LoFTPat: Low-Rank Subspace Optimization for Parameter-Efficient Fine-Tuning of Genomic Language Models in Pathogenicity Identification

Sajib Acharjee Dip[1], **Dipanwita Mallick**[2], **Liqing Zhang**[1]*
[1]Department of Computer Science
Virginia Tech, USA
{sajibacharjeedip,lqzhang}@vt.edu
[2]Department of Computer Science
North South University, Bangladesh
dipanwitamallick@nsu.edu

Abstract

Pathogen identification from genomic sequences is essential for infectious disease surveillance, antimicrobial resistance monitoring, and vaccine development. While Large Language Models (LLMs) have demonstrated remarkable success in genomic sequence modeling, existing approaches prioritize classification accuracy over computational efficiency, resulting in high memory overhead, prolonged training times, and scalability limitations. To address this, we introduce LoFTPat, a structurally constrained fine-tuning framework that integrates Low-Rank Adaptation within the self-attention mechanisms of PathoLM, enabling low-dimensional subspace optimization for task-specific weight modulation. By decoupling adaptation from full model retraining, LoFTPat significantly reduces parameter complexity while preserving model generalization, making it a scalable and efficient alternative to full fine-tuning.

Our method achieves a 4.02% reduction in total training time, a 64.3% decrease in peak GPU memory consumption, and a 99.24% reduction in trainable parameters, while surpassing full fine-tuning approaches in classification performance. Specifically, LoFTPat outperforms PathoLM with +0.44% higher accuracy, +0.44% F1-score, +0.02% AUC-ROC, and +0.52% balanced accuracy. Unlike previous models, LoFTPat efficiently adapts to both short-read and long-read sequences, demonstrating robust generalization across bacterial and viral pathogens. By optimizing hierarchical feature transformations with minimal parameter overhead, LoFTPat presents a scalable and computationally efficient framework for large-scale pathogen classification and genomic analysis.

1 Introduction

Recent advancements in Large Language Models (LLMs) have significantly impacted various domains of biology and healthcare, enabling breakthroughs in omics data analysis, protein and genome sequence modeling Li et al. (2021); Luu & Buehler (2024); Bi et al. (2024); Liu et al. (2024); Huang et al. (2024); Lam et al. (2024); Chau et al. (2024); Mangul et al. (2024), biomedical signal processing Soumma et al. (2025b); Liu et al. (2023); Fan et al. (2025); Soumma et al. (2025a), and drug discovery. By leveraging vast amounts of pretrained knowledge, LLMs can generalize across multiple biological tasks, including protein structure prediction Jumper et al. (2021), genome annotation, and biomedical text mining. In genomics, LLMs have shown promise in analyzing long and short-read sequences, accelerating research in disease prediction Shoham & Rappoport (2024); Farahmand et al. (2024); Rafsani et al. (2025), functional genomics, and evolutionary biology Morris et al. (2024). Despite these advancements, applying LLMs to pathogen identification remains an

*Corresponding author: lqzhang@cs.vt.edu

evolving field, with challenges in scalability, efficiency, and generalization across diverse microbial species.

Pathogens, including bacteria and viruses, pose significant threats to public health, agriculture, and global ecosystems. Rapid identification of pathogenic strains is critical for disease surveillance, vaccine development, and antibiotic resistance mitigation. The recent COVID-19 pandemic underscored the necessity of early pathogen detection to control outbreaks and guide public health interventions Organization (2023). Traditional genomic analysis pipelines, while effective, often rely on alignment-based or heuristic methods, which can be computationally expensive and slow. Traditional genomic analysis pipelines, while effective, often rely on alignment-based or heuristic methods, which can be computationally expensive and slow. Other machine and deep learning methods like PaPrBaG Deneke et al. (2017), BacPaCS Barash et al. (2019), DeePac Bartoszewicz et al. (2020) and DciPatho Jiang et al. (2023) are effective but rely on manual feature extraction and incur higher computational costs, leading to slower inference. Thus, LLM-driven approaches offer a scalable solution to accurately classify and characterize pathogens from raw genomic sequences.

Several studies have explored LLMs for pathogen identification, with models such as Nucleotide Transformer v2 Dalla-Torre et al. (2024) and GenaLM Fishman et al. (2025) demonstrating promising results in genomic sequence classification. PathoLM Dip et al. (2024) extended this line of research, leveraging transformer-based architectures to classify bacterial and viral genomes. However, existing approaches primarily focus on classification accuracy, often neglecting computational efficiency—a crucial factor in real-world pathogen surveillance. Many models are resource-intensive, requiring high GPU memory and extensive training time, limiting their scalability for large-scale genomic studies.

