Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 8, 2025

Structured Pruning-Driven High-Expressive Low-Rank Adapter Learning for Large Language Models

Ourong Lin1*, Gregory Brennan[2], Xiaosi Xu[2]
[1]University of California, Los Angeles, Los Angeles, USA
[2]University of Southern California, Los Angeles, USA
*Corresponding author: Ourong Lin; hughlinor@gmail.com

Abstract: This paper addresses the challenge of balancing expressive power and resource overhead in efficient fine-tuning of large language model parameters. It proposes a structured pruning-driven high-expressive low-rank adapter learning method. By freezing the main model parameters, low-rank incremental updates are injected only into key linear transformation layers to achieve lightweight task adaptation. To overcome the expression bottleneck caused by fixed low rank, the method employs residual injection to controllably adjust hidden representations and introduces differentiable gating vectors in the low-rank channel dimension, transforming adapter capacity allocation into a learnable structural selection process, thereby achieving interpretable sparsity at the structural unit granularity. Furthermore, a unified optimization objective is constructed, consisting of the task objective, structured sparsity regularization, and budget constraints. Channel-level structured pruning is achieved using gating sparsity induction, and the overall trainable capacity is controlled by budget consistency constraints, forming a sparse, low-rank integrated parameterization and training mechanism. This framework organically integrates structured sparsity and low-rank representation, improving the effective expression density of the update subspace while maintaining parameter efficiency. It provides a controllable, well-structured, and easily implementable technical solution for adapter fine-tuning in resource-constrained deployments.

Keywords: Structured gating; sparse budget constraints; effective rank; incremental parameterization

1. Introduction

Large language models demonstrate powerful versatility in natural language understanding and generation tasks, but their continuously growing parameter size puts significant pressure on full-parameter fine-tuning in terms of computational power consumption, storage usage, and deployment costs. In multi-task scenarios and with rapid iteration requirements, frequent full-parameter updates not only incur high training and maintenance costs but also increase the complexity of model version management and online updates [1]. Therefore, efficient parameter fine-tuning for large language models has gradually become an important technical path to improve deployment efficiency and reduce resource barriers. Its goal is to achieve effective adaptation to downstream tasks with as few trainable parameters as possible, while maintaining the model's expressive and generalization capabilities within a controllable cost [2].

Among efficient parameter fine-tuning methods, low-rank adapters, by imposing low-dimensional constraints on the weight update space, can adjust model behavior with a relatively small number of training parameters and are therefore widely adopted. However, the fixed low-rank assumption often implicitly assumes a limited dimension of the update subspace. When the task distribution is complex or the semantic transfer span is large, overly strong low-rank constraints may create an expressive bottleneck, manifesting as insufficient ability to characterize key semantic differences. Meanwhile, blindly increasing the rank to enhance expressiveness will rapidly increase the number of parameters and computational overhead, weakening the core advantage of efficient parameter fine-tuning. The resulting contradiction lies in the need to improve the expressiveness of the updated representation within a limited parameter budget, while simultaneously controlling redundancy and unnecessary degrees of freedom at the structural level, enabling adapter learning to achieve a more robust balance between efficiency and capability [3, 4].

Structured pruning provides a way to compress and redistribute the model's learnable capabilities at the structural level. Unlike unstructured sparsity, structured pruning reduces at the channel, head, block, or sub-module level, making it easier to achieve considerable speedup gains in practical hardware and inference frameworks, and contributing to a more interpretable parameter allocation pattern. Introducing structured pruning into adapter learning means no longer relying solely on low-rank constraints to control parameter size, but actively identifying and removing lower-contributing structural units through pruning mechanisms, thereby concentrating the limited parameter budget on more critical update directions [5]. If pruning and low-rank learning can be designed collaboratively, it is hoped that an adaptive capacity allocation mechanism oriented towards task requirements can be achieved, enabling the adapter to maintain a lightweight design while possessing higher effective expressive density.

