Journal of computer science and software applications

https://www.mfacademia.org/index.php/jcssa

ISSN:2377-0430

Vol. 5, No. 6, 2025

LoRA-Based Lightweight Adaptation of Pretrained Models for Low-Resource Text Summarization

Thayer Ellison
University of North Florida, Jacksonville, USA
tellison87@unf.edu

Abstract:

This study explores the efficient fine-tuning method of the LoRA (Low-Rank Adaptation) algorithm in a low-resource environment, and experimentally evaluates its performance under different data scales and low-rank matrix parameter r settings. For scenarios with limited computing resources, LoRA introduces a low-rank matrix to efficiently adapt the parameters of the pre-trained model, significantly reducing the memory usage and computing requirements. The experimental results show that with only 1% to 10% of the training data, the ROUGE indicator of LoRA fine-tuning is only slightly lower than that of full parameter fine-tuning, verifying its effectiveness in low-resource data environments. At the same time, the analysis of the r value found that a smaller r value can ensure better computational efficiency, while an appropriate increase in the r value can improve the quality of the summary, but when r exceeds a certain threshold, the performance gain tends to stabilize, indicating that LoRA needs to balance model performance and computational overhead. Overall, LoRA provides an efficient and feasible solution for large-scale model fine-tuning in a low-resource environment, and has extensive research value in future applications.

Keywords:

LoRA, low-resource fine-tuning, low-rank matrix, text summarization

1. Introduction

In the wave of artificial intelligence development in recent years, large-scale pre-trained language models (LLMs) have become an important cornerstone for promoting the advancement of natural language processing (NLP) technology. However, such models are often large in scale, with the number of parameters reaching tens of billions, making it difficult to perform efficient parameter fine-tuning in scenarios with limited computing resources. Traditional full-parameter fine-tuning methods not only require extremely high computing resources, but also face huge challenges in data storage and optimization. Therefore, in low-resource environments, how to effectively customize and fine-tune large models with minimal computing costs has become one of the important research directions for the current implementation of artificial intelligence applications. LoRA (Low-Rank Adaptation), as a parameter efficient fine-tuning technology, can achieve efficient adaptation of large models by introducing the idea of low-rank decomposition while only updating a very small number of trainable parameters, thereby providing a feasible solution in low-resource environments[1].

In low-resource scenarios, limited computing power makes it difficult to implement full-parameter fine-tuning methods, and the introduction of LoRA greatly alleviates this problem. LoRA avoids large-scale updates to the original model weights by adding low-rank matrices to specific layers and adjusting only these small-scale parameter matrices during training. This approach not only significantly reduces video memory usage, but also achieves an effect close to full parameter fine-tuning under limited data and computing

ISSN:2377-0430

Vol. 5, No. 6, 2025

conditions, making model adaptation in low-resource environments more feasible. For example, in application scenarios such as edge computing devices, mobile devices, and even embedded systems, LoRA can effectively reduce storage and computing costs, allowing large-scale pre-trained models that were originally difficult to adapt to be widely used. In addition, in industry applications such as financial risk control, medical diagnosis, and intelligent customer service, it is often difficult to fine-tune large models on a large scale due to the need for computing resources or privacy protection[2]. LoRA's low-rank adaptation strategy provides a new solution for such scenarios[3].

In addition to the limitation of computing resources, the lack of data resources is also an important challenge faced by model fine-tuning in low-resource environments. In many practical application scenarios, the cost of obtaining domain-specific data is high, and may even be strictly restricted due to data privacy or compliance requirements. Traditional full parameter fine-tuning usually requires a large amount of high-quality annotated data, while LoRA's low-rank optimization method has relatively low data requirements and can effectively adjust the model's behavior even with a small number of samples, thereby improving its performance on specific tasks[4]. This makes LoRA more adaptable in low-resource environments, especially in low-resource language modeling, few-shot learning, and personalized tasks in specific fields. For example, in medical text analysis, full parameter fine-tuning is difficult to achieve because the acquisition and annotation of high-quality medical data requires professional knowledge, while LoRA can achieve effective domain adaptation with only a small amount of data, improving the accuracy of medical text classification or summary generation[5].