In this work, we propose LoFTPat, a mathematically constrained, parameter-efficient fine-tuning framework designed for scalable and high-performance pathogen identification. Unlike traditional full fine-tuning approaches, which require updating all model parameters, LoFTPat leverages Low-Rank Adaptation (LoRA) Hu et al. (2021) within the self-attention layers of PathoLM, introducing a low-dimensional optimization space that significantly reduces parameter redundancy while retaining task-specific adaptation. By restructuring the weight update process into a low-rank factorized form, LoFTPat reduces trainable parameters by 99.24%, achieving near-equivalent expressivity to full fine-tuning while drastically improving computational efficiency.

Beyond parameter reduction, LoFTPat optimizes GPU utilization, accelerating convergence by reducing memory overhead by 64.3% and training time by 4.02%, making it highly practical for real-world pathogen classification pipelines. Unlike previous approaches that primarily focused on classification accuracy, our model is designed to handle both short and long-read genomic sequences, ensuring adaptability across bacterial and viral species with diverse sequence lengths. By systematically evaluating LoFTPat across multiple sequence resolutions and pathogenic families, we establish its generalization power, computational efficiency, and scalability, making it a transformative step toward real-time, resource-efficient genomic surveillance and pathogen detection.

2 METHODS

2.1 DATASET

We utilized the PathoLM dataset, obtained from its authors, which consists of genomic sequences from approximately 30 species, including both bacterial and viral pathogens. The dataset Gillespie et al. (2011) includes the seven ESKAPEE bacterial pathogens Ruekit et al. (2022): Escherichia coli, Enterococcus faecium, Staphylococcus aureus, Klebsiella pneumoniae, Acinetobacter baumannii, Pseudomonas aeruginosa, and Enterobacter spp., along with viral genomes such as Influenza, Norovirus, and Coronaviruses (SARS, MERS, OC43, NL63, 229E, HKU1). To ensure data quality and eliminate redundancy, we applied MMseqs2 Steinegger & Söding (2017) clustering to partition sequences based on similarity thresholds of 40%, 60%, and 80%. For our experiments, we used the most stringent 40% threshold, ensuring minimal sequence similarity between train and test sets to enhance model generalization. Each genomic sequence was standardized to 2,000 base pairs, providing consistency across samples.

To prevent overfitting and ensure fair evaluation, we partitioned the dataset into training (80%) and testing (20%) using MMseqs2, ensuring mutually exclusive clusters between the sets. This step

preserves biological diversity while removing homologous sequences, ensuring that the model does not memorize similar patterns from both splits. The viral dataset was curated from NCBI, selecting species with both pathogenic and non-pathogenic strains to ensure a balanced representation. We further included non-pathogenic viral genomes from wastewater metagenomics to address potential biases in health surveillance data. The resulting dataset allows LoFTPat to be evaluated on highly diverse genomic sequences, ensuring its robustness and generalization capabilities across unseen species.

2.2 PATHOLM ARCHITECTURE

PathoLM is based on the Nucleotide Transformer v2 (NT-v2) 50M Dalla-Torre et al. (2024), an encoder-only transformer model designed specifically for genomic data. The model embeds 6-mer sequences into dense vectors, utilizing Rotary Positional Embeddings (RoPE) to provide positional context up to 12,000 nucleotides. Each transformer layer applies multi-head self-attention to analyze sequence interdependencies, while residual connections and layer normalization ensure stability and efficient information retention. The feed-forward network employs Gated Linear Units (GLUs) with Swish activation, optimizing the model's computational efficiency. During pretraining, the model was trained on nucleotide-level masked language modeling (MLM), learning to predict missing nucleotides and thereby improving its ability to interpret genomic sequences.

For fine-tuning, we adapted PathoLM Dip et al. (2024) using the Hugging Face Transformers library, optimizing it for pathogen classification tasks. Sequence preprocessing involved padding shorter sequences and truncating longer ones beyond the 12kb RoPE context limit. The model was optimized for both binary pathogen classification and multi-class classification of ESKAPEE species, employing a dual-label and septenary classification strategy. Training utilized the Adam optimizer with a learning rate scheduler and a warm-up phase, preventing premature convergence while stabilizing updates. An early stopping mechanism was applied to mitigate overfitting, ensuring robust generalization across diverse pathogen genomes. These fine-tuning strategies effectively leveraged NT-v2's strong representation learning capabilities, enabling precise and efficient genomic classification.

2.3 LOFTPAT ARCHITECTURE

LoFTPat is a parameter-efficient fine-tuning (PEFT) approach that integrates LoRA (Low-Rank Adaptation) Hu et al. (2021) into PathoLM, enabling efficient adaptation of the model for pathogen classification while significantly reducing the number of trainable parameters. Unlike full fine-tuning, which updates all model weights, LoFTPat modifies only a subset of parameters by introducing low-rank decomposition into key transformer layers. This approach allows task-specific adaptation while preserving the generalization capability of the pretrained model, ensuring robust classification across known and novel pathogens.