Based on the aforementioned motivations, structured pruning-driven high-representation low-rank adapter learning

holds significant research importance. This research direction aims to use structured sparsity as a capacity scheduling signal, combined with low-rank parameterization, to achieve precise shaping of the update subspace, thereby obtaining stronger task adaptability and more stable training behavior with the same or lower trainable parameter scale [6]. For practical applications, this type of method can reduce fine-tuning and deployment costs, improve usability in environments with rapid multi-task switching and resource constraints, and provide more scalable technical support for high-frequency iteration of large language models in industry scenarios. Simultaneously, the combination of structured constraints and low-rank representations provides a new perspective for understanding the effective structure of parameter updates in large models, contributing to the further development of efficient parameter fine-tuning methods in terms of interpretability and controllability [7].

Recent studies on efficient and stable model adaptation provide useful methodological context for structured pruning-driven low-rank learning. Semantic energy-guided stable fine-tuning treats distribution control as an explicit part of large language model adaptation, highlighting the need to keep representation updates stable while improving controllability [8]. Retrieval-augmented long-text reasoning with semantic conflict detection further shows that long-context evidence integration can strengthen reasoning reliability when semantic signals are distributed across multiple passages [9]. These studies support the motivation of constraining adapter updates through interpretable structural mechanisms rather than relying only on unconstrained parameter growth.

Work on multi-scale representation and resource-aware learning also informs the proposed adapter design. Multi-scale feature decoupling with boundary-aware diagnosis demonstrates the value of separating feature components while preserving decision boundaries under complex morphology [10], and compressed representation learning for edge anomaly detection emphasizes lightweight feature extraction under strict resource constraints [11]. Hierarchical communication and computation scheduling for cloud-edge collaborative intelligent systems further connects model design with deployment-side coordination and resource allocation [12]. Together, these studies suggest that high-expressive low-rank adapters should jointly consider feature separation, compression efficiency, structural budget control, and practical deployment cost.

2. Datasets and Dataset Preprocessing

2.1 Dataset

This paper uses the OpenAssistant Conversations Dataset OASST1 as the corpus for instruction fine-tuning. This open-source dataset is designed for general assistant-style interaction scenarios, covering various instruction types such as question answering, writing, reasoning, and multi-domain task requests. It provides a structured instruction-to-response mapping foundation for efficient parameter fine-tuning. The dataset is licensed under Apache 2.0, facilitating reproduction and expansion in research and engineering practice, and maintaining good compatibility with reusable training paradigms such as adapter learning and structured sparsity.

2.2 Dataset Preprocessing

To meet the learning requirements of low-rank adapters driven by instruction fine-tuning and structured pruning, a unified preprocessing workflow is implemented for open-source dialogue data, including sample cleaning, context construction, quality control, and formatted export. First, the dialogue tree is parsed, and user/assistant turn pairs are extracted. Empty responses, duplicate fragments, anomalous symbol sequences, and invisible control characters are removed, and encoding and whitespace conventions are standardized. Then, historical turns are concatenated according to the dialogue path to form the input sequence. The current assistant response is used as the supervision target. Fixed delimiters are used to maintain consistent turn boundaries, and excessively long samples are truncated at the end to prioritize the preservation of nearest-neighbor context. Furthermore, quality score signals are used for filtering and reweighting to suppress the interference of low-quality or semantically incomplete samples on the update direction. Bucketing of the length distribution is also performed to mitigate the dominance of extreme samples. Finally, reproducible partitioning is performed at the dialogue granularity to avoid information leakage, and the data is exported in a standard instruction format, retaining metadata such as sample identifiers, turn indices, language, and quality scores for subsequent controllable training configuration under sparse budget constraints and structured gating.

Figure 1 shows a comparison of data characteristics before and after preprocessing from four dimensions: length distribution, quality score distribution, proportion of noisy samples, and length coverage. After preprocessing, the density distribution of sequence length is more concentrated in a reasonable range, and the proportion of long-tailed and extremely long samples is reduced, thus making the input scale more stable and reducing the interference of abnormal samples on the training signal. The quality score distribution shifts to higher segments overall, indicating that low-quality responses are further filtered or downweighted, improving the usability and consistency of the corpus. In the sample state statistics, the proportion of noise categories such as Empty, Duplicate, and Invalid decreases significantly, while the proportion of Clean increases, indicating that the cleaning rules effectively remove empty responses, duplicate segments, and invalid content. The empirical cumulative distribution curve of length rises faster after preprocessing, indicating that a larger proportion of samples can be covered within a shorter length range, and the overall data is more regular, which is conducive to stable learning under batch processing and truncation strategies.