From a technical perspective, LoRA also has strong advantages in multi-task learning and transfer learning scenarios by reducing the amount of parameter updates required during training. In multi-task learning, due to the shared information between multiple tasks, traditional full parameter fine-tuning often leads to parameter redundancy and overfitting problems, while LoRA can learn key features between different tasks in a more compact way, thereby improving the generalization ability of the model. At the same time, in cross-language migration tasks, LoRA can effectively adapt to the characteristics of different languages without retraining the entire model, which greatly reduces the computational cost of cross-language migration. In addition, LoRA's modular design enables it to be efficiently deployed in different model architectures. Whether it is NLP models such as BERT and GPT, or ViT (Vision Transformer) for computer vision, LoRA's optimization strategy can achieve efficient parameter fine-tuning[6].

In summary, as an efficient fine-tuning method, LoRA shows great application potential in low-resource scenarios. It can not only alleviate the challenges brought by limited computing resources, but also achieve efficient model adaptation in data-constrained environments, while showing good generalization capabilities in scenarios such as multi-task learning and cross-language migration. With the continuous development of artificial intelligence technology and the gradual expansion of large model application scenarios, the research and application of LoRA in low-resource environments will become an important direction in the future. By deeply exploring LoRA's optimization strategies and application scenarios, further improving its adaptation efficiency and model performance, it will help promote the implementation of artificial intelligence technology in more fields and provide more efficient and feasible solutions for large model fine-tuning in low-resource environments.

2. Related Work

Recent advancements in time-series modeling have enabled more efficient feature representation and forecasting, particularly with the integration of Transformer architectures. Cheng proposed a multivariate forecasting framework that automates feature extraction and captures long-term dependencies, offering insights valuable for adapting pretrained models to sequential text generation tasks in low-resource

ISSN:2377-0430

Vol. 5, No. 6, 2025

environments [7]. Complementing this, reinforcement learning (RL) methods have gained traction in distributed computing. Duan applied TD3-based continuous control for load balancing, demonstrating how adaptive policy optimization can enhance system performance under resource constraints [8].

Deep probabilistic modeling further expands the toolkit for learning in sparse and noisy data environments. Dai et al. employed mixture density networks for user behavior anomaly detection, enabling a nuanced probabilistic representation of complex patterns [9]. In parallel, Lou introduced a capsule network-based approach tailored for structured data mining, showing how adaptive feature representation contributes to effective learning in compact neural architectures [10]. These works align closely with LoRA's goal of capturing essential task-specific information using minimal model updates.

Decision-making in distributed systems also benefits from graph-structured policy design. Wang proposed a multi-agent reinforcement learning framework that incorporates topological awareness for optimized scheduling [11]. This idea parallels LoRA's ability to focus learning within structured parameter subspaces, making adaptation both efficient and context-aware. Additionally, Cui and Liang developed a diffusion-transformer framework targeting high-dimensional sparse data, emphasizing deep representation learning under constraints, a challenge frequently encountered in low-resource fine-tuning tasks [12].

Temporal dependencies in sequence modeling continue to be a focal point in modern AI systems. Sheng's hybrid LSTM-GRU framework for loan default prediction exemplifies how multiple sequential architectures can be integrated to better capture data dynamics in sparse scenarios [13]. Such strategies are particularly relevant when applying LoRA to tasks like summarization, where context length and temporal consistency are essential.

Beyond recurrent architectures, reinforcement learning remains a powerful tool for controlling model complexity and optimizing computational efficiency. Liu's proposal of reinforcement learning-controlled subspace ensemble sampling enables more flexible data selection and learning, contributing to model robustness in uncertain environments [14]. Similarly, Ren et al. introduced a trust-constrained policy learning mechanism for distributed traffic scheduling, which emphasizes decentralized adaptability-an important consideration for scalable LoRA-based systems [15].

