Empirical Analysis of Parameter-Efficient Fine-Tuning Strategies for Domain-Specific Time-Series Anomaly Detection

Seoyeon Kim
Department of Artificial Intelligence Software
Hanbat National University
Daejeon, Republic of Korea
seoyeon@edu.hanbat.ac.kr

Yunho Jeon†
Department of Artificial Intelligence Software
Hanbat National University
Daejeon, Republic of Korea
yhjeon@hanbat.ac.kr

***Abstract*—Recent studies have confirmed the potential of large language models (LLMs) to demonstrate excellent performance in time-series anomaly detection. However, achieving such performance requires substantial computational costs and memory resources, rendering practical deployment difficult in environments with limited resources. To address this issue, this paper investigates the structural adaptation of small-scale LLMs using LoRA, focusing on how structural choices determine performance and efficiency. A comprehensive experimental investigation was conducted, encompassing three structural dimensions: module selection (All, Attention, MLP, and single-MLP modules), fine-tuning layer scopes (Upper, Middle, Lower), and rank (1, 4, 8). Using the Llama 3.2-1B model for anomaly detection on an ECG dataset, We observe that module selection and layer-scope selection are central factors that significantly impact performance. The MLP modules exhibit consistent and robust performance across parameter budgets. Notably, the Gate module, when trained alone, shows high efficiency beyond a certain parameter threshold. While rank affects overall performance, the relative ordering among modules remains largely unchanged. These findings highlight that structural LoRA design, particularly the selection of modules to train at which layers, is effective in adapting small LLMs to the task of time-series anomaly detection.**

***Index Terms*—Time-Series Anomaly Detection, Small LLMs, LoRA, Structural Fine-tuning, Domain-Specific Adaptation**

I. Introduction

Large language models (LLMs) demonstrate strong performance not only in natural language processing but also in time-series analysis tasks [1]. Ultra-large models such as the GPT-4/5 family capture complex patterns through extensive pretraining and achieve high accuracy in time-series anomaly detection and forecasting. Yet, their effectiveness relies on substantial computational and memory demands, which limit practical deployment in on-device environments. This motivates a transition toward smaller models (1B–7B parameters) for real-world applications operating under resource constraints. However, prior studies indicate that small LLMs experience notable performance degradation when limited to zero-shot inference or simple prompting techniques [2].

In this study, we posit that small LLMs can achieve competitive performance on time-series tasks by selectively updating only task-critical components rather than uniformly modifying the entire model. To address the resulting adaptation challenges, we adopt structural fine-tuning that targets these time-series-critical components. Such selective adaptation has the potential to substantially reduce the number of trainable parameters while maintaining or even improving accuracy. We embody this idea using Parameter-Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA), and systematically analyze how adaptation effects vary based on the location and content updated within the Transformer. Specifically, we analyze distinct performance and efficiency trade-offs arising from (i) module selection, (ii) layer scopes (which block ranges are adapted), and (iii) LoRA rank, demonstrating that PEFT efficiency varies substantially across these structural choices. These results indicate that, although small LLMs exhibit limited general capabilities, they can still provide sufficient expressiveness for domain-specific time-series anomaly detection when structurally optimized. The main contributions of this paper are as follows:

- We conduct a comprehensive empirical study of LoRA-based structural fine-tuning for domain-specific time-series anomaly detection, decomposing the adaptation design space into three axes: module selection, layer depth, and LoRA rank.
- We show that optimal layer ranges are module-dependent, providing a practical basis for selecting efficient module–layer adaptation strategies under tight parameter budgets.

II. Related Work

A. LLMs for Time-series Anomaly Detection

Recent studies demonstrate that LLMs can be applied to anomaly detection. Xie et al. [3] evaluated whether LLMs could generalize across diverse time-series tasks and showed that LLMs are capable of interpreting not only textual information but also altering patterns in continuous signals,

*This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00240379).

†Corresponding author

including time-series data. Concurrently, they also emphasized that LLMs are highly sensitive to input format and prompt selection, exhibiting significant instability in zero-shot scenarios. Chen et al. [4] compared zero-shot and few-shot prompting approaches for time-series anomaly detection. Their finding indicated that LLMs can partially identify time-series patterns even when only basic time-series-to-text conversion is employed. This indicates that LLMs can leverage their internal reasoning capabilities to interpret semantic changes or anomaly signals within time-series data. Zhang et al. [5] proposed a model architecture tailored to time-series processing by introducing dedicated temporal tokens, cross-modal alignment, and LoRA-based structural adaptation. It also introduced DynaLoRA, demonstrating that selectively adapting only certain model components offers substantial advantages in learning time-series patterns. These results collectively suggest the effectiveness of selective fine-tuning for adapting LLMs to time-series tasks. The collective findings of these studies underscore the importance of methodically selecting which components of an LLM to adapt for time-series tasks.