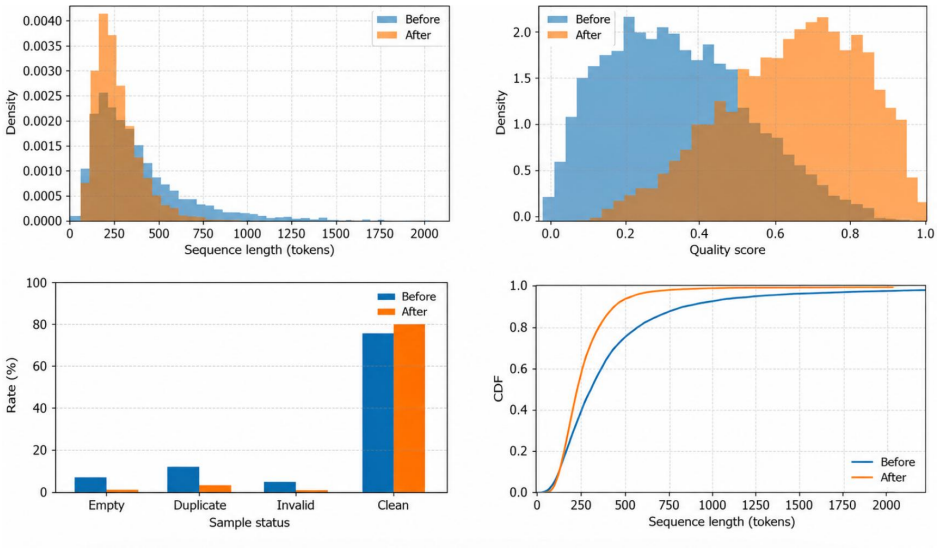

Figure 1. Comparison of dataset preprocessing results

3. Method

In the presence of a parameter set Θ from a pretrained large language model and a downstream supervised dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, parameter-efficient fine-tuning adapts the model to the target task by freezing Θ and learning only a small set of incremental parameters. Let $f(\cdot;\Theta)$ denote the mapping of the pretrained model. The standard autoregressive conditional likelihood objective can be formulated as minimizing the negative log-likelihood loss. In this paper, we introduce learnable adapters only in a subset of linear transformation layers while keeping the remaining parameters fixed. The overall model architecture is also presented in this paper, as shown in Figure 2.

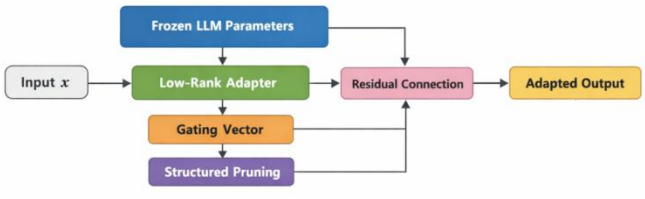

Figure 2. Overall model architecture

This design constrains the number of trainable parameters and the additional computation within a controllable range, and simultaneously provides explicit tunable structural units and measurable capacity-allocation space for subsequent structured pruning.

$$\min_{\Phi} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\log p(y|x;\Theta,\Phi)\right]$$

To enhance representational capacity under a limited parameter budget, we adopt a highly expressive low-rank adapter parameterization and perform controlled adjustments to the original representations via intra-layer residual injection. For the l-th linear mapping $W_l \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$, we introduce a low-rank update ΔW_l and apply it to the input h_l in a residual manner, where $A_l \in \mathbb{R}^{d_{\text{out}}\times r}$, $B_l \in \mathbb{R}^{r\times d_{\text{in}}}$, and $r \ll \min(d_{\text{out}}, d_{\text{in}})$. To increase expression density, the adapter update is scaled in the subspace by a learnable amplitude factor, enabling different layers to adaptively select suitable update strength and improve the effective degrees of freedom without increasing the number of full model parameters.

$$\Delta W_l = A_l B_l h_{l+1} = W_l h_l + \alpha_l \Delta W_l h_l$$

Building upon this, we introduce a structured-pruning-driven mechanism that transforms adapter capacity allocation from a static design choice into a learnable structural selection process. Specifically, we incorporate a gating vector $g_l \in [0,1]^r$ along the structural dimension of the adapter to modulate the activation of low-rank channels in a structured manner, thereby enabling differentiable structural selection at the granularity of rank dimensions. The gate is applied to the rank-channel outputs of B_l, or equivalently, to the intermediate low-dimensional representation, so that each rank channel corresponds to an interpretable structural unit. Subsequently, thresholding can convert continuous gates into discrete keep-or-drop decisions, yielding a sparse low-rank adapter after structured pruning.