Wang et al. explored A3C reinforcement learning in microservice scheduling, where multi-task learning and policy updating can enhance inference quality without significantly increasing resource demands [16]. This aligns with LoRA's low-rank framework, which seeks to preserve performance by localizing updates to selective model layers. Further support comes from Xing's sequential recommendation model, which integrates time-awareness and convolutional user modeling to encode temporal behavior using efficient structures [17].

Wang and colleagues provided a deeper theoretical re-examination of LoRA itself, proposing a smarter adaptation strategy with reduced redundancy and improved efficiency, offering a valuable advancement in scalable fine-tuning [18]. Addressing data imbalance is also crucial in low-resource adaptation. Lou proposed a probabilistic graphical model combined with variational inference, enabling more balanced training under skewed label distributions [19]. Complementary to this, Wang developed a Markov network classifier with adaptive weighting to mitigate the impact of underrepresented classes during learning [20].

The utility of LoRA and related adaptation strategies extends into multimodal and financial prediction contexts. Liu's multimodal factor models, designed for stock forecasting, leverage cross-source information to improve generalization-a concept analogous to cross-domain LoRA adaptation [21]. Deng's hybrid approach combining association rules with LSTM for network congestion prediction further supports the role of structured and temporal modeling in real-time systems [22]. Finally, Liu et al. tackled market turbulence

ISSN:2377-0430

Vol. 5, No. 6, 2025

and risk control using improved A3C reinforcement learning, reinforcing the relevance of adaptive policy design in dynamically changing environments, which is equally applicable to optimizing LoRA-based inference [23].

3. Methodology: LoRA-Based Low-Rank Parameter Adaptation

In low-resource scenarios, efficient fine-tuning methods need to minimize computational overhead while ensuring model adaptability[24]. LoRA (Low-Rank Adaptation) achieves efficient parameter fine-tuning by introducing low-rank matrices in the key layers of the pre-trained model. Its model architecture is shown in Figure 1.

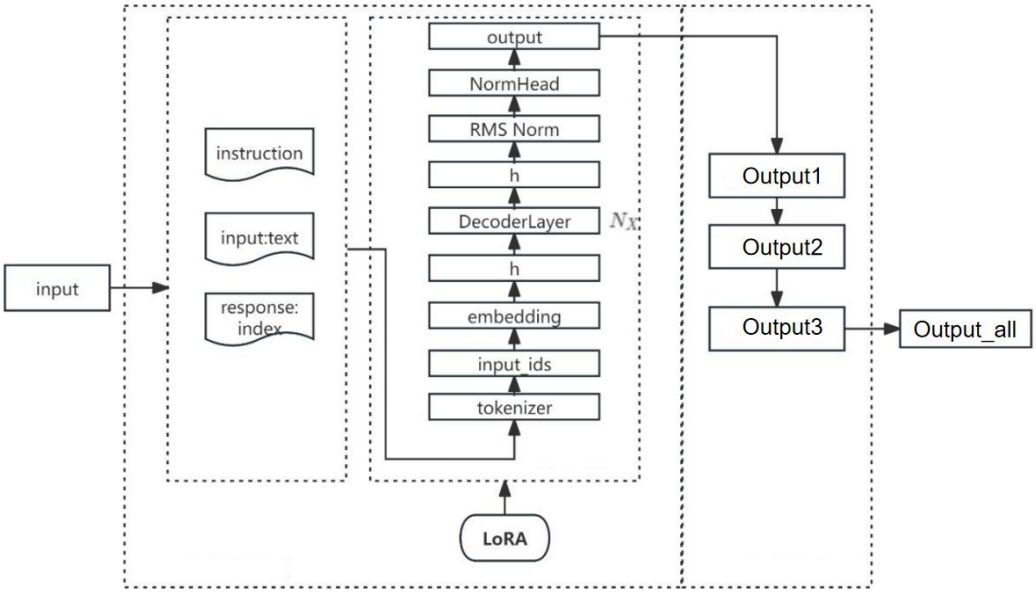

Figure 1. Lora Architecture

Assume that the weight matrix of the original pre-trained model is $W_0 \in R^{d_{out} \times d_{in}}$. The core idea of LoRA is to introduce a trainable low-rank update matrix ΔW to approximate parameter changes without modifying the original weights, that is:

$$W = W_0 + \Delta W \ , \Delta W = AB$$

Among them, $A \in R^{d_{out} \times r}$, $B \in R^{r \times d_{in}}$, and $r << \min(d_{out}, d_{in})$, by limiting the size of R, LoRA is able to significantly reduce the amount of parameters updated during training while maintaining high expressiveness. In addition, LoRA training only involves the optimization of A and B, while W_0 remains frozen, which greatly reduces the computing resources and storage requirements required for fine-tuning.

During the optimization process, LoRA follows the standard gradient descent optimization rule, with the goal of minimizing the task-specific loss function L . The output of the fine-tuned model is defined as $f(x; W_0 + \Delta W)$, and its gradient update only acts on A and B, that is:

ISSN:2377-0430

Vol. 5, No. 6, 2025

$$\frac{\partial L}{\partial A} = \frac{\partial L}{\partial f} \cdot \frac{\partial f}{\partial W} \cdot B^T, \frac{\partial L}{\partial B} = \frac{\partial L}{\partial f} \cdot \frac{\partial f}{\partial W} \cdot A^T$$

Due to the low rank of A and B, this optimization method has much lower computational complexity than full parameter fine-tuning, allowing low-resource devices to efficiently perform gradient updates. In addition, due to the form of parameter decomposition, LoRA can reuse the same W_0 but learn different A, B combinations in multi-task fine-tuning, further reducing storage requirements and improving model scalability.

In the inference phase, the trained low-rank matrices A and B can be directly merged back into the original weights, i.e., $W = W_0 + AB$, without affecting the inference efficiency of the model. Compared with the full parameter fine-tuning method, LoRA avoids a substantial increase in video memory usage, so that an effect close to full parameter fine-tuning can be achieved with limited computing resources. At the same time, LoRA's low-rank structure allows the model to share basic weights between different tasks and only stores the fine-tuned adaptation matrix. This feature is particularly important for devices with limited storage. Therefore, LoRA not only greatly reduces the computing cost in the training phase, but also ensures high efficiency in the inference phase, providing a practical solution for deep learning models in low-resource environments[25].

4. Experimental Evaluation: Data Efficiency and Performance Analysis

This study uses the XSum (Extreme Summarization) dataset for experiments. This dataset is widely used in text summarization tasks and has high-quality annotations and clear semantic structures. XSum consists of British Broadcasting Corporation (BBC) news articles, each of which is accompanied by a single-sentence summary with an average length of about 23 words. The characteristic of this dataset is that its summaries are highly abstract rather than simply extracting the original sentences, so it requires a high generation ability of the model. In a low-resource environment, efficient fine-tuning of such data can verify the adaptability and effectiveness of the LoRA method in text generation tasks.

The XSum dataset contains 204,045 news articles, and the data is divided into a training set (204,045 articles), a validation set (11,332 articles), and a test set (11,334 articles), ensuring the stability and generalization ability of the experiment. During the experiment, only the subset of data required for fine-tuning was trained to simulate low-resource scenarios. For example, this study only uses 1% to 10% of the training data to evaluate the performance of LoRA under data-constrained conditions. At the same time, since the summaries of XSum usually contain highly compressed information, the difference in the quality of the summaries between the fine-tuning methods can be clearly measured by comparing the output results of the full parameter fine-tuning and LoRA fine-tuning models.