B. Low-Rank Adaptation

Hu et al. [6] proposed Low-Rank Adaptation (LoRA), which constrains weight updates to a low-rank decomposition during fine-tuning to drastically reduce trainable parameters. LoRA keeps the pre-trained weights fixed and represents the weight increment matrix as a product of low-rank matrices. This approach significantly reduces the number of trainable parameters and thereby lowers memory usage and computational cost, while still achieving performance comparable to or even exceeding that of full fine-tuning. Consequently, LoRA has become a leading PEFT method. Zhang et al. [7] pointed out that LoRA's fixed rank structure fails to reflect differences in importance between modules. They proposed a method that uses SVD-based importance analysis to automatically assign higher ranks to important modules and lower ranks to less important ones. Yao et al. [8] focused on layer selection. After training with LoRA inserted into all layers, they found that removing more than half of the LoRA modules from low-importance layers did not decrease perplexity. However, removing modules from high-importance layers led to a sharp performance drop. Based on these findings, they proposed an importance-based automatic layer selection method that introduces importance-aware sparse tuning, activating PEFT modules only on certain layers.

III. Methodology

A. Motivation

LoRA is a widely used parameter-efficient fine-tuning method that inserts low-dimensional trainable matrices in place of updating the full set of model weights. Existing studies have demonstrated that reducing LoRA's rank does not significantly impair performance and that performance varies substantially depending on which layers are adapted and which modules LoRA is applied to [6]–[8]. These findings, however, have been primarily established in natural language processing and instruction-following tasks. Time-series anomaly detection presents different data characteristics, including noise sensitivity and context-dependent semantics, which may alter the relative importance of model components. Therefore, it remains unclear whether the same structural principles observed in prior LoRA studies generalize to the time-series domain, motivating a systematic examination specific to time-series anomaly detection.

Based on these observations, this study investigates how structural LoRA configurations affect the adaptation of small LLMs to time-series anomaly detection. We focus on three key design axes.

Rank controls the representational capacity of the low-rank approximation in LoRA. It determines how much task-specific information can be encoded through the additional low-rank matrices after adaptation. A lower rank yields more compact updates but may limit expressive power, while higher ranks expand representational capacity at the cost of increased parameters.

Layer scopes define the depth at which LoRA is applied within the Transformer hierarchy. Since different layer scopes contribute differently to semantic understanding or local feature extraction, selecting where adaptation occurs plays a key role in shaping how the model incorporates time-series-specific patterns.

Module selection specifies which internal components of the Transformer—such as Attention or MLP submodules—are updated during adaptation. Each module processes fundamentally different aspects of the representation, choosing the most significant components enables more effective performance improvement under limited parameter budgets.

Through this analysis, we identify settings where structural choices enable performance improvements beyond simple parameter reduction, as well as configurations that significantly lower computational cost without compromising accuracy. Ultimately, we provide design guidelines for LoRA configurations tailored to the characteristics of time-series anomaly detection.

B. Experiment Setup

This study created a consistent experimental environment for evaluating LoRA-based structural fine-tuning for ECG anomaly detection. This section details the components used, including the dataset, model, preprocessing method, fixed training settings, and evaluation metrics.

Dataset. The experiment used an ECG time-series dataset compiled from the MIT-BIH Arrhythmia Database [9]. Each sample consists of a single lead signal of length 187. The original four abnormal classes were consolidated into two: normal(0) and abnormal(1). This reconfigured the data into a binary classification format.

Model. This study used Llama 3.2-1B as the base model. The overall model architecture is presented in Fig. 1. All experiments were conducted by varying only the LoRA configuration while sharing the same initial weights.