2.3.1 MATHEMATICAL FORMULATION

For a given input sequence x, LoFTPat first tokenizes it into overlapping 6-mers, embedding them into a high-dimensional space:

$$X = \text{Embed}(x) \in \mathbb{R}^{L \times d}$$

where L is the sequence length and d is the embedding dimension. These token embeddings are then processed through PathoLM's transformer layers, which use self-attention to capture sequence dependencies.

In standard fine-tuning, the model updates all weight matrices W requiring storage and computation of full-rank parameter changes:

$$W' = W + \Delta W$$

where ΔW represents the fine-tuned weight updates. However, LoFTPat replaces ΔW with a low-rank adaptation, defined as:

$$\Delta W = BA$$

where $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$ are low-rank matrices with rank $r \ll d$. These matrices are trainable, but much smaller than W, ensuring efficient fine-tuning.

Thus, instead of applying full-rank updates, LoFTPat modifies only a compact subset of parameters, leading to a new weight representation:

$$W'x = Wx + BAx$$

This structure preserves the pretrained knowledge stored in W while allowing task-specific adaptation via ΔW.

2.3.2 LoFTPat in Self-Attention Layers

Transformers rely on query (Q), key (K), and value (V) matrices to compute self-attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where:

$$Q = W_q X, \quad K = W_k X, \quad V = W_v X$$

With LoFTPat, we apply LoRA-based weight updates to the query and key projection matrices:

$$W_q' = W_q + B_q A_q, \quad W_k' = W_k + B_k A_k$$

Substituting these into the query and key definitions:

$$Q' = (W_q + B_q A_q)X, \quad K' = (W_k + B_k A_k)X$$

The modified self-attention equation incorporating LoFTPat becomes:

$$\text{Attention}(Q', K', V) = \text{softmax}\left(\frac{(W_q + B_q A_q)X((W_k + B_k A_k)X)^T}{\sqrt{d_k}}\right)V$$

This formulation explicitly captures how LoFTPat modifies the standard self-attention mechanism, introducing additional adaptation terms while maintaining efficiency. By leveraging low-rank updates, LoFTPat ensures task-specific learning without modifying all model weights, making it an effective fine-tuning strategy.

2.3.3 Computational Benefits of LoFTPat

The parameter savings of LoFTPat can be derived as follows: - Standard fine-tuning requires $O(d^2)$ parameters per layer. - LoFTPat (with LoRA) reduces this to $O(r \cdot d)$. - The parameter reduction ratio is:

$$\frac{r(d+d)}{d^2} = \frac{2r}{d}$$

Since $r \ll d$, this results in an almost $O(1/d)$ reduction, making LoFTPat significantly more efficient.

LoFTPat preserves the pretrained weights, ensuring that the foundational knowledge remains intact and reducing the risk of catastrophic forgetting during fine-tuning. By employing low-rank adaptation, it selectively updates only the most task-relevant parameters, effectively minimizing overfitting while maintaining generalization. Additionally, the smaller parameter updates introduced by LoRA lead to smoother optimization dynamics, enhancing training stability and convergence speed. As a result, LoFTPat achieves state-of-the-art efficiency and predictive performance, making it an optimal fine-tuning strategy for large-scale genomic sequence classification.

2.4 Experimental Setup

We used two NVIDIA TITAN RTX GPUs (24GB VRAM each) with CUDA 12.6, enabling efficient fine-tuning of PathoLM with Low Rank Adaptation. The training pipeline leveraged Hugging Face Transformers with mixed-precision (FP16) to optimize memory usage, ensuring stable training across LoRA ranks (4, 8, 16, 32) while minimizing computational overhead.

Fine-tuning was performed on the same dataset as PathoLM, comprising genomic sequences from 30 pathogen species. We used the Adam optimizer (learning rate 5×10^{-5}) with 500 warmup steps, training for 40 epochs with a batch size of 4 per GPU and weight decay of 0.01 for regularization. An epoch-based evaluation strategy saved the best model based on minimum validation loss, ensuring generalization. Training logs were recorded every 100 steps, allowing efficient monitoring and optimization of LoFTPat's performance across pathogen classification tasks.

2.5 EVALUATION

LoFTPat was evaluated using accuracy, F1-score, AUC-ROC, and balanced accuracy, alongside efficiency metrics like training time, GPU memory usage, and parameter reduction. Balanced accuracy, which ensures fair evaluation across imbalanced classes, is defined as:

$$\text{Balanced Accuracy} = \frac{1}{C}\sum_{c=1}^{C}\frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}$$

where C is the number of classes, TP_c (true positives) represents correctly classified instances of class c, and FN_c (false negatives) represents misclassified instances that belong to class c but were predicted otherwise.

