp. 1–10, 2024.
[12] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in International Conference on Learning Representations, Toulon, France, 2017.
[13] J. Sun, Y. Zhai, P. Liu, and Y. Wang, "Memristor-based neural network circuit of associative memory with overshadowing and emotion congruent effect," IEEE Trans. Neural Netw. Learn. Syst., pp. 1–14, 2024.
[14] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in Advances in Neural Information Processing Systems, Montreal, Canada, 2015, pp. 1135–1143.
[15] N. Lee, T. Ajanthan, and P. H. S. Torr, "SNIP: Single-shot network pruning based on connection sensitivity," in International Conference on Learning Representations, New Orleans, LA, USA, 2019.
[16] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in Proceedings of the 10th International Conference on Learning Representations (ICLR), Virtual Conference, Apr. 25–29, 2022.
[17] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. V. Guttag, "What is the state of neural network pruning?" Proc. Mach. Learn. Syst., vol. 2, pp. 129–146, 2020.
[18] Y. Mao, Y. Ge, Y. Fan, W. Xu, Y. Mi, Z. Hu, and Y. Gao, "A survey on LoRA of large language models," Front. Comput. Sci., vol. 19, no. 3, p. 195349, 2025.
[19] X. Jin, K. Wang, D. Tang, Y. Zhang, and Q. Liu, "Conditional LoRa parameter generation," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, Aug. 11–16, 2024.
[20] H. Jing, Q. Sun, Z. Dang, and H. Wang, "Intention recognition of space noncooperative targets using large language models," Space: Sci. Technol., vol. 5, p. 0271, 2025.