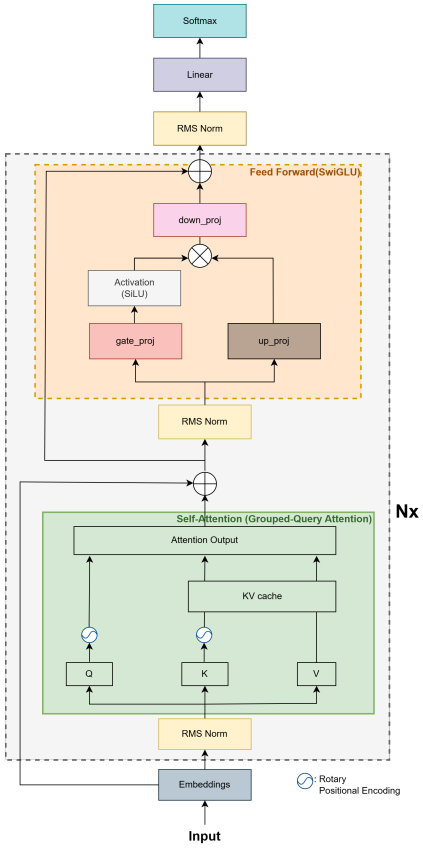

Fig. 1: Overall architecture of the Llama-3

The model is composed of a hierarchical structure in which the same Transformer block is repeated N times. First, it undergoes a Self-Attention module. Then, it passes through a Feed-Forward Network (FFN). The Self-Attention module generates Query, Key, and Value through representations from the hidden states, computes attention between tokens, and then integrates the weighted information to produce the output features. The subsequent FFN is a nonlinear structure based on SwiGLU and composed of three projection layers: {up_proj, gate_proj, down_proj}.

Tokenization. Time-series values were scaled by 1,000, rounded to the nearest integer, and converted into space-separated sequences. These sequences were then tokenized using the default Llama tokenizer.

Train Setting. To ensure consistent comparison across all experiments, we fixed the training environment as follows: the model was trained for 3 epochs with a batch size of 4 for both training and evaluation. The LoRA scaling factor (alpha) was set to twice the rank, and a dropout rate of 0.1 was applied to LoRA modules. During the experiment, the only variables altered were rank, layer intervals, and module combinations. All other settings remained fixed. Additionally, layers to which LoRA was not applied were frozen to prevent parameter updates during training.

Evaluation Metric. The model's anomaly detection performance was evaluated using Average Precision (AP) score as the primary metric.

IV. Experiment

A. Search Space

This study designed an experimental setup centered on three axes—rank, fine-tuning layer scopes, and module selection—to analyze the impact of LoRA's structural design on its adaptation performance in the ECG domain.

Rank. The decomposition dimension r of LoRA was set to {1, 4, 8}, and performance changes were compared under identical conditions for each rank.

Layer. The Llama 3.2-1B model consists of 16 Transformer blocks. In this experiment, we selectively train half of these, 8 blocks. We divided them into three segments: the top 8(0-7), middle 8(4-11), and bottom 8(8-15) for comparison.

Module. The applied modules are categorized into two domains: Self-Attention (q/k/v/o) and FFN (up/gate/down). Experimental combinations were configured by applying either the entire module (All), Attention-only, MLP-only, or a single internal module (up/gate/down) within the MLP.

B. Module/Layer-based Structural Fine-Tuning

This experiment compared the impact on performance of the LoRA-attached module and the training-included layer section. As illustrated by Fig. 2, the performance variation is visualized according to the module-layer combination under the same rank=8 condition. The parameter on the x-axis refers to the number of trainable parameters used during the LoRA training process.

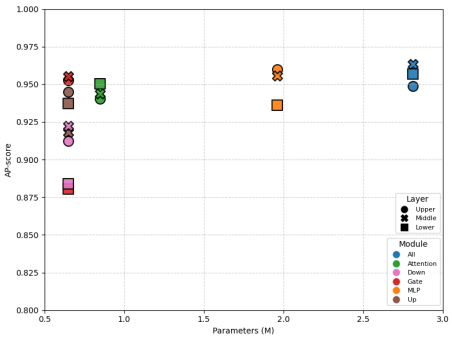

Fig. 2: Impact of LoRA layer scope selection on AP-score

The experimental results clearly showed that each module operates most effectively at distinctly different fine-tuning layer scopes. The All, MLP, Gate, and Down modules achieved their highest performance in the Middle layer. In contrast, the Attention module performed best in the Lower layer, while the Up module demonstrated its optimal performance in the Upper layer. Additionally, despite using fewer trainable parameters than the MLP module, the Gate module achieved an AP score nearly identical to that of the highest-performing MLP module. The All module exhibits strong performance, but its parameter efficiency is lower compared to the other modules.

Table I shows the optimal layer ranges for each LoRA module, determined from the visual comparison results described earlier. Based on this table, the optimal layer position

TABLE I: LoRA module selection results (rank 8).

Module	Layer	Detection performance				Params (M)
		Precision	Recall	F1-score	AP-score	
All	Middle	0.93	0.88	0.90	0.9634	2.81
Attention	Lower	0.83	0.90	0.87	0.9504	0.85
MLP	Upper	0.96	0.86	0.90	0.9598	1.96
Up	Upper	0.96	0.82	0.89	0.9449	0.65
Gate	Middle	0.98	0.80	0.88	0.9552	0.65
Down	Middle	0.97	0.76	0.85	0.9223	0.65

for each module was fixed in all subsequent experiments. Furthermore, within the MLP structure, the Gate module exhibited the highest AP score and was also confirmed to demonstrate outstanding parameter efficiency. Accordingly, subsequent experiments additionally performed performance analysis under various conditions, with a particular focus on the Gate module.