Efficiency gains were measured through parameter reduction percentage:

$$\text{Reduction} = 100 \times \left(1 - \frac{\text{Trainable Parameters}}{\text{Total Parameters}}\right)$$

and GPU memory efficiency, computed as:

$$\text{Peak Memory (MB)} = \frac{\max \text{Memory Allocated (bytes)}}{10^6}$$

Training time was evaluated via average epoch duration:

$$\text{Epoch Time} = \frac{\text{Total Training Time}}{\text{Number of Epochs}}$$

These metrics confirm that LoFTPat significantly reduces computational cost while maintaining strong classification performance across pathogen species.

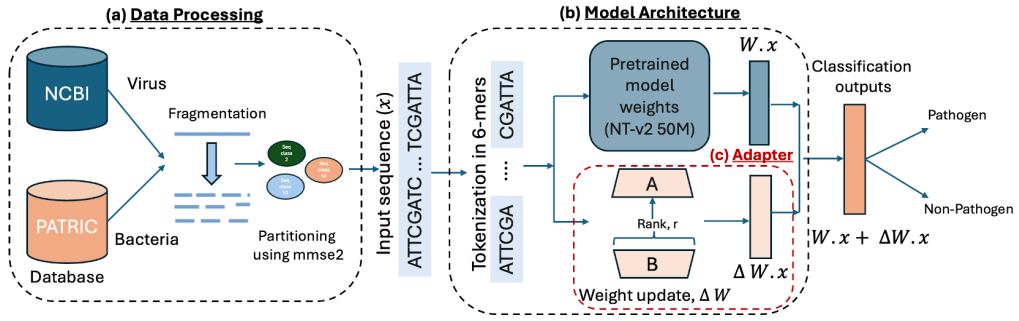

Figure 1: LoFTPat Model Architecture with LoRA Fine-Tuning. (a) Data Processing: Pathogen genomic sequences from NCBI (viruses) and PATRIC (bacteria) are fragmented and then clustered using MMseqs2 for train-test partitioning. (b) Model Architecture: Tokenized 6-mer sequences are processed by the pretrained NT-v2 50M model, with LoFTPat introducing low-rank adaptation (LoRA) for efficient fine-tuning. (c) Adapter Module: LoRA injects low-rank matrices (A, B) into weight updates, applying a rank-r transformation ($\Delta W \cdot x$). The final classification ($W \cdot x + \Delta W \cdot x$) distinguishes pathogenic from non-pathogenic sequences efficiently.

3 RESULTS AND DISCUSSIONS

3.1 EFFICIENCY GAINS WITH LORA

Parameter-efficient fine-tuning (PEFT) methods aim to balance computational efficiency and predictive performance while reducing resource overhead. Table 1 presents a comparison of LoFTPat (various ranks) against PathoLM and other baseline methods, highlighting gains in training time, memory usage, and model size. The results confirm that LoFTPat (Rank = 4) provides the best trade-off, achieving substantial efficiency improvements over full fine-tuning (PathoLM) while maintaining high classification performance.

Table 1: Performance Comparison of LoFTPat vs. baseline methods

Models	Total Training Time (sec)	Training Time / Epoch (sec)	GPU Memory Allocated (MB)	Peak GPU Used (MB)	Model Size (MB)	Param Reduction	Accuracy	F1-Score	AUC-ROC	Balanced Acc
NT-v2 (50m)	-	-	-	-	430	-	0.5783	0.4372	0.6143	0.5150
PathoLM	21652	2165	663	21321	215	0%	0.9911	0.9911	0.9995	0.9907
PathoLM + IA3	21789	2180	**236**	21754	**1.39**	99.38%	0.9945	0.9945	0.9950	0.9947
LoFTPat (Rank = 4)	**20781**	**2078**	237	**20303**	1.69	**99.24%**	**0.9955**	**0.9955**	**0.9997**	**0.9959**
LoFTPat (Rank = 8)	20820	2082	239	20310	2.28	98.97%	0.9946	0.9946	0.9997	0.9949
LoFTPat (Rank = 16)	20862	2086	243	20862	3.46	98.44%	0.9955	0.9955	0.9981	0.9959
LoFTPat (Rank = 32)	20788	2078	250	20349	5.82	97.38%	0.9953	0.9953	0.9997	0.9957

3.1.1 Training Time Reduction

LoFTPat significantly reduces training time compared to PathoLM, with LoFTPat (Rank = 4) completing training in 20,781 seconds—a 4.02% reduction compared to PathoLM (21,652 sec). This efficiency is attributed to LoRA's low-rank adaptation, which enables fine-tuning with fewer trainable parameters while maintaining effective knowledge transfer. The observed reduction in per-epoch training time follows a similar trend, with LoFTPat (Rank = 4) achieving the lowest per-epoch time (2,078 sec), compared to PathoLM (2,165 sec) and PathoLM + IA3 (2,180 sec) as shown in Table 1.

