# Efficient Large Language Model Fine-Tuning with Joint Structural Pruning and Parameter Sharing

Rui Wang , Yumin Chen , Mengmeng Liu , Guiran Liu , Binrong Zhu , Wuyang Zhang *

Posted Date: 18 September 2025

doi: 10.20944/preprints202509.1618.v1

Keywords: model compression; parameter sharing; language model fine-tuning; structural sparsity

 doi:10.20944/preprints202509.1618.v1



*Article*

# Efficient Large Language Model Fine-Tuning with Joint Structural Pruning and Parameter Sharing

**Rui Wang [1], Yumin Chen [1], Mengmeng Liu [2], Guiran Liu [3], Binrong Zhu [3] and Wuyang Zhang [4,*]**

[1] Carnegie Mellon University, Pittsburgh, USA
[2] Johns Hopkins University, Baltimore, USA
[3] San Francisco State University, San Francisco, USA
[4] University of Massachusetts Amherst, Amherst, USA
* Correspondence: noctis@umass.edu

**Abstract**

This paper addresses the challenges of high computational cost and severe parameter redundancy in the fine-tuning of large language models. It proposes an efficient fine-tuning algorithm that integrates structural pruning with parameter sharing. The method operates from both the architectural and optimization perspectives. It prunes redundant connections dynamically while keeping the core model frozen and introduces task-conditioned cross-layer sharing modules to enhance representation power and parameter efficiency. A pruning residual compensation mechanism is designed to preserve semantic coherence, and a conditional sharing mapping is constructed to improve task-level consistency. The training objective jointly optimizes task loss, sparsity regularization, and inter-layer consistency constraints, achieving unified parameter compression and semantic retention. The proposed method is systematically evaluated using perplexity, accuracy, and inference speed-up across different pruning rates, learning rates, input lengths, and data distribution settings. Experimental results show that the algorithm consistently outperforms mainstream fine-tuning techniques across multiple dimensions. It achieves joint optimization of accuracy and efficiency with minimal parameter tuning, making it well-suited for large language model deployment and transfer learning across diverse scenarios.

**Keywords:** model compression; parameter sharing; language model fine-tuning; structural sparsity

## I. Introduction

Large language models have achieved remarkable progress in the field of natural language processing. They are widely used in tasks such as text generation, question answering, and information extraction. However, these models often contain tens or even hundreds of billions of parameters, leading to extremely high demands on computational resources for training and fine-tuning[1,2]. This poses a significant barrier to deployment in resource-constrained environments. In practical scenarios involving multiple tasks and domains, frequent model fine-tuning has become the norm. Reducing computation and storage costs without compromising performance is now a critical challenge in both academia and industry. To address this, model compression and efficient fine-tuning techniques have received increasing attention and have become key directions for the sustainable development of large language models[3].

Traditional model compression methods focus mainly on parameter pruning, knowledge distillation, and quantization. However, these approaches usually require additional training stages or redundant reconstruction modules, making them less flexible and less suitable for real-world deployment. In contrast, parameter-efficient fine-tuning methods such as adapter modules and low-rank decomposition introduce a small number of trainable parameters on top of pre-trained models. This allows quick adaptation to new tasks without altering the main structure. Nevertheless, current methods often face limitations in architectural design or information transfer ability. Balancing

 doi:10.20944/preprints202509.1618.v1



compression ratio, expressive power, and transfer generalization remains challenging. In this context, a dual-driven strategy that integrates structural pruning and parameter sharing offers a new paradigm for efficient fine-tuning of large models[4].

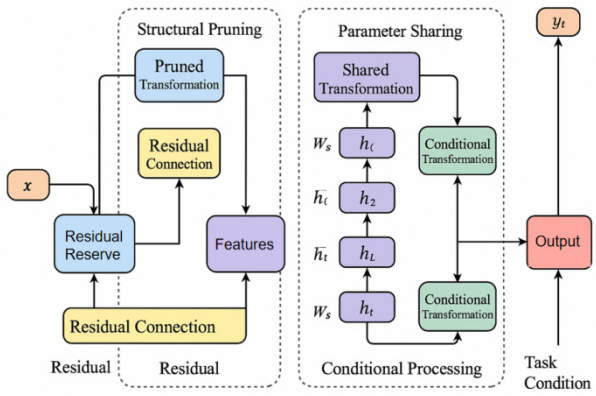
Structural pruning is a compression method that directly targets the model's topology. It can significantly reduce model size and computation while preserving its original functionality. By identifying and removing redundant neurons or substructures, pruning improves inference speed and is naturally friendly to hardware deployment. This makes it suitable for edge devices and heterogeneous computing environments. Additionally, the sparse patterns created through pruning provide insights into the model's internal representation mechanisms. This helps guide further structural optimization. Complementarily, parameter sharing encourages different layers or modules to share weights or representation vectors. This reduces parameter redundancy, enhances consistency, and stabilizes training. In multi-task learning and cross-domain modeling, this approach shows strong structural transfer capabilities and improves generalization in complex settings[5].