C. Impact of LoRA Rank

Experiments conducted on All, Attention, MLP, and single-MLP (Gate) configurations across rank = {1, 4, 8} revealed clear differences in the sensitivity of each configuration to rank variation. As shown in Fig. 3, we examine the correlation between alterations in rank and the extent of trainable parameters subsequent to the stabilization of module configurations. The layer intervals for each module were set to the optimal ranges empirically determined through previous experiments.

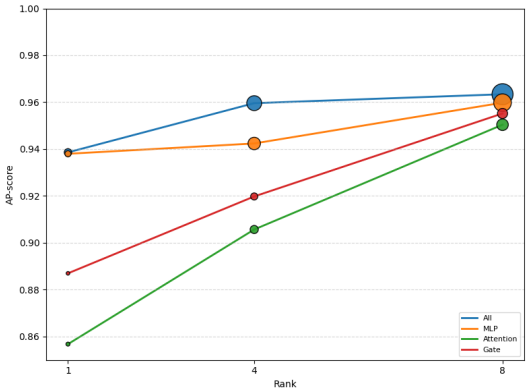

Fig. 3: Effect of LoRA rank scaling on AP-score.

While the All and single-MLP modules showed relatively gradual performance degradation, the Attention and Gate modules exhibited steep performance declines as rank decreased. Moreover, despite the alterations in rank, the module performance rankings observed in the preceding experiments demonstrated stability.

To further evaluate how rank affects parameter efficiency, we analyze the relationship between the number of trainable parameters and AP performance. As shown in Fig. 4, a graph is presented that plots trainable parameters on the x-axis for the same experimental data. As the number of trainable parameters increases during training, a general trend of increasing AP scores emerges for each module. This confirms that the expansion of expressive power as LoRA rank increases leads to actual improvements in anomaly detection performance. In the

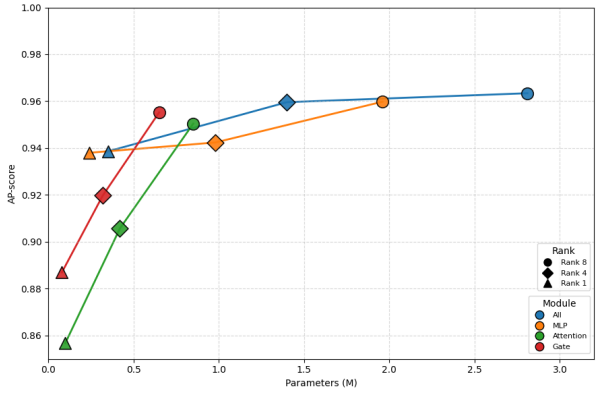

Fig. 4: Parameters vs AP-score by Module and Rank

range below 0.5M parameters, the MLP module demonstrates the highest efficiency. The model achieves an AP of 0.93–0.94 even with relatively few parameters (0.2–0.5M), outperforming Gate and Attention within the same parameter range.

Conversely, in regions where parameters exceed 0.5M, the Gate module's performance surges dramatically, transforming it into the module with the highest performance efficiency relative to parameters. While Gate exhibits unstable performance in low-parameter ranges, its AP-score recovers rapidly once sufficient trainable parameters are secured, approaching the performance levels of MLP or even demonstrating higher efficiency in certain segments.

TABLE II: AP-score and parameters by LoRA rank.

Module	Layer	Rank	AP-score	Params (M)
All	Middle	1	0.9386	0.35
		4	0.9596	1.40
		8	0.9634	2.81
Attention	Lower	1	0.8568	0.10
		4	0.9057	0.42
		8	0.9504	0.85
MLP	Upper	1	0.9380	0.24
		4	0.9424	0.98
		8	0.9598	1.96
Gate	Middle	1	0.8870	0.08
		4	0.9198	0.32
		8	0.9552	0.65

Table II reports the AP-score and trainable parameters for each module across rank = 1, 4, 8, with layer scopes fixed to the previously selected optima. It serves as a concise reference for the rank–parameter trade-off underlying Figs. 3–4.