While other LoFTPat configurations (Ranks 8, 16, 32) exhibit slightly longer training times, the differences remain minimal (within 0.4% of Rank 4) and still outperform PathoLM. The results confirm that LoRA-based fine-tuning reduces computational burden without sacrificing training effectiveness, making it a superior alternative to full fine-tuning methods.

3.1.2 GPU Memory Savings

Memory efficiency is crucial for real-world deployment, especially in resource-limited settings. PathoLM requires 663 MB, while LoFTPat (Rank = 4) uses only 237 MB, achieving a 64.3% reduction with higher accuracy (0.9955 vs. 0.9911), as shown in Table 1.

PathoLM + IA3 (236 MB) offers similar memory savings but lags in accuracy (0.9945 vs. 0.9955), highlighting LoRA's superior adaptation. The consistent memory usage across LoFTPat ranks confirms LoRA's efficiency, ensuring scalability across different hardware constraints.

3.1.3 Model Size Compression

Storage efficiency is another critical factor when deploying fine-tuned models. PathoLM results in a saved model size of 215 MB, while LoFTPat (Rank = 4) achieves an extreme compression, reducing the model size to just 1.69 MB—a 99.24% reduction shown in Table 1. This drastic compression enables LoFTPat-based models to be stored, transferred, and deployed with minimal hardware constraints, making it particularly advantageous for edge computing and low-resource applications.

While PathoLM + IA3 achieves a slightly better compression rate (1.39 MB, 99.38% reduction), it does so at the cost of slightly lower predictive performance. The trend across different LoFTPat ranks shows that higher ranks (e.g., Rank = 32) increase model size (5.82 MB) but do not improve accuracy, reaffirming that Rank = 4 strikes the best balance between compression and effectiveness.

3.2 Accuracy and Robustness of LoFTPat

Despite using 99.24% fewer parameters, LoFTPat (Rank = 4) outperforms PathoLM across all predictive metrics (Table 1). Compared to PathoLM (accuracy = 0.9911, F1-score = 0.9911, AUC-ROC = 0.9995, balanced accuracy = 0.9907), LoFTPat achieves +0.44% accuracy, +0.44% F1-score, +0.02% AUC-ROC, and +0.52% balanced accuracy. While small, these improvements indicate better generalization and robustness, particularly in class-imbalanced settings.

LoRA's low-rank adaptation matrices allow task-specific fine-tuning without disrupting pre-trained knowledge, preventing overfitting while enhancing decision boundary separation (higher AUC-ROC). IA3, despite achieving similar memory efficiency, lags in accuracy, confirming that LoRA captures task-specific variations more effectively. Additionally, LoFTPat avoids unnecessary parameter drift seen in full fine-tuning, leading to better generalization.

Overall, LoFTPat (Rank = 4) surpasses both PathoLM and IA3, proving that parameter-efficient fine-tuning can achieve superior predictive performance while maintaining efficiency.

3.3 TRADE-OFF BETWEEN EFFICIENCY VS PREDICTION PERFORMANCE

Achieving an optimal balance between computational efficiency and predictive performance is crucial in fine-tuning large language models. Figure 2a presents the relationship between accuracy and training time per epoch, while Figure 2b illustrates accuracy versus GPU memory allocation across different parameter-efficient fine-tuning (PEFT) methods. These comparisons highlight the effectiveness of LoFTPat (Rank = 4) in achieving high predictive performance while maintaining competitive efficiency.