In terms of data preprocessing, all texts are standardized, including the removal of special characters, punctuation normalization, and filtering of low-frequency words. In addition, in order to ensure the rationality of low-resource scenarios, the experiment adopts the method of Few-shot Learning to train the low-rank matrix adapted by LoRA under different data amounts to observe its impact on model performance. The experimental evaluation mainly uses ROUGE (Recall-Oriented Understudy for Gisting Evaluation) as the automatic evaluation indicator, among which ROUGE-1, ROUGE-2 and ROUGE-L are used to measure the effectiveness of the LoRA method. This dataset and experimental design can fully verify the feasibility of LoRA in low-resource environments and provide an experimental basis for further optimization of low-rank adaptation methods in the future.

ISSN:2377-0430

Vol. 5, No. 6, 2025

First, the fine-tuning effect evaluation of LoRA under different data scales is given, and the experimental results are shown in Table 1.

Table 1: Experimental results

Data ratio (%)	Method	ROUGE-1	ROUGE-2	ROUGE-L
1%	Full parameter fine-tuning	28.5	8.2	25.3
1%	LoRA fine-tuning	27.8	7.9	24.8
5%	Full parameter fine-tuning	35.2	12.1	30.5
5%	LoRA fine-tuning	34.6	11.7	29.9
10%	Full parameter fine-tuning	40.3	15.5	35.7
10%	LoRA fine-tuning	39.8	15.0	35.2
100%	Full parameter fine-tuning	45.9	18.8	41.4
100%	LoRA fine-tuning	45.1	18.3	40.9

The experimental results show that the performance gap between the LoRA fine-tuning method and the full parameter fine-tuning method is small under different data ratios, especially at 1% and 5% data scale, LoRA can still achieve ROUGE scores close to those of full parameter fine-tuning. For example, at 1% data scale, the ROUGE-1/ROUGE-2/ROUGE-L of full parameter fine-tuning are 28.5/8.2/25.3 respectively, while the scores of LoRA fine-tuning are 27.8/7.9/24.8, with only a slight decrease, indicating that LoRA can still effectively adapt the model even in extremely low resource conditions. In addition, at 5% data scale, LoRA still maintains similar summary quality, and the ROUGE index decreases by no more than 0.6 compared with the full parameter fine-tuning method, further verifying the adaptability of LoRA in low data volume environments.

When the data scale increases to 10% and above, the fine-tuning performance of LoRA remains stable, but the gap in ROUGE indicators gradually widens compared to full parameter fine-tuning. For example, at 100% data scale, the ROUGE-1 score of full parameter fine-tuning reaches 45.9, while LoRA is only 45.1, and ROUGE-2 and ROUGE-L also show a similar downward trend. This shows that under sufficient data conditions, full parameter fine-tuning can more fully adjust the model weights to achieve better performance. Despite this, the results of LoRA are still close to full parameter fine-tuning, indicating that it can provide a lightweight optimization solution on large-scale datasets while avoiding the problem of computing resource consumption caused by full parameter fine-tuning.

Overall, LoRA has a significant fine-tuning advantage in low-resource scenarios, especially when the amount of data is small, its fine-tuning effect is almost equivalent to the full parameter method, and it can greatly reduce the training and storage costs. However, when the amount of data is large, full parameter fine-tuning can still bring higher text summary quality. Therefore, in practical applications, the appropriate fine-tuning

ISSN:2377-0430

Vol. 5, No. 6, 2025

strategy can be selected according to the data scale and computing resources: when data and computing resources are limited, LoRA is an efficient and feasible choice, while when resources are sufficient, full parameter fine-tuning can bring better performance improvement.

Secondly, this paper explores the impact of different r values on model performance, and the experimental results are shown in Table 2.