D. Discussion

This study analyzed the impact of modifications in the application location and rank changes of LoRA across modules and layers on the AP-score from multiple perspectives. Firstly, module selection has a more substantial influence on performance than rank or parameter budget. While MLP and Gate consistently achieved superior performance or efficiency across all intervals, Attention and Down exhibited considerably lower effectiveness despite having identical parameter and

rank conditions. Secondly, each module exhibited variability across selected layer scopes, showing a tendency to perform better in specific intervals. MLP, Gate, and Down achieved optimal performance in middle layers, Attention in lower layers, and Up in upper layers. Thirdly, rank reduction caused performance degradation across all modules, but MLP and All remained relatively robust even at low ranks, while Attention and Gate were sensitive to variations in rank.

The MLP module confirmed the trend previously observed in other studies of its superior performance [4]. In contrast, when applying a single-MLP module, we observed strong performance specifically when only the Gate module was trained, which differs from previous studies [10] that incorporated additional MLP components. The superior performance of the Gate is expected to be partially related to the characteristics of the time-series anomaly detection data. Time-series data is often subject to the presence of noise and outliers, and a single value may bear different meanings depending on its placement in a sequence or the context in which it is presented. The Gate's structural design, which modulates the flow of information, may have functioned to either mitigate or accentuate the irregular patterns that are pervasive throughout the time-series.

V. Conclusion and Future Work

This study systematically investigated performance variations in LLM-based time-series anomaly detection by comparing different LoRA module configurations, layer scopes, and rank settings. The results show that performance varies substantially depending on the selected module, even under the same parameter budget. In particular, MLP and Gate demonstrated the highest stability and parameter efficiency across experiments. While the rank settings did affect absolute accuracy, the relative effectiveness of each module remained largely consistent. These findings suggest that the selection of modules and layers constitutes a more significant influence on the structural design than does rank adjustment when adapting LLMs to time-series anomaly detection.

Through experimental analysis, we confirmed the impact of module, layer, and rank on time-series anomaly detection performance. However, a sufficiently established theoretical basis to clearly support these observations has yet to be developed. Specifically, while the consistent superior performance of MLP and Gate modules was explained through empirical interpretations related to the characteristics of time-series data, the precise manner in which structural elements contribute requires deeper analysis in subsequent research. Furthermore, since experiments were conducted focusing on a single time-series domain, it is necessary to validate whether the observed trends hold consistently across diverse time-series environments through experiments with additional datasets. This extension will be a crucial future research task to clarify whether the findings identified in this study represent general characteristics or phenomena specific to particular data types.

References

[1] J. Jin, L. Zhang, Z. Zhao, J. Li, and S. Liu, "Time-LLM: Time series forecasting by reprogramming large language models," in Proc. Int. Conf. Learn. Represent. (ICLR), Vienna, Austria, May 2024.

[2] T. Zhang, J. Yuan, and S. Avestimehr, "Revisiting OPRO: The limitations of small-scale LLMs as optimizers," in Proc. Findings of the Assoc. for Comput. Linguistics: ACL, Bangkok, Thailand, Aug. 2024, pp. 1727–1735.

[3] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One Fits All: Power general time series analysis by pretrained LM," in Proc. Conf. Neural Inf. Process. Syst. (NeurIPS), New Orleans, LA, USA, Dec. 2023.

[4] M. Dong, H. Huang, and L. Cao, "Can LLMs serve as time series anomaly detectors?" in Findings Assoc. Comput. Linguistics: EMNLP, Miami, FL, USA, Nov. 2024, pp. 1–27.

[5] J. Zhang, J. Gao, W. Ouyang, W. Zhu, and H. Y. Leong, "Time-LlaMA: Adapting Large Language Models for Time Series Modeling via Dynamic Low-rank Adaptation," in Proc. ACL Student Research Workshop, pp. 1145–1157, Jul. 2025.

[6] E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," arXiv preprint arXiv:2106.09685, version 2, 2021.

[7] R. Zhang et al., "ADALoRA: Adaptive budget allocation for parameter-efficient fine-tuning," in Proc. Int. Conf. Learn. Represent. (ICLR), Kigali, Rwanda, May 2023.

[8] K. Yao et al., "Layer-wise importance matters: Less memory for better performance in parameter-efficient fine-tuning of large language models," in Findings Assoc. Comput. Linguistics: EMNLP, Miami, FL, USA, Nov. 2024, pp. 1977–1992.

[9] Shayan Fazeli, "Heartbeat: MIT-BIH Arrhythmia Dataset," Kaggle. Available: https://www.kaggle.com/datasets/shayanfazeli/heartbeat. [Accessed: Nov. 5, 2025].

[10] Y. Xue and B. Mirzasoleiman, "LoRA is All You Need for Safety Alignment of Reasoning LLMs," arXiv preprint arXiv:2507.17075, Jul. 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.17075