$$\Delta W_l(g_l) = A_l \,\text{diag}(g_l)\, B_l$$

To ensure stable progress of pruning during training and explicitly satisfy the parameter-budget constraint, we add structured sparsity regularization and a budget-consistency constraint in addition to the task loss. The structured sparsity term uses the l_1 norm of the gating vectors to promote rank-channel-level sparsification, and an optional entropy-style or squared penalty can be used to suppress intermediate gate states and improve usability after discretization. Furthermore, we control the overall adapter capacity by imposing an upper-bound constraint on the expected retained ranks $\sum_l \|g_l\|_1$, so that the training procedure optimizes under an interpretable

structural budget, forming a structured-pruning-driven learning objective for highly expressive low-rank adapters.

$$\mathcal{L}(\Phi, g) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\log p\left(y|x;\Theta,\Phi,g\right)\right] + \lambda \sum_{l} \| g_l \|_1$$

$$\sum_{l} \| g_l \|_1 \leq R_{\max}$$

4. Experimental Results and Analysis

To comprehensively compare the adaptability and optimization characteristics of structured pruning and low-rank adapter learning methods in scenarios requiring efficient parameter fine-tuning, representative studies in the same direction were selected as references, and Qwen7B was chosen as the base model. A unified evaluation index system was used for cross-sectional characterization. This comparison aims to provide more intuitive quantitative comparison criteria from dimensions such as generation quality, semantic consistency, and matching accuracy, thereby supporting the necessity and rationality of the method design. The experimental results are shown in Table 1.

Table 1. Experimental results compared with other models

Method	PPL	ROUGE-L	BLEU-4	METEOR	BERTScore	Exact Match	Token-F1	Win Rate
Ma et al. [13]	12.45	21.33	5.12	13.44	0.81	23.11	25.77	46.12
Zhang et al. [14]	11.87	22.01	5.66	13.92	0.82	24.02	26.33	47.88
Chen et al. [15]	12.01	21.77	5.44	13.55	0.81	23.56	25.98	47.01
Ding et al. [16]	11.62	22.33	5.71	14.01	0.83	24.44	26.88	48.22
Huang et al. [17]	11.55	22.66	5.80	14.22	0.84	24.78	27.11	48.90
Liu et al. [18]	11.44	22.88	5.91	14.33	0.84	24.99	27.44	49.12
Guo et al. [19]	11.33	23.01	6.02	14.55	0.85	25.22	27.66	49.55
Ours	10.11	24.55	7.88	15.77	0.89	27.44	29.88	52.44

Table 1 shows that the proposed method achieves stronger overall performance across multiple generations and matching metrics, maintaining advantages in language generation quality, semantic consistency, and answer matching accuracy simultaneously. This result demonstrates that structured pruning-driven high-expression low-rank adapter learning can generate more effective update representations under efficient parameter fine-tuning, reducing noisy updates caused by invalid degrees of freedom, and concentrating limited trainable capacity on more critical task-related directions, thereby improving overall output quality and stability.

Furthermore, the proposed method outperforms the perplexity metric, indicating that the model more fully models the target distribution, generating data that is more consistent with patterns and more predictable. Combined with the simultaneous improvement in multiple dimensions, it can be inferred that the method achieves more reasonable capacity allocation and higher expression density during optimization, enabling the adapter to maintain a lightweight design while possessing stronger task adaptability and better generation reliability.

Adapter updates typically exhibit low-dimensional dominance, but structured gating further alters the shape of its available subspace. By performing singular value spectral decomposition on the incremental weights of the adapter, the effective rank and energy concentration of different layers can be intuitively characterized. The experimental results are shown in Figure 3.