In the fine-tuning of large language models, combining structural pruning with parameter sharing achieves both parameter and computation reduction. It also significantly lowers the task-specific tuning cost. Structural pruning dynamically removes redundant structures, freeing up capacity to accommodate new knowledge. Parameter sharing maintains continuity and stability in knowledge transfer[6,7]. Together, they build an efficient fine-tuning framework that balances compactness and expressiveness. This dual strategy breaks the limitations of single-dimensional optimization. It offers theoretical and practical support for constructing generalized, low-overhead, and high-fidelity fine-tuning paradigms. In real-world scenarios where generalization and resource efficiency often conflict, this approach has strong research value and wide application prospects.

Furthermore, as large models are increasingly applied in sensitive sectors such as finance, healthcare, education, and public services, their adaptability and controllability become more crucial. Rapid adaptation to diverse scenarios, ensuring data security, and optimizing resource allocation are now key challenges to scaling intelligent services[8]. The lightweight, efficient, and generalizable characteristics of structural pruning and parameter sharing align with the future needs of edge, customized, and sustainable AI systems. Research into the deep integration and joint optimization of these two mechanisms represents a systematic breakthrough in fine-tuning strategies. It also offers a promising path toward equitable computing and inclusive intelligence.

## II. Proposed Methodology

This study proposes an efficient fine-tuning framework for large language models that integrates structural pruning and parameter sharing mechanisms. The core idea is to achieve efficient adaptation to downstream tasks through sparse structure and module-level sharing optimization without changing the core capabilities of the pre-trained model. First, a pruning strategy based on gradient importance is introduced at the structural level, and redundant units are dynamically screened by evaluating parameter sensitivity. The overall model architecture is shown in Figure 1.



 doi:10.20944/preprints202509.1618.v1



**Figure 1.** Overall model architecture.

Specifically, let the original model parameters be a matrix $W \in R^{m \times n}$, and define the importance of each weight as the absolute value of the product of its corresponding gradient and weight, that is:

$$S_{ij} = \left| \frac{\vartheta L}{\vartheta W_{ij}} \cdot W_{ij} \right| \tag{1}$$

Where L represents the loss function of the current task, and $S_{ij}$ measures the necessity of retaining the parameter $W_{ij}$. Then, pruning is performed according to the global threshold $\tau$, and connections that satisfy $S_{ij} \geq \tau$ are retained to form a sparse connection matrix $\widetilde{W}$.

To enhance the stability of the model structure and the coherence of the information path after pruning, this paper introduces a residual pruning retention mechanism, that is, retaining part of the original features after pruning the main path for linear compensation. Let x be the input feature, and $f(\widetilde{W}, x)$ be the transformed output after pruning, then the final residual output is:

$$y = f(\widetilde{W}, x) + \alpha x \tag{2}$$

Where $\alpha \in [0,1]$ is the residual weight coefficient, which is used to balance the proportional relationship between retained information and new features. This mechanism effectively alleviates the performance drop caused by pruning and improves the dynamic response of parameters during fine-tuning.

In terms of parameter sharing, the model reduces redundant representations by constructing cross-layer shared mappings. Let $h^{(l)} \in R^d$ represent the hidden representation of a layer $l$, then define a shared weight block $W_s \in R^{d \times d}$, and multiple layers share the same transformation function, that is:

$$h^{(l+1)} = \sigma(W_s h^{(l)} + b) \tag{3}$$

Where $\sigma(\cdot)$ represents the nonlinear activation function, and b is the shared bias. This mechanism not only reduces the size of adjustable parameters but also improves the representation consistency and task migration capabilities between different layers. In addition, in multi-task scenarios, a conditional sharing mechanism is further introduced to define the task condition matrix $C_t \in R^{d \times d}$ so that the shared mapping can be dynamically adjusted with the task:

$$h_t^{(l+1)} = \sigma((W_s \otimes C_t) h_t^l + b) \tag{4}$$

Where $\otimes$ represents the Hadamard product, which realizes the dynamic reconstruction of shared features based on the task context.