Table 2: The impact of different r values on model performance

r	**Data ratio (%)**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
4	10%	38.2	14.3	33.5
8	10%	39.1	14.8	34.2
16	10%	39.8	15.0	35.2
32	10%	40.0	15.2	35.4
4	100%	43.5	17.5	39.1
8	100%	44.2	18.0	39.8
16	100%	45.1	18.3	40.9
32	100%	45.3	18.4	41.1

The experimental results show that the rank value r of LoRA's low-rank matrix has a certain impact on the model performance, especially under a smaller data scale (10% data ratio), the ROUGE index of different r values changes significantly. When r increases from 4 to 32, the ROUGE-1 index increases from 38.2 to 40.0, ROUGE-2 increases from 14.3 to 15.2, and ROUGE-L increases from 33.5 to 35.4, indicating that appropriately increasing the r value helps the model capture richer features and improve the quality of the summary. However, as the r value increases, the performance improvement gradually decreases, which means that there is an optimal range for LoRA's low-rank approximation. After exceeding a certain r value, the additional computational overhead does not bring significant performance improvement.

On the full data set (100% data ratio), the overall ROUGE index is significantly improved compared with the 10% data scale, and as r increases, the ROUGE index still increases. For example, when r increases from 4 to 32, ROUGE-1 increases from 43.5 to 45.3, but the increase (1.8) is more moderate than the increase (1.8) when the data is 10%, indicating that with sufficient data, the low-rank structure of LoRA can effectively capture key information, and further increasing r can only bring marginal benefits. Therefore, in practical applications, the appropriate r value should be selected according to the data scale and computing resources. For example, r=16 may be a better compromise point, which can maintain a high summary quality without increasing too much computing burden.

5. Conclusion and Future Work

This study proposes a Mask2Former semantic segmentation algThis study explores the efficient fine-tuning method of the LoRA algorithm in a low-resource environment, and verifies its adaptability under different data scales and low-rank matrix settings through experiments. The experimental results show that LoRA can still maintain performance close to full parameter fine-tuning under extremely low data volume and computing resource constraints, especially when the data ratio is 10% or less, its ROUGE index only

ISSN:2377-0430

Vol. 5, No. 6, 2025

decreases slightly, indicating that this method has good application value in low-resource scenarios. At the same time, LoRA significantly reduces the computational overhead by introducing low-rank matrices to optimize weight updates, providing an effective solution for lightweight adaptation of large-scale pre-trained models.When analyzing the rank value r of the low-rank matrix, the experimental results show that appropriately increasing r can improve model performance, but when r exceeds a certain threshold, the performance gain gradually stabilizes. This shows that LoRA achieves efficient feature learning through low-rank approximation, but too large r values may lead to increased computational overhead, and the performance improvement is limited. Therefore, in practical applications, the appropriate r value should be selected according to the computing resources and data scale of the specific task to strike a balance between computational efficiency and model performance.In general, LoRA, as a parameter-efficient fine-tuning method, has shown strong adaptability and generalization capabilities in low-resource environments. Compared with full parameter fine-tuning, it can reduce computing resource consumption while ensuring model performance, making large-scale pre-trained models more widely used in computing-constrained scenarios. Future research can further explore LoRA's optimization strategies for different tasks and architectures, such as combining other parameter-efficient fine-tuning methods or adaptively adjusting the r value to further improve its practicality and generalization capabilities in low-resource environments.

References

[1] Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. Advances in Neural Information Processing Systems (NeurIPS).

[2] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. International Conference on Learning Representations (ICLR).

[3] Zhang, R., Brown, J., & Gao, H. (2023). Efficient Low-Rank Adaptation for Neural Network Fine-Tuning in Resource-Constrained Environments. Journal of Machine Learning Research (JMLR), 24(102), 1-22.

[4] Kovaleva, O., Li, J., & Ribeiro, M. (2022). Revisiting Parameter-Efficient Fine-Tuning Methods for Large-scale Transformers. Transactions of the Association for Computational Linguistics (TACL), 10, 47-63.

[5] Goyal, S., Madani, A., & Wang, K. (2024). Towards Practical Low-Rank Adaptation in NLP: Scaling Efficiently with Minimal Compute. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[6] Park, J., Lin, T., & Singh, M. (2023). Low-Rank Adaptation with Structured Sparsity for Memory-Efficient Model Fine-Tuning. International Conference on Machine Learning (ICML), 2023.