This figure characterizes the structured low-rank update features learned by the proposed method from four perspectives: spectral decay pattern, cross-layer energy distribution, effective rank variation, and cumulative energy aggregation. The singular value spectrum of the representative layer shows that it dominates at low indices and decays rapidly to near zero at high indices, indicating that the update energy is mainly concentrated on a few principal components, consistent with the goal of high-expression, low-rank adaptation. The cross-layer heatmap further shows that the energy concentration varies across different layers, with some layers showing greater concentration at the first few singular values, reflecting the inter-layer selectivity of capacity allocation by the structured gating. The effective rank curve shows that the hierarchical update complexity is not uniformly distributed, but rather lower or higher in some layers, reflecting adaptive structure selection under sparse budget constraints. The corresponding cumulative energy curve achieves high coverage within a relatively small index range, verifying that the update representation has obvious low-dimensional dominance and energy aggregation characteristics, thus supporting the ability of structured pruning-driven adapter learning to form a more compact and controllable update subspace.

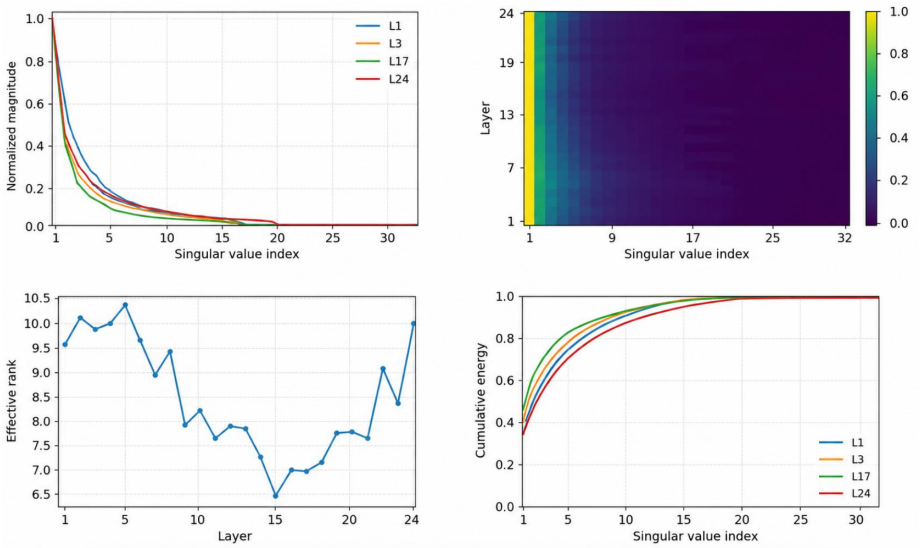

Figure 3. The method presented in this paper describes the singular value spectrum characteristics and effective rank characterization of the incremental weights of the low-rank adapter. The top left shows the normalized singular value spectrum curves of several representative layers, and the top right shows a cross-layer spectral heatmap used to characterize the energy distribution patterns of different layers. The bottom left shows the hierarchical effective rank changes calculated based on spectral entropy, and the bottom right shows the cumulative energy curves of representative layers used to reflect the energy aggregation process of principal components.

5. Conclusion

It proposes a structured pruning-driven high-representational low-rank adapter learning framework. Under the premise of freezing the main parameters, lightweight adaptation is achieved through low-rank incremental updates, and structured gating is introduced to transform capacity allocation from a static setting to a learnable structural selection process. This method organically combines structured sparsity constraints with low-rank parameterization, enabling the adapter to achieve higher effective representational density within a limited parameter budget. Simultaneously, it achieves more controllable parameter compression and deployment friendliness by using interpretable structural units as the granularity. This approach provides a unified modeling perspective for efficient parameter fine-tuning, namely, improving the utilization efficiency of available capacity through the structured shaping of the update subspace. This provides a scalable technical path for the rapid iteration and large-scale deployment of large models in resource-constrained environments. At the application level, the proposed framework can reduce the training and storage costs of model customization in industry scenarios and improve maintainability under conditions of frequent multi-task switching and rapid version updates. It has direct engineering significance for applications requiring high-frequency adaptation, such as financial risk control, text understanding, medical document generation and summarization, government and enterprise knowledge question answering, compliance review, and intelligent customer service.