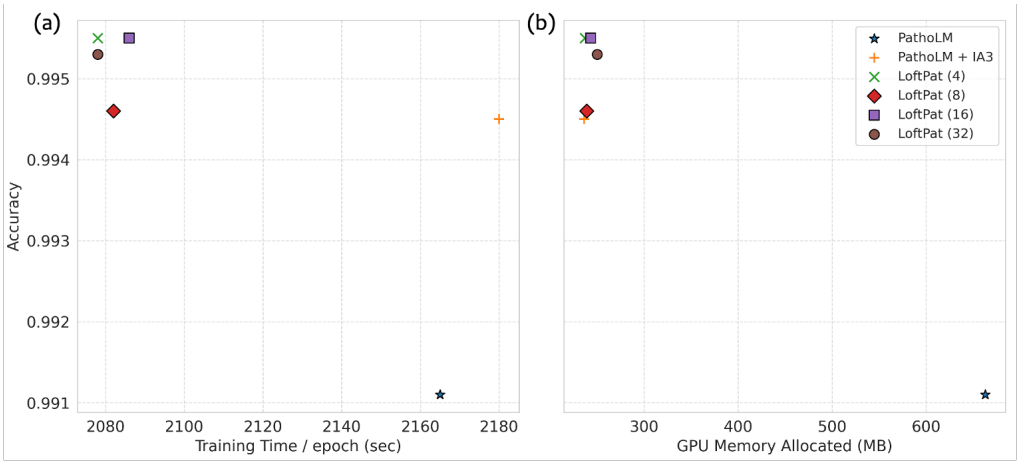

Figure 2: Efficiency-Performance Trade-off in LoFTPat. (a) Accuracy vs. Training Time: LoFTPat (Rank 4) achieves the highest accuracy with reduced training time. (b) Accuracy vs. GPU Memory: LoFTPat significantly lowers memory usage (64.3%) while maintaining strong performance.

From Figure 2a, LoFTPat (Rank = 4) achieves the highest accuracy (0.9955) with one of the lowest training times per epoch (2078 sec), outperforming PathoLM, which requires 2165 sec but achieves lower accuracy (0.9911), making it less efficient. PathoLM + IA3 shows moderate training time (2180 sec) and slightly better accuracy (0.9945) than PathoLM but still falls short of LoFTPat (Rank = 4), confirming LoRA's superior adaptation over IA3. Figure 2b highlights LoFTPat's efficiency with just 237 MB GPU memory, a 64% reduction compared to PathoLM (663 MB, 0.9911 accuracy). While PathoLM + IA3 achieves similar memory savings (236 MB), its accuracy (0.9945) remains lower, indicating LoRA's advantage in expressive adaptation without additional overhead.

Although both IA3 and LoRA optimize efficiency, IA3 reweights activations without modifying weights, limiting expressiveness, whereas LoRA learns a low-rank adaptation, enabling better task-specific tuning. LoFTPat (Rank = 4) thus outperforms IA3 in accuracy while drastically reducing GPU usage, establishing itself as the optimal fine-tuning approach for computationally efficient pathogen classification.

3.4 ABLATION STUDY ON LORA PARAMETERS

To assess the impact of the LoRA scaling factor (α) on model efficiency and performance, we conducted an ablation study with $\alpha = 8, 16, 32, 64$, keeping all other hyperparameters fixed. As shown in Table 2, the total training time and per-epoch time remain stable, with a slight increase at $\alpha = 64$, while GPU memory usage stays constant at 237 MB, indicating LoRA primarily affects weight adaptation rather than memory consumption.

Table 2: Ablation Study on LoRA α Parameter (Rank = 4)

α	Total Training Time (sec)	Training Time / Epoch (sec)	GPU Memory Allocated (MB)	Peak GPU Memory (MB)	Param Reduction	Accuracy	F1-Score	AUC-ROC	Balanced Accuracy
8	20807	2080	237	20303	99.24%	0.9950	0.9950	0.9997	0.9953
16	**20781**	**2078**	237	20303	99.24%	**0.9955**	**0.9955**	**0.9997**	**0.9959**
32	20812	2079	237	20303	99.24%	0.9953	0.9953	0.9997	0.9958
64	20828	2082	237	20303	99.24%	0.9952	0.9952	0.9997	0.9957

For classification performance, $\alpha = 16$ achieves the highest Accuracy (0.9955), F1-Score (0.9955), and Balanced Accuracy (0.9959), suggesting it strikes an optimal balance between adaptation strength and regularization. Lower values like $\alpha = 8$ show slight underfitting, while higher values ($\alpha = 32, 64$) yield diminishing returns, with minor degradations in Balanced Accuracy. These findings emphasize that $\alpha = 16$ provides the best trade-off between performance and computational efficiency, making it a robust choice for resource-constrained fine-tuning.

3.5 Performance on varied lengths of sequence

Table 3: Effectiveness in Varied Sequence Lengths

Seq Length (base pair)	Model	Total Training Time (sec)	Training Time / Epoch (sec)	GPU Memory (MB)	Saved Model Size (MB)	Param Reduction	F1-Score	Balanced Acc
Short (150)	PathoLM	132456	13245	669	21325	0%	0.9734	0.9810
	LoFTPat	128749	12874	237	20303	99.24%	0.9864	0.9867
	Δ% LoFTPat	**-2.8%**	**-2.8%**	**-64.6%**	**-4.8%**	**+99.24%**	**+1.34%**	**+0.58%**
Medium (2k)	PathoLM	21652	2165	663	21321	0%	0.9911	0.9907
	LoFTPat	20781	2078	237	20303	99.24%	0.9955	0.9959
	Δ% LoFTPat	**-4.0%**	**-4.0%**	**-64.2%**	**-4.8%**	**+99.24%**	**+0.44%**	**+0.52%**
Long (50k)	PathoLM	5786	579	671	21329	0%	0.9932	0.9912
	LoFTPat	5539	553	245	20312	99.24%	0.9932	0.9935
	Δ% LoFTPat	**-4.3%**	**-4.5%**	**-63.5%**	**-4.7%**	**+99.24%**	**+0.00%**	**+0.23%**

Fine-tuning across different sequence lengths is essential for real-world applications. Table 3 shows that LoFTPat (Rank = 4) consistently improves efficiency and accuracy across short (150bp), medium (2kbp), and long (50kbp) sequences, outperforming PathoLM in both computation and prediction.