To coordinate structural compression and information preservation as a whole, this method jointly introduces sparsity regularization terms and shared consistency constraints during the optimization process. The final loss function is defined as:

$$L_{total} = L_{task} + \lambda_1 |\widetilde{W}|_1 + \lambda_2 \sum_l || h^{(l)} - h^{(l+1)} ||^2 \tag{5}$$

Where $L_{task}$ represents the specific task loss, the first term is the pruning sparsity control term, and the second term is the smoothing regularization term between shared layers, which is used to enhance the consistency of representations between different layers. By jointly optimizing the above

 doi:10.20944/preprints202509.1618.v1



objectives, the model ensures semantic preservation and cross-task adaptability while maintaining the compression rate. The overall method has the dual characteristics of structural flexibility and semantic robustness, providing an efficient and generalizable solution for fine-tuning large models in resource-sensitive scenarios.

## III. Dataset and Evaluation Results

### *A. Dataset*

This study adopts the OpenWebText dataset as the primary corpus for fine-tuning large language models. The dataset is a general-purpose English text collection, constructed following standard open-source community procedures. Content is selected from high-quality public web pages, with low-quality texts, duplicate segments, and advertisement noise removed. The final corpus covers a wide range of domains, including news articles, popular science, technical documents, and literary texts. It offers rich linguistic diversity and broad semantic coverage.

OpenWebText, a UTF-8 encoded plain text corpus exceeding 40 GB with tens of millions of paragraphs, preserves paragraph logic and contextual coherence, making it suitable for training medium- to large-scale language models with average lengths of 300–500 tokens. To meet fine-tuning requirements for structural pruning and parameter sharing, this study applied preprocessing by removing redundant samples and filtering sequences to 256–1024 tokens, ensuring exposure to complex semantic patterns while maintaining consistency for parameter sharing. As a widely used pretraining benchmark, OpenWebText provides both generality and reproducibility, supporting contextual reasoning and long-text modeling in experimental evaluation.

### *B. Experimental Results*

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

**Table 1.** Comparative experimental results.

| Model | Trainable | Perplexity | Avg Acc. | Speedup |
|---|---|---|---|---|
| LoRA[9] | 1.80 | 17.26 | 84.3 | 1.00 |
| PrunePEFT[10] | 1.35 | 18.91 | 82.5 | 1.52 |
| IFPruning[11] | 1.20 | 20.04 | 81.2 | 1.68 |
| AdaLoRA[12] | 1.65 | 16.84 | 85.1 | 1.13 |
| Ours | 1.25 | 15.97 | 86.4 | 1.79 |

Our method requires only 1.25% trainable parameters, significantly less than LoRA (1.80%), AdaLoRA (1.65%), and pruning-based approaches, demonstrating superior efficiency in parameter compression while keeping the backbone frozen to ease deployment in resource-limited settings. It achieves the best perplexity (15.97 vs. IFPruning's 20.04), confirming that joint structural pruning and parameter sharing preserve and enhance semantic representation. With the highest average accuracy (86.4%) compared to LoRA (84.3%) and AdaLoRA (85.1%), the method shows strong adaptability and cross-task generalization. Additionally, it delivers a 1.79× inference speedup—far exceeding LoRA (1.00) and AdaLoRA (1.13)—by simplifying inference paths and reducing redundant computations. Overall, the dual-driven framework balances compression, efficiency, and expressive capacity, with pruning rate effects detailed in Figure 2.