Looking to the future, structured pruning-driven adapter learning is expected to become a crucial foundational module for the controllable adaptation of large models, synergizing with broader system-level optimization. On one hand, gating and budget constraints can be extended to finer-grained or more diverse structural units, making capacity allocation strategies more closely aligned with the differentiated needs of various tasks for attention interactions and feedforward transformations. This enhances cross-domain transfer and continuous adaptation capabilities while maintaining a lightweight design. On the other hand, combining hardware-friendly structured sparse execution with edge deployment requirements on the inference side creates an integrated parameter efficiency loop from training to inference, further improving throughput and latency performance in practical deployments. As the application of large language models continues to deepen in key industries, the demand for stable, interpretable, and controllable fine-tuning strategies will become increasingly prominent. The fusion paradigm of structured pruning and low-rank adaptation provides a more universal direction for building a low-cost, sustainably iterative, and engineering-constrained large model customization system, and is expected to drive more robust upgrades in efficiency, reliability, and governance for related applications.

References

[1] T. Dettmers, A. Pagnoni, A. Holtzman, et al., "QLoRA: Efficient Finetuning of Quantized LLMs," Advances in Neural Information Processing Systems, vol. 36, pp. 10088-10115, 2023.

[2] Q. Zhang, M. Chen, A. Bukharin, et al., "AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning," arXiv preprint arXiv:2303.10512, 2023.

[3] S. Y. Liu, C. Y. Wang, H. Yin, et al., "DoRA: Weight-Decomposed Low-Rank Adaptation," in Forty-first International Conference on Machine Learning, 2024.

[4] D. J. Kopiczko, T. Blankevoort, and Y. M. Asano, "VeRA: Vector-Based Random Matrix Adaptation," arXiv preprint arXiv:2310.11454, 2023.

[5] R. Zhang, J. Han, C. Liu, et al., "LLaMA-Adapter: Efficient Fine-Tuning of Language Models with Zero-Init Attention," arXiv preprint arXiv:2303.16199, 2023.

[6] Z. Wu, A. Arora, Z. Wang, et al., "ReFT: Representation Finetuning for Language Models," Advances in Neural Information Processing Systems, vol. 37, pp. 63908-63962, 2024.

[7] M. Sun, Z. Liu, A. Bair, et al., "A Simple and Effective Pruning Approach for Large Language Models," arXiv preprint arXiv:2306.11695, 2023.

[8] H. Zhuang and A. Shen, "Semantic Energy-Guided Stable Fine-Tuning for Distribution-Controlled Generation in Large Language Models," 2025.

[9] S. Y. Huang, "Retrieval-Augmented Long-Text Reasoning with Semantic Conflict Detection and Cross-Paragraph Evidence Integration," 2024.

[10] K. Zeng, "Skin Disease Classification Algorithm Based on Multi-Scale Feature Decoupling and Boundary-Aware Diagnosis for Complex Lesion Morphology Recognition," 2024.

[11] X. Zhang and A. He, "Lightweight Anomaly Detection for Edge Intelligence through Compressed Representation Learning," 2025.

[12] R. Meng, "Hierarchical Communication and Computation Scheduling for Efficient Federated Learning in Cloud-Edge Collaborative Intelligent Systems," 2024.

[13] X. Ma, G. Fang, and X. Wang, "LLM-Pruner: On the Structural Pruning of Large Language Models," Advances in Neural Information Processing Systems, vol. 36, pp. 21702-21720, 2023.

[14] M. Zhang, H. Chen, C. Shen, et al., "LoRAPrune: Structured Pruning Meets Low-Rank Parameter-Efficient Fine-Tuning," in Findings of the Association for Computational Linguistics: ACL 2024, 2024, pp. 3013-3026.

[15] T. Chen, T. Ding, B. Yadav, et al., "LoRAShear: Efficient Large Language Model Structured Pruning and Knowledge Recovery," arXiv preprint arXiv:2310.18356, 2023.

[16] N. Ding, X. Lv, Q. Wang, et al., "Sparse Low-Rank Adaptation of Pre-Trained Language Models," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 4133-4145.

[17] W. Huang, Y. Zhang, X. Zheng, et al., "Dynamic Low-Rank Sparse Adaptation for Large Language Models," arXiv preprint arXiv:2502.14816, 2025.

[18] Y. Liu, H. Yang, Y. Chen, et al., "PAT: Pruning-Aware Tuning for Large Language Models," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 23, 2025, pp. 24686-24695.

[19] J. Guo, X. Chen, Y. Tang, et al., "SlimLLM: Accurate Structured Pruning for Large Language Models," arXiv preprint arXiv:2505.22689, 2025.