LoFTPat reduces training time by 2.8%–4.5% and GPU memory usage by 63%, with the largest efficiency gains for long sequences (4.3% total training time, 4.5% per-epoch speedup). It also enhances F1-score (+1.34%) and Balanced Accuracy (+0.58%) for short sequences while maintaining stable performance for long reads. These gains stem from LoRA's efficient fine-tuning, which captures both short- and long-range dependencies while minimizing parameter drift. The significant memory savings reinforce LoFTPat's scalability as a robust solution for diverse genomic sequences.

4 Discussions

LoFTPat (Rank = 4) effectively balances efficiency and predictive performance, outperforming full fine-tuning methods like PathoLM while significantly reducing training time (4.02%), GPU memory usage (64%), and model size (99.24%). Compared to PathoLM + IA3, which achieves similar memory efficiency, LoFTPat maintains superior accuracy, F1-score, and balanced accuracy, demonstrating LoRA's advantage over activation reweighting methods like IA3.

Beyond efficiency, LoFTPat generalizes across sequence lengths, reducing training time consistently for short (150bp), medium (2kbp), and long (50kbp) sequences, with the largest efficiency gains for longer inputs. It reduces per-epoch training time by 4.5% for 50kbp sequences, maintaining accuracy parity with PathoLM while achieving higher balanced accuracy (+0.58% for short, +0.23% for long sequences). These improvements highlight LoRA's structured weight adaptation, which controls parameter updates while preserving pre-trained knowledge, avoiding the computational overhead of full fine-tuning. LoFTPat emerges as a scalable, resource-efficient alternative that maintains strong classification performance across diverse genomic inputs.

5 Future Works

As future work, we aim to expand the training dataset beyond 30 species to include fungi and other microbial taxa, enhancing adaptability across diverse pathogens while maintaining efficiency. A broader microbial taxonomy will improve generalization across pathogenic families. We will also evaluate robustness on unseen species to assess adaptability to novel sequences. Additionally, integrating sequence-level interpretability will refine prediction insights, ensuring practical application in real-world pathogen detection and classification.

REFERENCES

Eran Barash, Neta Sal-Man, Sivan Sabato, and Michal Ziv-Ukelson. Bacpacs—bacterial pathogenicity classification via sparse-svm. *Bioinformatics*, 35(12):2001–2008, 2019.

Jakub M Bartoszewicz, Anja Seidel, Robert Rentzsch, and Bernhard Y Renard. Deepac: predicting pathogenic potential of novel dna with reverse-complement neural networks. *Bioinformatics*, 36 (1):81–89, 2020.

Zhenyu Bi, Sajib Acharjee Dip, Daniel Hajialigol, Sindhura Kommu, Hanwen Liu, Meng Lu, and Xuan Wang. Ai for biomedicine in the era of large language models. *arXiv preprint arXiv:2403.15673*, 2024.

Tran N Chau, Xuan Wang, John M McDowell, and Song Li. Advancing plant single-cell genomics with foundation models. *Current Opinion in Plant Biology*, 82:102666, 2024.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, pp. 1–11, 2024.

Carlus Deneke, Robert Rentzsch, and Bernhard Y Renard. Paprbag: A machine learning approach for the detection of novel pathogens from ngs data. *Scientific reports*, 7(1):39194, 2017.

Sajib Acharjee Dip, Uddip Acharjee Shuvo, Tran Chau, Haoqiu Song, Petra Choi, Xuan Wang, and Liqing Zhang. Patholm: Identifying pathogenicity from the dna sequence through the genome foundation model. *arXiv preprint arXiv:2406.13133*, 2024.

Yongqi Fan, Kui Xue, Zelin Li, Xiaofan Zhang, and Tong Ruan. An llm-based framework for biomedical terminology normalization in social media via multi-agent collaboration. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10712–10726, 2025.

Ebrahim Farahmand, Shovito Barua Soumma, Nooshin Taheri Chatrudi, and Hassan Ghasemzadeh. Hybrid attention model using feature decomposition and knowledge distillation for glucose forecasting. *arXiv preprint arXiv:2411.10703*, 2024.