[7] Cheng, Y. (2025). Multivariate Time Series Forecasting through Automated Feature Extraction and Transformer-Based Modeling. Journal of Computer Science and Software Applications, 5(5).

[8] Duan, Y. (2024). Continuous Control-Based Load Balancing for Distributed Systems Using TD3 Reinforcement Learning. Journal of Computer Technology and Software, 3(6).

[9] Dai, L., Zhu, W., Quan, X., Meng, R., Cai, S., & Wang, Y. (2025). Deep Probabilistic Modeling of User Behavior for Anomaly Detection via Mixture Density Networks. arXiv preprint arXiv:2505.08220.

[10] Lou, Y. (2024). Capsule Network-Based AI Model for Structured Data Mining with Adaptive Feature Representation. Transactions on Computational and Scientific Methods, 4(9).

[11] Wang, B. (2025). Topology-Aware Decision Making in Distributed Scheduling via Multi-Agent Reinforcement Learning. Transactions on Computational and Scientific Methods, 5(4).

[12] Cui, W., & Liang, A. (2025). Diffusion-Transformer Framework for Deep Mining of High-Dimensional Sparse Data. Journal of Computer Technology and Software, 4(4).

ISSN:2377-0430

Vol. 5, No. 6, 2025

[13] Sheng, Y. (2024). Temporal Dependency Modeling in Loan Default Prediction with Hybrid LSTM-GRU Architecture. Transactions on Computational and Scientific Methods, 4(8).

[14] Liu, J. (2025). Reinforcement Learning-Controlled Subspace Ensemble Sampling for Complex Data Structures.

[15] Ren, Y., Wei, M., Xin, H., Yang, T., & Qi, Y. (2025). Distributed Network Traffic Scheduling via Trust-Constrained Policy Learning Mechanisms. Transactions on Computational and Scientific Methods, 5(4).

[16] Wang, Y., Tang, T., Fang, Z., Deng, Y., & Duan, Y. (2025). Intelligent Task Scheduling for Microservices via A3C-Based Reinforcement Learning. arXiv preprint arXiv:2505.00299.

[17] Xing, Y., Wang, Y., & Zhu, L. (2025). Sequential Recommendation via Time-Aware and Multi-Channel Convolutional User Modeling. Transactions on Computational and Scientific Methods, 5(5).

[18] Wang, Y., Fang, Z., Deng, Y., Zhu, L., Duan, Y., & Peng, Y. (2025). Revisiting LoRA: A Smarter Low-Rank Approach for Efficient Model Adaptation. arXiv preprint arXiv: not available.

[19] Lou, Y., Liu, J., Sheng, Y., Wang, J., Zhang, Y., & Ren, Y. (2025). Addressing Class Imbalance with Probabilistic Graphical Models and Variational Inference. arXiv preprint arXiv:2504.05758.

[20] Wang, J. (2025). Markov network classification for imbalanced data with adaptive weighting. Journal of Computer Science and Software Applications, 5(1), 43-52.

[21] Liu, J. (2025). Multimodal Data-Driven Factor Models for Stock Market Forecasting. Journal of Computer Technology and Software, 4(2).

[22] Deng, Y. (2025). A hybrid network congestion prediction method integrating association rules and LSTM for enhanced spatiotemporal forecasting. Transactions on Computational and Scientific Methods, 5(2).

[23] Liu, J., Gu, X., Feng, H., Yang, Z., Bao, Q., & Xu, Z. (2025). Market Turbulence Prediction and Risk Control with Improved A3C Reinforcement Learning.

[24] Chen, X., Patel, R., & Johnson, D. (2022). Low-Rank Model Adaptation for Multilingual and Multitask NLP. Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).

[25] Wu, Z., Fernandez, J., & Yang, B. (2023). Exploring the Impact of Low-Rank Approximation on Model Performance in Resource-Constrained Environments. Neural Networks, 169, 72-88.