 doi:10.20944/preprints202509.1618.v1



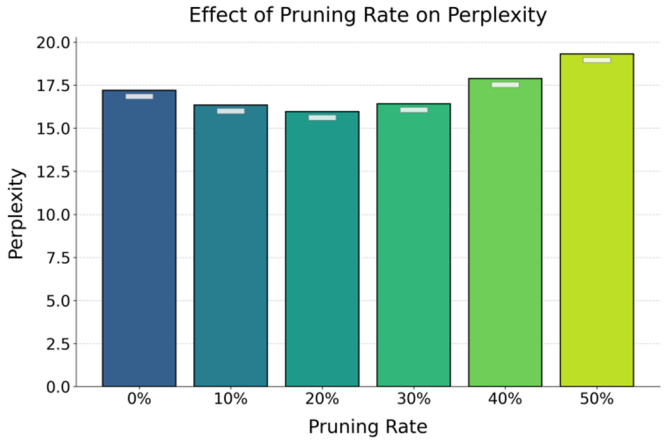



**Figure 2.** Effects of different pruning rates on model performance.

The figure shows that perplexity decreases as pruning increases up to 20%, indicating that moderate pruning removes redundancy, mitigates overfitting, and improves generalization, while pruning beyond 30% steadily degrades performance, with perplexity nearing 20 at 50% due to the loss of key dependency structures. Within the 10%–30% range, perplexity remains stable, confirming the robustness of the pruning mechanism supported by residual preservation and sparsity guidance, which maintain semantic integrity under compression. These findings demonstrate that structural pruning enhances performance within a controlled range while reducing resource cost, and when combined with parameter sharing, the dual-driven framework maximizes model capacity in low-resource settings, offering both theoretical support and practical guidance for large-scale model fine-tuning and deployment. The impact of input text length on perplexity is further illustrated in Figure 3.

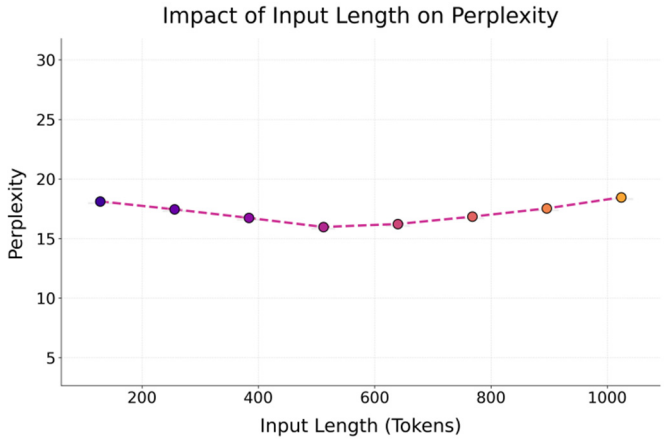



**Figure 3.** The impact of input text length changes on perplexity.

The figure shows that model perplexity follows a U-shaped trend with input length, reaching its lowest point around 512 tokens, where contextual information is effectively leveraged without the drawbacks of short texts lacking content or long texts suffering from representational dilution. Perplexity is higher below 384 tokens due to insufficient information for stable context modeling, indicating that even under pruning the model requires adequate input density to preserve semantic integrity. Beyond 640 tokens, perplexity rises again, reflecting memory decay, attention dispersion, and sparse structure limitations, as well as constraints of parameter sharing for long texts. These findings highlight the need for dynamic fine-tuning strategies that adapt pruning and sharing to input characteristics, offering guidance for input-aware compression scheduling in resource-constrained deployments. Sensitivity analysis of the learning rate is further presented in Figure 4.

 doi:10.20944/preprints202509.1618.v1



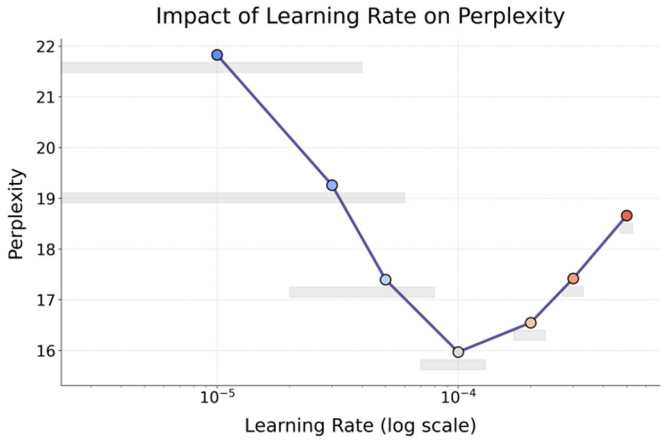



**Figure 4.** Evaluation of the sensitivity of the learning rate to the proposed algorithm.

The figure shows that perplexity decreases as the learning rate increases from $1\times10^{-5}$ to $1\times10^{-4}$, reaching its optimum at $1\times10^{-4}$, indicating that a moderate rate enables effective optimization under pruning and parameter sharing by enhancing semantic modeling and training stability. At $1\times10^{-5}$, perplexity stays high (~22) due to overly conservative updates and weak gradient effects in reduced parameter spaces, leading to poor convergence. Beyond $2\times10^{-4}$, perplexity rises again as excessive updates destabilize sparse structures and harm generalization, highlighting the strong coupling between learning rate and model structure. Overall, these results underscore that careful learning rate selection is crucial for unlocking the dual-driven framework's potential, and practical deployment should adapt rate strategies to task demands and hardware constraints to balance compression efficiency with representational quality.