Veniamin Fishman, Yuri Kuratov, Aleksei Shmelev, Maxim Petrov, Dmitry Penzar, Denis Shepelin, Nikolay Chekanov, Olga Kardymon, and Mikhail Burtsev. Gena-lm: a family of open-source foundational dna language models for long sequences. *Nucleic Acids Research*, 53(2):gkae1310, 2025.

Joseph J Gillespie, Alice R Wattam, Stephen A Cammer, Joseph L Gabbard, Maulik P Shukla, Oral Dalay, Timothy Driscoll, Deborah Hix, Shrinivasrao P Mane, Chunhong Mao, et al. Patric: the comprehensive bacterial bioinformatics resource with a focus on human pathogenic species. *Infection and immunity*, 79(11):4286–4298, 2011.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Kaixuan Huang, Yuanhao Qu, Henry Cousins, William A Johnson, Di Yin, Mihir Shah, Denny Zhou, Russ Altman, Mengdi Wang, and Le Cong. Crispr-gpt: An llm agent for automated design of gene-editing experiments. *arXiv preprint arXiv:2404.18021*, 2024.

Gaofei Jiang, Jiaxuan Zhang, Yaozhong Zhang, Xinrun Yang, Tingting Li, Ningqi Wang, Xingjian Chen, Fang-Jie Zhao, Zhong Wei, Yangchun Xu, et al. Dcipatho: deep cross-fusion networks for genome scale identification of pathogens. *Briefings in Bioinformatics*, 24(4):bbad194, 2023.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Hilbert Yuen In Lam, Xing Er Ong, and Marek Mutwil. Large language models in plant biology. *Trends in Plant Science*, 2024.

Hong-Liang Li, Yi-He Pang, and Bin Liu. Bioseq-blm: a platform for analyzing dna, rna and protein sequences based on biological language models. *Nucleic acids research*, 49(22):e129–e129, 2021.

Chunyu Liu, Yongpei Ma, Kavitha Kothur, Armin Nikpour, and Omid Kavehei. Biosignal copilot: Leveraging the power of llms in drafting reports for biomedical signals. *medRxiv*, pp. 2023–06, 2023.

Jilei Liu, Hongru Shen, Kexin Chen, and Xiangchun Li. Large language model produces high accurate diagnosis of cancer from end-motif profiles of cell-free dna. *Briefings in Bioinformatics*, 25(5):bbae430, 2024.

Rachel K Luu and Markus J Buehler. Bioinspiredllm: Conversational large language model for the mechanics of biological and bio-inspired materials. *Advanced Science*, 11(10):2306724, 2024.

Serghei Mangul, Viorel MUNTEANU, Timur SUHODOLSCHI, Dumitru CIORBA, and Wei WANG. Biollmbench: A comprehensive benchmarking of large language models in bioinformatics. 2024.

Clint Morris, Michael Jurado, and Jason Zutty. Llm guided evolution-the automation of models advancing models. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 377–384, 2024.

World Health Organization. *Future surveillance for epidemic and pandemic diseases: a 2023 perspective*. World Health Organization, 2023.

Fazle Rafsani, Devam Sheth, Yiming Che, Jay Shah, Md Mahfuzur Rahman Siddiquee, Catherine Chong, Simona Nikolva, Gina Dumkrieger, Baoxin Li, Teresa Wu, et al. Using large-scale contrastive language-image pre-training to maximize brain mri-based headache classification (p4-12.007). In *Neurology*, volume 104, pp. 2728. Lippincott Williams & Wilkins Hagerstown, MD, 2025.

Sirigade Ruekit, Apichai Srijan, Oralak Serichantalergs, Katie R Margulieux, Patrick Mc Gann, Emma G Mills, William C Stribling, Theerasak Pimsawat, Rosarin Kormanee, Suthisak Nakornchai, et al. Molecular characterization of multidrug-resistant eskapee pathogens from clinical samples in chonburi, thailand (2017–2018). *BMC infectious diseases*, 22(1):695, 2022.

Ofir Ben Shoham and Nadav Rappoport. Cpllm: Clinical prediction with large language models. *PLOS Digital Health*, 3(12):e0000680, 2024.

Shovito Barua Soumma, Abdullah Mamun, and Hassan Ghasemzadeh. Domain-informed label fusion surpasses llms in free-living activity classification. 2025a.

Shovito Barua Soumma, Fahim Shahriar, Umme Niraj Mahi, Md Hasin Abrar, Md Abdur Rahman Fahad, and Abu Sayed Md Latiful Hoque. Design and implementation of a scalable clinical data warehouse for resource-constrained healthcare systems. *arXiv preprint arXiv:2502.16674*, 2025b.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.