## IV. Conclusion

This paper addresses the issues of high resource consumption and parameter redundancy in the fine-tuning of large language models. It proposes an efficient fine-tuning framework that integrates structural pruning with parameter sharing. By dynamically removing redundant structures and introducing condition-aware sharing strategies, the method significantly reduces training cost and inference load while preserving the model's representational capacity. Systematic evaluations across multiple dimensions confirm that the framework achieves a strong balance between parameter efficiency, modeling accuracy, and computational acceleration. This offers a new solution for the current paradigm of large model fine-tuning.

From a methodological perspective, structural pruning and parameter sharing are designed as complementary mechanisms that operate in close coordination. Structural pruning guides the model to perceive input paths sparsely, enhancing its focus on task-relevant regions. Parameter sharing introduces weight coordination and task-conditioned modulation across modules, which helps maintain semantic consistency and generalization robustness under limited resources. The combined compression and expressiveness advantages of both techniques address common issues such as overfitting, low efficiency, and poor transferability during fine-tuning. This also lays a structural foundation for building unified and modular, efficient language models.

The proposed fine-tuning framework shows strong transferability and inference efficiency across multi-task and multi-domain language modeling, making it highly applicable to real-world settings such as finance, medical text analysis, legal retrieval, and cross-domain QA, where efficiency and reliability are essential. Its adjustable sparsity constraints and sharing strategies provide flexibility for task-adaptive compression, while future research could explore pruning-based path search, dynamic parameter distillation, and reinforcement or meta-learning to enhance structural and parametric control. Extending the approach to multimodal or graph-based models would further improve resource adaptability and broaden its applicability to scenarios like on-device computing, federated learning, and privacy-aware fine-tuning.

 doi:10.20944/preprints202509.1618.v1



## References


1. Liu Y, Yang H, Chen Y, et al. PAT: Pruning-Aware Tuning for Large Language Models[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2025, 39(23): 24686-24695.
2. Wang L, Chen S, Jiang L, et al. Parameter-efficient fine-tuning in large language models: a survey of methodologies[J]. Artificial Intelligence Review, 2025, 58(8): 227.
3. Ma X, Fang G, Wang X. Llm-pruner: On the structural pruning of large language models[J]. Advances in neural information processing systems, 2023, 36: 21702-21720.
4. Lu L, Wang Z, Bao R, et al. All-in-one tuning and structural pruning for domain-specific llms[J]. arXiv preprint arXiv:2412.14426, 2024.
5. Zhang M, Chen H, Shen C, et al. LoRAPrune: Structured pruning meets low-rank parameter-efficient fine-tuning[J]. arXiv preprint arXiv:2305.18403, 2023.
6. Lu X, Zhou A, Xu Y, et al. Spp: Sparsity-preserved parameter-efficient fine-tuning for large language models[J]. arXiv preprint arXiv:2405.16057, 2024.
7. Gao S, Lin C H, Hua T, et al. Disp-llm: Dimension-independent structural pruning for large language models[J]. Advances in Neural Information Processing Systems, 2024, 37: 72219-72244.
8. Sanh V, Wolf T, Rush A. Movement pruning: Adaptive sparsity by fine-tuning[J]. Advances in neural information processing systems, 2020, 33: 20378-20389.
9. Hu E J, Shen Y, Wallis P, et al. Lora: Low-rank adaptation of large language models[J]. ICLR, 2022, 1(2): 3.
10. Yu T, Zhang Z, Zhu G, et al. PrunePEFT: Iterative Hybrid Pruning for Parameter-Efficient Fine-tuning of LLMs[J]. arXiv preprint arXiv:2506.07587, 2025.
11. Hou B, Chen Q, Wang J, et al. Instruction-Following Pruning for Large Language Models[J]. arXiv preprint arXiv:2501.02086, 2025.
12. Zhang Q, Chen M, Bukharin A, et al. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning[J]. arXiv preprint arXiv:2303.10512, 2023.